



UNIVERSITÀ DEGLI STUDI DI PERUGIA
Dipartimento di Matematica e Informatica



Cybersecurity

Appunti Bistarelli

Anno accademico 2022/2023

Indice

1 Cybersecurity	8
1.1 Introduzione alla sicurezza informatica	8
1.1.1 Terminologie	11
1.1.2 Asset di un sistema informatico	13
1.1.3 Vulnerabilità, minacce e attacchi	13
1.1.4 Asset e minacce	17
1.1.5 Attacchi passivi e attivi	19
1.1.6 Requisiti di sicurezza	21
2 Capitolo 2	25
2.1 Riservatezza con la crittografia simmetrica	25
2.1.1 Crittografia Simmetrica	26
2.1.2 Algoritmi di crittografia a blocchi simmetrici	27
2.1.3 Cifrari a flusso	30
2.2 Autenticazione basata su messaggio e funzioni Hash	31
2.2.1 Autenticazione usando la crittografia simmetrica	31
2.2.2 Autenticazione dei messaggi senza crittografia dei messaggi	32
2.2.3 Altre applicazioni delle funzioni hash	38
2.3 Crittografia a chiave pubblica	38
2.3.1 Struttura della crittografia a chiave pubblica	38
2.3.2 Applicazioni dei sistemi crittografici a chiave pubblica	41
2.3.3 Requisiti per la crittografia a chiave pubblica	42
2.3.4 Algoritmi di crittografia asimmetrica	43
2.4 Firme digitali e gestione delle chiavi	44
2.4.1 Firme digitali	44
2.4.2 Certificati a chiave pubblica	46
2.4.3 Scambio di chiavi simmetriche con crittografia a chiave pubblica	48

2.4.4	Buste Digitali	49
2.5	Applicazione pratica: Crittografia dei dati memorizzati	50
3	Capitolo3	52
3.1	Principi dell'autenticazione digitale	52
3.1.1	Un modello per l'autenticazione digitale degli utenti	53
3.1.2	Mezzi di Autenticazione	55
3.1.3	Valutazione dei rischi per l'autenticazione degli utenti	56
3.2	Autenticazione basata su password	58
3.2.1	Vulnerabilità delle password	59
3.2.2	Uso di password con hash	61
3.2.3	Cracking delle password scelte dall'utente	64
3.2.4	Controllo dell'accesso ai file di password	65
3.2.5	Strategie selezione password	66
3.3	Autenticazione Basata sui token	67
3.3.1	Memory Cards	67
3.3.2	Smart Cards	68
3.3.3	Elettronica Identity Cards	70
3.4	Autenticazione Biometrica	71
3.4.1	Caratteristiche fisiche utilizzate nelle applicazioni biometriche	71
3.4.2	Funzionamento di un sistema di autenticazione biometrica	73
3.4.3	Precisione Biometrica	75
3.4.4	Protocollo delle password	77
3.4.5	Protocollo di Token	78
3.4.6	Protocollo biometrico statico	79
3.4.7	Protocollo Biometrico Dinamico	79
4	Keberos	80
4.0.1	Caratteristiche	81
4.0.2	Descrizione del protocollo	83
4.0.3	Accesso basato su client utente	84
4.0.4	Autenticazione client	84
4.0.5	Autenticazione del servizio clienti	84
4.0.6	Richiesta di servizio clienti	85
4.0.7	Inconvenienti e limitazioni	85
4.0.8	Vulnerabilità	86

4.0.9	Altra descrizione del protocollo (wikipedia)	87
5	Capitolo4	90
5.1	Principi di controllo dell'accesso	90
5.1.1	Contesto del controllo dell'accesso	92
5.1.2	Politiche di controllo dell'accesso	93
5.1.3	Soggetti oggetti e diritti d'accesso	94
5.1.4	Controllo dell'accesso discrezionale	95
5.1.5	Un modello di controllo d'accesso	99
5.2	Esempio: Controllo di accesso ai file Unix	103
5.2.1	Controllo di accesso ai file UNIX tradizionale	104
5.2.2	Liste di controllo d'accesso in UNIX	107
5.3	Controllo d'accesso basato sul ruolo	109
5.3.1	Modelli di riferimento RBAC	112
5.4	Controllo degli accessi basato su attributi	116
5.4.1	Attributi	117
5.4.2	Architettura logica ABAC	119
5.4.3	Politiche ABAC	121
5.5	Gestione dell'entità, delle credenziali e dell'accesso	125
5.5.1	Gestione dell'identità	126
5.5.2	Gestione delle credenziali	128
5.5.3	Gestione degli accessi	129
5.5.4	Federazione delle identità	130
5.6	Strutture di fiducia	130
5.6.1	Approccio tradizionale allo scambio di identità	131
5.6.2	Approccio di fiducia per l'identità aperta	132
5.7	Caso di studio: Sistema RBAC per una banca	136
6	Capitolo 5	139
6.1	La necessità di sicurezza del database	139
6.2	Sistemi di gestione dei database	140
6.3	Sql Injection Attacks	142
6.3.1	Un tipico attacco SQLi	143
6.3.2	Vie e tipi di attacco SQLi	145
6.3.3	Contromisure per attacchi SQLi	147
6.4	Controllo dell'accesso al database	148

6.4.1	Autorizzazioni a cascata	148
6.4.2	Controllo dell'accesso basato sui ruoli	150
6.5	Interferenze	153
6.6	Crittografia del database	156
6.7	Sicurezza dei data center	158
6.7.1	Elementi dei data center	158
6.7.2	Considerazioni sulla sicurezza dei data center	159
6.7.3	TIA-492	161
7	Capitolo 22	163
7.1	Posta elettronica sicura e S/MIME	163
7.1.1	MIME	163
7.1.2	S/MIME	163
7.2	Domainkeys che identifica la posta	167
7.2.1	Architettura della posta elettronica	167
7.2.2	Strategia DKIM	169
7.3	Secure Sockets Layer (SSL) e Transport Layer Security (TLS)	171
7.3.1	Architettura TLS	171
7.3.2	Protocollo TLS	172
7.3.3	Attacchi SSL/TLS	179
7.4	HTTPS	179
7.4.1	Avvio della connessione	180
7.4.2	Chiusura della connessione	180
7.5	Sicurezza IPv4 e IPv6	181
7.5.1	Il campo di applicazione di IPsec	183
7.5.2	Security Associations	183
7.5.3	L'Encapsulating Security Payload	185
7.5.4	Transport and Tunnel Modes	186
8	Secure Interoperation	188
8.0.1	Acces Reconfiguration	192
8.0.2	Acces Transitivity	196
8.0.3	Sistemi speciali: multilevel security	197
9	Capitolo 27	202
9.1	Il modello Bell-LaPadula per la sicurezza informatica	202

9.1.1	Modelli di sicurezza informatica	202
9.1.2	Descrizione Generale	203
9.1.3	Descrizione formale del modello	205
9.1.4	Operazioni Astratte	207
9.1.5	Esempio di utilizzo Bella-Pabula	208
9.1.6	Esempio di implementazione Multics	213
9.1.7	Limitazioni modello BLP	214
9.2	Altri modelli per la sicurezza informatica	215
9.2.1	Modello di integrità Biba	215
9.2.2	Modello di integrità Clark-Wilson	216
9.2.3	Modello Muraglia Cinese	220
9.3	Il concetto di sistemi fidati	223
9.3.1	Applicazione sicurezza multilivello	223
9.3.2	Sicurezza multilivello per il controllo dell'accesso basato sui ruoli	224
9.3.3	Sicurezza dei database e sicurezza multilivello	225
9.4	Trusted Computing e il Trusted Platform Module	227
9.4.1	Servizio di avvio autenticato	228
9.4.2	Servizio di certificazione	228
9.4.3	Servizio di crittografia	229
9.4.4	Funzioni TPM	230
9.4.5	Requisiti	232
9.4.6	Profili e Obiettivi	234
9.4.7	Esempio di protezione di un profilo	235
9.5	Assicurazione e valutazione	237
9.5.1	Destinatari	237
9.5.2	Ambito di garanzia	238
9.5.3	Processo di valutazione	240
10	Blockchain e Bitcoin	242
10.1	Blockchain	242
10.2	Bitcoin	243
10.2.1	Bitcoin core	245
10.2.2	Rischi di transazioni digitali	248
10.2.3	Coinbase	251
10.2.4	Dettagli su come si sbloccano le transazioni	255
10.2.5	Pay to Public Key Hash (P2PKH)	259

10.2.6 M-of-N Multi-signature (multisig)	263
10.2.7 Data Output (OP Return)	266
10.2.8 Contenuti tipici transazioni non standard	270
10.2.9 Sicurezza della blockchain	273
10.2.10 Double Spending	275
10.2.11 Wallet Attack	279
10.2.12 Transaction Malleabilit	279
10.2.13 Code vulnerability	281

Capitolo 1

Cybersecurity

1.1 Introduzione alla sicurezza informatica

Il rapporto interno/interagenzia NIST NISTIR 7298 (Glossario di informazioni chiave Termini di sicurezza, maggio 2013) definisce il termine sicurezza informatica come segue:

Misure e controlli che garantiscono riservatezza, integrità, e disponibilità delle risorse del sistema informativo inclusi hardware, software, firmware, e le informazioni che vengono elaborate, archiviate e comunicate.

Questa definizione introduce tre obiettivi chiave che sono al centro della cybersecurity:

- **Confidentiality** (Riservatezza): conservazione delle restrizioni autorizzate all’accesso alle informazioni e divulgazione, compresi i mezzi per proteggere la privacy personale e le proprie informazione. Una perdita di riservatezza è la divulgazione non autorizzata di informazioni. Questo termine copre due concetti correlati:
 - **Data Confidentiality** : garantisce che le informazioni private o riservate non siano disponibili o divulgare a soggetti non autorizzati.
 - **Privacy**: assicura che le persone controllino o influenzino le informazioni ad essi relativi, esse possono essere raccolte e conservate, inoltre si definisce da chi e a chi possono essere divulgare.
- **Integrity** (Integrità): prevenire la modifica o la distruzione impropria delle informazioni, compresa la garanzia del non ripudio e dell’autenticità delle informazioni. Una perdita di l’integrità è la modifica o la distruzione non autorizzata di informazioni. Questo termine copre due concetti correlati:

- **Data integrity**: garantisce che le informazioni e i programmi vengano modificati solo in modo determinato e autorizzato.
- **System integrity**: assicura che un sistema svolga la sua funzione prevista in modo inalterato, libero da intenzionali o involontarie manipolazioni non autorizzate del sistema.
- **Availability**(Disponibilità): garantisce un accesso tempestivo e affidabile nell'utilizzo delle informazioni. Una perdita di disponibilità è l'interruzione dell'accesso o dell'uso di informazioni o un sistema informativo.

Questi tre concetti formano quella che viene spesso definita la triade della CIA. I tre concetti incarnano gli obiettivi di sicurezza fondamentali sia per i dati che per le informazioni e servizi informatici. Ad esempio, lo standard FIPS 199 del NIST (Standards for Security Categorization of Federal Information and Information Systems , febbraio 2004) elenca la riservatezza, integrità e disponibilità come i tre obiettivi di sicurezza per le informazioni e per i sistemi informativi.

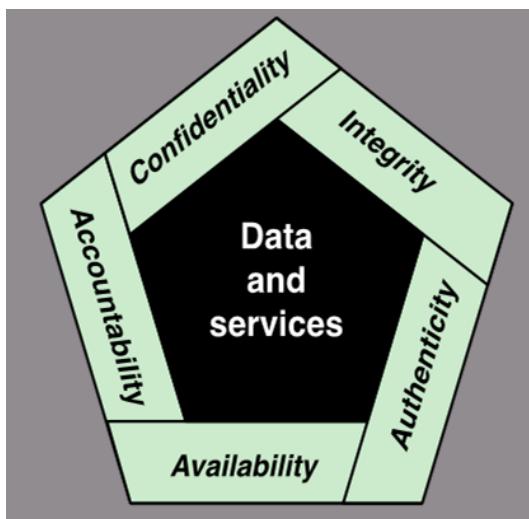


Figura 1.1: Requisiti essenziali in Cybersecurity.

Sebbene l'uso della triade della CIA per definire gli obiettivi di sicurezza sia ben consolidato, alcuni nel campo della sicurezza ritengono che siano necessari concetti aggiuntivi da presentare un quadro completo. Due dei più comunemente citati sono i seguenti:

- **Authenticity** (Autenticità): la proprietà di essere genuini e di poter essere verificati e di essere trusted. Fiducia nella validità di una trasmissione di un messaggio o di un

messaggio originatore. Ciò significa verificare che gli utenti siano chi dicono di essere e che ogni input che arriva al sistema proviene da una fonte attendibile.

- **Accountability**(Rendicontabilità): è la capacità di un sistema di identificare un singolo utente, di determinarne le azioni e il comportamento all'interno del sistema stesso. La rendicontabilità è un aspetto del controllo di accesso e si basa sulla concezione che gli individui siano responsabili delle loro azioni all'interno del sistema. Questo supporta il non ripudio, deterrenza, isolamento dei guasti, rilevamento e prevenzione delle intrusioni, il recupero post-azione in concomitanza con l'azione legale . Poiché i sistemi veramente sicuri non sono ancora un obiettivo realizzabile, dobbiamo essere in grado di tracciare una violazione della sicurezza al/ai responsabile/i. I sistemi devono tenere traccia delle loro attività per consentire successive analisi forensi per rintracciare violazioni della sicurezza o per aiutare nelle controversie sulle transazioni.

Si noti che FIPS 199 include l'autenticità sotto integrità.

La sicurezza informatica è allo stesso tempo affascinante e complessa, alcuni dei motivi sono:

1. *La sicurezza informatica non è così semplice come potrebbe sembrare a un principiante.* I requisiti sembrano essere semplici,in effetti, la maggior parte dei requisiti principali per i servizi di sicurezza possono essere definiti con etichette formate autoesplicative formate da una sola parola: riservatezza, autenticazione, non ripudio e integrità. Ma i meccanismi utilizzati per soddisfare tali requisiti possono essere piuttosto complessi, e capirli può portare a un ragionamento piuttosto sottile.
2. *Nello sviluppo di un particolare meccanismo di sicurezza o algoritmo, bisogna sempre considerare potenziali attacchi a tali funzionalità di sicurezza.* In molti casi gli attacchi di successo sono progettati guardando un problema in un modo completamente differente, dunque sfruttando una debolezza inaspettata del meccanismo.
3. *A causa del punto 2 , le procedure usate per fornire dei servizi particolari sono spesso controidutive.* Tipicamente, un meccanismo di sicurezza è complesso e non è ovvio dalle dichiarazioni di una particolare esigenza che tali misure elaborate sono necessarie. Solo quando si prendono in considerazione i vari aspetti della minaccia si elaborano i meccanismi di sicurezza hanno un senso.
4. *I meccanismi di sicurezza in genere coinvolgono più di un particolare algoritmo o protocollo.* Richiedono inoltre che i partecipanti siano in possesso di un'informazione segreta (ad es. una chiave di crittografia), che sollevano domande sulla creazione,

distribuzione e protezione di tali informazioni segrete. Potrebbe esserci anche una dipendenza sui protocolli di comunicazione il cui comportamento può complicare il compito di sviluppare il meccanismo di sicurezza. Ad esempio, se il corretto funzionamento del meccanismo di sicurezza richiede la definizione di limiti di tempo per il tempo di transito di un messaggio dal mittente al destinatario, allora qualsiasi protocollo o rete che introduce variabili e/o ritardi imprevedibili può rendere tali termini privi di significato.

5. *La sicurezza informatica è essenzialmente una battaglia di ingegni tra un perpetratore che prova a trovare buchi e il progettista o l'amministratore che tenta di chiuderli.* Il grande vantaggio che l'attaccante ha è che lei o lui ha solo bisogno di trovare una singola vulnerabilità, mentre il progettista deve trovare e eliminare tutte le vulnerabilità per ottenere una sicurezza perfetta.
6. *La sicurezza è ancora troppo spesso un'"aggiunta" (surplus) per essere incorporata in un sistema dopo che il progetto è completo, piuttosto che essere parte integrante del processo di progettazione.*
7. *La sicurezza richiede un monitoraggio regolare, anche costante, e questo è difficile nei tempi attuali.*
8. *C'è una naturale tendenza da parte di utenti e gestori di sistema a percepire pochi vantaggi nell'investimento sulla sicurezza fino a quando non si verifica un problema.*
9. *Molti utenti e persino gli amministratori della sicurezza vedono una sicurezza forte come un ostacolo al funzionamento o all'uso efficiente di un sistema informativo o di un'informazione.*

1.1.1 Terminologie

La maggior parte delle terminologie sono riportate nel Capitolo ?? degli appunti Prof. Santini, di seguito riporto alcuni termini non citati in precedenza.

Risorsa di sistema (Asset)

Una applicazione maggiore, un sistema di supporto generale, un programma ad alto impatto, un impianto fisico, un sistema mission-critical, personale, apparecchiature o un gruppo di sistemi logicamente correlati.

Minaccia

Qualsiasi circostanza o evento che potrebbe avere un impatto negativo sulle operazioni organizzative (inclusi missione, funzioni, immagine o reputazione), risorse organizzative, individui, altre organizzazioni o la Nazione stessa attraverso un sistema informativo tramite accesso, distruzione, divulgazione, modifica non autorizzati delle informazioni , e/o negazione del servizio.

Contromisure

Dispositivo o tecniche che hanno come obiettivo la compromissione dell'efficacia operativa di attività indesiderate o contraddittorie, o la prevenzione di spionaggio, sabotaggio, furto o accesso o utilizzo non autorizzato di informazioni sensibili o di sistemi informativi.

Rischio

Una misura del grado in cui un'entità è minacciata da una potenziale circostanza o evento, e tipicamente una funzione di stima:

1. degli impatti negativi che si verificherebbero se la circostanza o l'evento si verificassero
2. della probabilità che si verifichi.

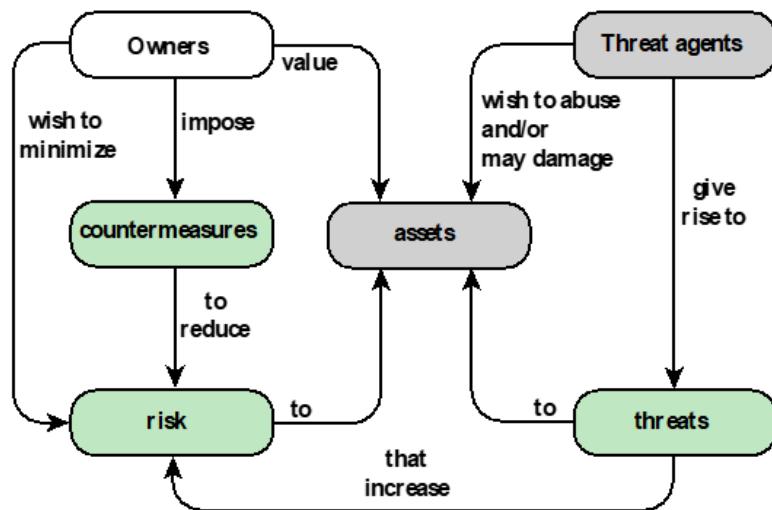


Figura 1.2: Concetti di sicurezza e le loro relazioni.

1.1.2 Asset di un sistema informatico

Gli asset di un sistema informatico possono essere suddivisi come di seguito:

- **Hardware:** compresi i sistemi informatici e altri trattamenti di dati, archiviazione dei dati,e dispositivi di comunicazione dati;
- **Software:** include il sistema operativo, le utilità di sistema e le applicazioni;
- **Data:** inclusi file e database, nonché dati relativi alla sicurezza, ad esempio file di password.
- **Strutture e reti di comunicazione:** rete locale e geografica collegamenti di comunicazione, bridge, router e così via.

1.1.3 Vulnerabilità, minacce e attacchi

Nel contesto della sicurezza, la nostra preoccupazione riguarda le vulnerabilità delle risorse del sistema. [NRC02] elenca le seguenti categorie generali di vulnerabilità di un sistema informatico o di una risorsa di rete:

- Il sistema può essere danneggiato (**corrupted**), quindi fa la cosa sbagliata o dà risposte sbagliate. Ad esempio, i valori dei dati memorizzati possono differire da quello che dovrebbero essere perché sono stati modificati in modo improprio.
- Il sistema avere delle perdite (**be leaky**). Ad esempio, qualcuno che non dovrebbe avere accesso ad alcune o a tutte le informazioni disponibili attraverso la rete ottengono tale accesso.
- Il sistema può diventare non disponibile (**unavailable**) o molto lento. Cioè, usando il sistema o la rete diventa impossibile o impraticabile.

Questi tre tipi generali di vulnerabilità corrispondono ai concetti di integrità, riservatezza e disponibilità, enumerati in precedenza. Una **minaccia** rappresenta un potenziale danno alla sicurezza di una risorsa. Un **attacco** è una minaccia che viene eseguita (azione di minaccia) e, in caso di successo, comporta una violazione indesiderata della sicurezza o a una conseguenza della minaccia. L'agente che effettua l'attacco viene definito **attaccante** o **agente di minaccia**. Possiamo distinguere gli attacchi in due tipi:

- **Attacco attivo:** un tentativo di alterare le risorse del sistema o di influenzare il funzionamento.

- **Attacco passivo:** un tentativo di imparare o di fare uso delle informazioni da un sistema che non influenza le risorse di quest'ultimo.

Possiamo classificare gli attacchi in base all'origine di questi:

- **Attacco interno:** iniziato da un entità interna al perimetro di sicurezza (un "insider"). L'insider è autorizzato all'accesso alle risorse del sistema ma le usa in un modo non approvato da coloro che ne garantiscono l'accesso.
- **Attacco esterno:** iniziato fuori dal perimetro, da un utente non autorizzato o illegittimo del sistema (un "outsider"). Su Internet, potenziale aggressori esterni variano dai dilettanti "burloni" a criminali organizzati, internazionali terroristi e governi ostili.

Infine, una contromisura è qualsiasi mezzo adottato per affrontare un attacco alla sicurezza. Idealmente, una contromisura può essere escogitata per prevenire un particolare tipo di attacco dall'avere successo. Quando la prevenzione non è possibile, o in alcuni casi fallisce, l'obiettivo è rilevare l'attacco e poi riprendersi dagli effetti . Una contromisura stessa può introdurre nuove vulnerabilità. In ogni caso, vulnerabilità residue possono rimanere dopo l'imposizione di contromisure. Tali vulnerabilità possono essere sfruttato da attaccanti che rappresentano un livello di rischio residuo per gli asset. I proprietari dell'asset cercheranno di ridurre al minimo tale rischio dati altri vincoli.

Threat Consequence	Threat Action (Attack)
Unauthorized Disclosure A circumstance or event whereby an entity gains access to data for which the entity is not authorized.	Exposure: Sensitive data are directly released to an unauthorized entity. Interception: An unauthorized entity directly accesses sensitive data traveling between authorized sources and destinations. Inference: A threat action whereby an unauthorized entity indirectly accesses sensitive data (but not necessarily the data contained in the communication) by reasoning from characteristics or by-products of communications. Intrusion: An unauthorized entity gains access to sensitive data by circumventing a system's security protections.
Deception A circumstance or event that may result in an authorized entity receiving false data and believing it to be true.	Masquerade: An unauthorized entity gains access to a system or performs a malicious act by posing as an authorized entity. Falsification: False data deceive an authorized entity. Repudiation: An entity deceives another by falsely denying responsibility for an act.
Disruption A circumstance or event that interrupts or prevents the correct operation of system services and functions.	Incapacitation: Prevents or interrupts system operation by disabling a system component. Corruption: Undesirably alters system operation by adversely modifying system functions or data. Obstruction: A threat action that interrupts delivery of system services by hindering system operation.
Usurpation A circumstance or event that results in control of system services or functions by an unauthorized entity.	Misappropriation: An entity assumes unauthorized logical or physical control of a system resource. Misuse: Causes a system component to perform a function or service that is detrimental to system security.

Figura 1.3: Conseguenze delle minacce e azioni che le causano

La tabella 1.3 basata su RFC 4949, descrive quattro tipi di conseguenze ed elenca i tipi di attacchi che risultano in ciascuna conseguenza.

La divulgazione non autorizzata (Unauthorized disclosure) è una minaccia alla riservatezza. I seguenti tipi di attacchi possono portare a queste conseguenze

- **Esposizione:** quando un insider rilascia intenzionalmente informazioni sensibili a un estraneo, come ad esempio i numeri di carta di credito. Può anche essere il risultato di un errore umano, hardware o software, che si traduce nell'azione da parte di un'entità di avere l'accesso non autorizzato di dati sensibili. Ce ne sono stati numerosi casi di questo, come le università che pubblicano accidentalmente le informazioni confidenziali degli studenti sul Web.
- **Intercettazione:** l'intercettazione è un attacco comune nel contesto delle comunicazioni. Su una rete locale condivisa (LAN), come una LAN wireless o a broadcast Ethernet, qualsiasi dispositivo collegato alla LAN può ricevere una copia dei pacchetti destinati a un altro dispositivo. Su Internet, un determinato hacker può accedere al

traffico di posta elettronica e ad altri trasferimenti di dati. Tutte queste situazioni possono portare all'accesso non autorizzato ai dati.

- **Inferenza:** un esempio di inferenza è noto come analisi del traffico, in cui un avversario è in grado di ottenere informazioni osservando l'andamento del traffico una rete, come la quantità di traffico tra particolari coppie di host sulla rete. Un altro esempio è l'inferenza di informazioni dettagliate da un database di un utente che ha solo un accesso limitato, questo è realizzato da query ripetute i cui risultati combinati consentono l'inferenza.
- **Intrusione:** un esempio di intrusione è un avversario che ottiene l'accesso non autorizzato a dati sensibili superando le protezioni di controllo degli accessi del sistema.

L'inganno (Deception) è una minaccia per l'integrità del sistema o per l'integrità dei dati.

I seguenti tipi di attacchi possono portare a queste conseguenze:

- **Masquerade:** un esempio di masquerade è un tentativo di accesso a un sistema da parte di un utente non autorizzato spacciandosi per uno autorizzato, questo può succedere se l'utente non autorizzato conosce l'ID di accesso e la password di un altro utente. Un altro esempio è la logica dannosa (malicious logic), come un cavallo di Troia, che appare per eseguire una funzione utile o desiderabile, ma in realtà ottiene l'accesso non autorizzato alle risorse di sistema o induce un utente a eseguire altre logiche dannose.
- **Falsificazione:** si riferisce all'alterazione o sostituzione di dati validi o all'introduzione di dati falsi in un file o database. Ad esempio, uno studente può alterare i suoi voti su un database scolastico.
- **Ripudio:** in questo caso, un utente nega l'invio di dati o nega di ricevere o possedere i dati.

L'interruzione (Disruption) è una minaccia alla disponibilità o all'integrità del sistema. I

seguenti tipi di attacchi possono portare a queste conseguenze:

- **Incapacità:** questo è un attacco alla disponibilità del sistema. Ciò potrebbe verificarsi come risultato della distruzione fisica o del danneggiamento dell'hardware del sistema. Più tipicamente, un software dannoso, come Trojan, virus o worm, potrebbero operare in modo tale da disabilitare un sistema o alcuni dei suoi servizi.

- **Corruzione:** questo è un attacco all'integrità del sistema. Un Software dannoso in questo contesto potrebbe funzionare in modo tale che le risorse di sistema o i servizi funzionino in modo non intenzionale. Oppure un utente potrebbe ottenere l'accesso non autorizzato a un sistema e modificarne alcune funzioni. Un esempio di quest'ultimo è un posizionamento di una logica backdoor (backdoor logic) nel sistema per fornire il successivo accesso al sistema stesso e alle sue risorse con una procedura diversa da quella abituale.
- **Ostruzione:** un modo per ostacolare il funzionamento del sistema è interferire con le comunicazioni disabilitando i collegamenti di comunicazione o alterando la comunicazione delle informazioni di controllo. Un altro modo è sovraccaricare il sistema mettendo un carico in eccesso sul traffico di una comunicazione o sulle risorse di elaborazione.

L'usurpazione (Usurpation) è una minaccia per l'integrità del sistema. I seguenti tipi di attacchi possono portare a queste conseguenze:

- **Appropriazione indebita:** può includere il furto del servizio. Un esempio è un attacco Denial of Service distribuito, quando il software dannoso è installato su degli host da utilizzare come piattaforme per avviare il traffico verso un host di destinazione. In questo caso, il software maligno fa uso non autorizzato delle risorse del processore e del sistema operativo.
- **Uso improprio:** l'uso improprio può verificarsi per mezzo di una malicious logic o di un hacker che ha ottenuto un accesso non autorizzato a un sistema. In entrambi i casi, le funzioni di sicurezza possono essere disabilitate o contrastate.

1.1.4 Asset e minacce

Le risorse di un sistema informatico possono essere classificate come hardware, software, dati, linee e reti di comunicazione. In questa sottosezione li descriviamo brevemente e mettendoli in relazione con i concetti di integrità, riservatezza e disponibilità.

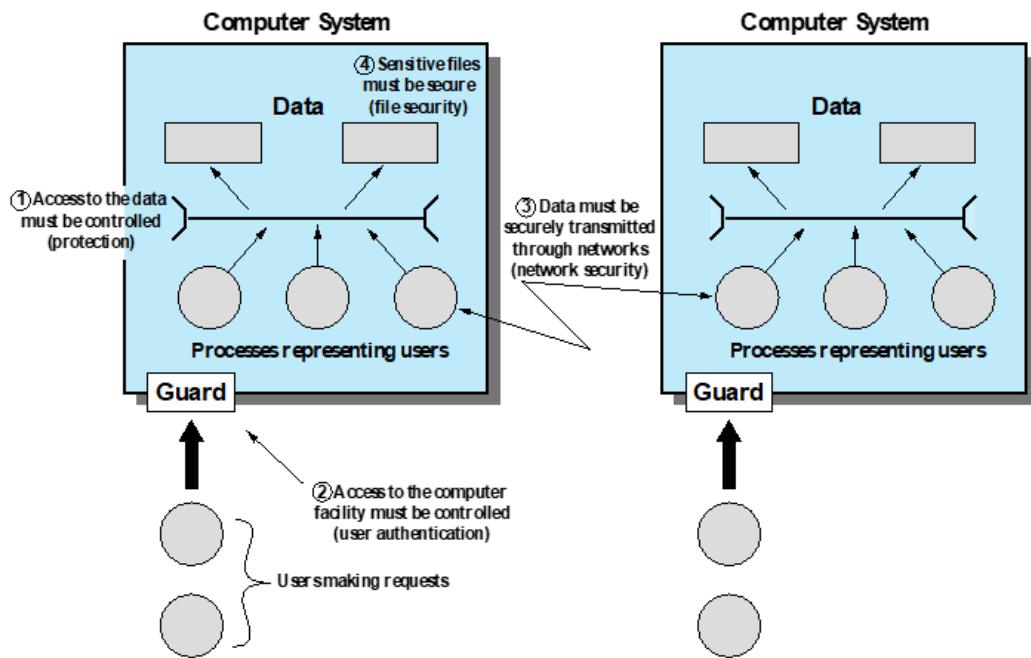


Figura 1.4: Scopo della sicurezza informatica.

Hardware. Una delle principali minacce per l'hardware del computer è la minaccia alla disponibilità. L'hardware è il più vulnerabile agli attacchi e il meno suscettibile ai controlli automatizzati. Le minacce includono danni accidentali e deliberati alle apparecchiature così come il furto. La proliferazione di personal computer e workstation e l'uso diffuso delle LAN aumenta il potenziale di perdite in quest'area. Il furto delle unità USB può portare alla perdita di riservatezza. Le misure di sicurezza fisiche e amministrative sono necessarie per far fronte a queste minacce.

Software. Il software include il sistema operativo, le utilità e l'applicazione programmi.

Una delle principali minacce al software è un attacco alla disponibilità. Il Software, in particolare quello applicativo, è spesso facile da eliminare. Il software può anche essere modificato o danneggiato per renderlo inutilizzabile. Un'attenta gestione della configurazione del software, che include il mantenere dei backup della versione più recente, può mantenere una disponibilità alta. Un problema più difficile da affrontare è la modifica del software che si ha in un programma il quale funziona ancora ma che si comporta in modo diverso rispetto a prima, questa è una minaccia per l'integrità/autenticità. Rientrano in questa categoria i virus informatici e i relativi attacchi. Un ultimo problema è la

protezione contro la pirateria del software. Sebbene siano disponibili alcune contromisure, in linea di massima il problema di copie non autorizzate del software non è stata risolta.

Data. Un problema molto più diffuso è la sicurezza dei dati, che coinvolge file e altre forme di dati controllati da individui, gruppi e organizzazioni aziendali.

I problemi di sicurezza relativi ai dati sono ampi e comprendono disponibilità, segretezza e integrità. In caso di disponibilità, la preoccupazione è con la distruzione di file di dati, che può verificarsi accidentalmente o intenzionalmente. Una preoccupazione evidente per la segretezza è la lettura non autorizzata di file di dati o database, e quest'area è stata forse oggetto di ulteriori ricerche e sforzi rispetto a qualsiasi altro settore della sicurezza informatica. Una minaccia meno ovvia alla segretezza comporta l'analisi dei dati e si manifesta nell'utilizzo delle cosiddette banche dati statistiche, che forniscono informazioni di sintesi o aggregate. Presumibilmente, l'esistenza delle informazioni aggregate non minacciano la privacy delle persone coinvolte. Tuttavia, con la crescita dell'uso delle banche dati statistiche, c'è un rischio crescente per la divulgazione di informazioni personali. In sostanza, le caratteristiche di un individuo possono essere identificate attraverso un'analisi attenta. Ad esempio, se una tabella registra l'aggregato dei redditi degli intervistati A, B, C e D e un altro registra l'aggregato dei redditi di A, B, C, D ed E, la differenza tra i due aggregati sarebbe il reddito di E. Questo problema è esasperato dal desiderio crescente di combinare set di dati. In molti casi, abbinando diversi set di dati per coerenza tra i diversi livelli di aggregazione è necessario l'accesso alle singole unità. Pertanto, le singole unità, che sono oggetto di problemi di privacy, sono disponibili in varie fasi del trattamento dei set di dati. Infine, l'integrità dei dati è una delle principali preoccupazioni nella maggior parte delle installazioni. Le modifiche ai file di dati possono avere conseguenze che vanno da minori a disastrose.

1.1.5 Attacchi passivi e attivi

Gli attacchi alla sicurezza della rete possono essere classificati come attacchi passivi e attacchi attivi. Un attacco passivo tenta di imparare o fare uso delle informazioni del sistema, ma non influisce sulle risorse di quest'ultimo. Un attacco attivo tenta di alterare le risorse di sistema o di influenzare il loro funzionamento.

Attacchi passivi

Gli **attacchi passivi** generalmente riguardano l'intercettazione o il monitoraggio di trasmissioni di dati. L'obiettivo dell'attaccante è ottenere le informazioni che vengono

trasmesse. Due tipi di attacchi passivi sono il rilascio del contenuto dei messaggi e dell'analisi del traffico.

Rilascio dei contenuti di un messaggio Il **rilascio dei contenuti** del messaggio è facilmente comprensibile. Una conversazione telefonica, un messaggio di posta elettronica e un file trasferito possono contenere dati sensibili o informazioni confidenziali. Vorremmo impedire a un avversario di imparare il contenuto di queste trasmissioni.

Analisi del traffico. Un secondo tipo di attacco passivo, **l'analisi del traffico**, è più sottile. Supponiamo che noi abbiamo un modo per mascherare il contenuto dei messaggi o altre informazioni del traffico di dati, in modo che gli oppositori, anche se hanno catturato il messaggio, non possono estrarre le informazioni dal messaggio. La tecnica comune per mascherare i contenuti è la crittografia. Anche se disponiamo di una protezione crittografica, un avversario potrebbe comunque essere in grado di osservare lo schema di questi messaggi. L'avversario potrebbe determinare la posizione e l'identità degli host nella comunicazione e potrebbe osservare la frequenza e la lunghezza dei messaggi scambiati. Queste informazioni potrebbero essere utili per indovinare la natura della comunicazione che stava avvenendo.

Gli attacchi passivi sono molto difficili da rilevare perché non coinvolgono alterazione dei dati. In genere, il traffico dei messaggi viene inviato e ricevuto in un modo apparentemente normale e né il mittente né il destinatario sono consapevoli che una terza parte ha letto i messaggi o osservato l'andamento del traffico. Tuttavia, è possibile prevenire il successo di questi attacchi, di solito mediante crittografia. Pertanto, l'enfasi nell'affrontare gli attacchi passivi è sulla prevenzione piuttosto che il rilevamento.

Attacchi attivi

Gli attacchi attivi comportano alcune modifiche del flusso di dati o la creazione di un falso flusso, esso può essere suddiviso in quattro categorie: replay, masquerade, modifica dei messaggi e denial of service.

Replay. Il replay comporta l'acquisizione passiva di un'unità di dati e la sua successiva ritrasmissione per produrre un effetto non autorizzato.

Masquerade. Una masquerade ha luogo quando un'entità finge di essere un'entità diversa. Un attacco di questo tipo di solito include una delle altre forme di attacco attivo.

Per esempio, le sequenze di autenticazione possono essere catturate e riprodotte dopo che è avvenuta una sequenza di autenticazione valida, abilitando così un'entità autorizzata con pochi privilegi a ottenere privilegi extra impersonando un'entità che dispone di tali privilegi.

Modifica di un messaggio. La modifica dei messaggi significa semplicemente che una parte di un legittimo messaggio è alterato, o che i messaggi sono ritardati o riordinati, per produrre un effetto non autorizzato. Ad esempio, un messaggio che afferma: "Consenti a John Smith di leggere dati di file riservati" viene modificato per dire "Consenti a Fred Brown di leggere dati di file riservati".

DOS. La negazione del servizio impedisce o inibisce il normale utilizzo o gestione delle strutture di comunicazione. Questo attacco può avere un obiettivo specifico, per esempio un'entità può sopprimere tutti i messaggi diretti a una particolare destinazione (ad esempio, la sicurezza del servizio di audit). Un'altra forma di rifiuto del servizio è l'interruzione di un'intera rete, o disabilitando la rete o sovraccaricandola di messaggi in modo da degradarne le prestazioni.

Gli attacchi attivi presentano le caratteristiche opposte degli attacchi passivi. Invece gli attacchi passivi sono difficili da rilevare, sono disponibili misure per prevenirli con successo. D'altra parte, è abbastanza difficile prevenire assolutamente gli attacchi attivi, perché per farlo richiederebbe la protezione fisica di tutte le strutture e i percorsi di comunicazione in ogni momento. Invece, l'obiettivo è rilevarli e riprendersi da qualsiasi disservizio o ritardi da essi causati. Poiché il rilevamento ha un effetto deterrente, esso può anche contribuire alla prevenzione.

1.1.6 Requisiti di sicurezza

Esistono diversi modi per classificare e caratterizzare le contromisure che possono essere utilizzate per ridurre le vulnerabilità e affrontare le minacce alle risorse di sistema. In questa sottosezione, vediamo contromisure in termini di requisiti funzionali, e seguiamo la classificazione definita in FIPS 200. Questo standard enumera 17 aree relative alla sicurezza con riguardo alla protezione della riservatezza, dell'integrità e della disponibilità delle informazioni di sistemi e le informazioni elaborate, archiviate e trasmesse da tali sistemi.

1. **Accesso controllato:** limitare l'accesso al sistema informativo agli utenti autorizzati, ai processi che agiscono per conto degli utenti autorizzati, o ai dispositivi (inclusi altri

sistemi informativi) e alle tipologie di transazioni e funzioni che gli utenti autorizzati possono esercitare.

2. **Consapevolezza e Formazione:** garantire che i gestori e gli utenti dei sistemi informativi organizzativi siano consapevole dei rischi per la sicurezza associati alle proprie attività e delle leggi, dei regolamenti e delle politiche applicabili relativi alla sicurezza dei sistemi informativi organizzativi e garantire che il personale sia adeguatamente addestrato a svolgere i compiti e le responsabilità assegnate in materia di sicurezza delle informazioni.
3. **Audit e responsabilità:** creare, proteggere e conservare i record di audit del sistema informativo per consentire il monitoraggio, l'analisi, l'indagine e la segnalazione di atti illeciti, non autorizzati o di attività non appropriate del sistema informativo. Garantire inoltre che le azioni dei singoli individui nel sistema possano essere ricondotte in modo univoco a tali utenti in modo che possano essere ritenuti responsabili di esse.
4. **Certificazione, accreditamento e valutazioni di sicurezza:** valutare periodicamente i controlli di sicurezza nei sistemi informativi organizzativi per determinare se i controlli sono efficaci nella loro applicazione. Sviluppare e attuare piani d'azione volti a correggere le carenze e ridurre o eliminare le vulnerabilità in questi sistemi. Autorizzare l'esercizio dei sistemi informativi organizzativi ed eventuali connessioni associate a questi. Monitorare continuamente i controlli di sicurezza del sistema informativo per garantire la continua efficacia di essi.
5. **Gestione della configurazione:** stabilire e mantenere le configurazioni di base e gli inventari dei sistemi (inclusi hardware, software, firmware e documentazione) durante i rispettivi cicli di vita di sviluppo del sistema. Stabilire e far rispettare le impostazioni di configurazione di sicurezza per i prodotti informatici utilizzati nei sistemi.
6. **Pianificazione di emergenza:** stabilire, mantenere e implementare piani di risposta alle emergenze, operazioni di backup, e il ripristino post-disastro per i sistemi in modo da garantire la disponibilità di risorse informative critiche e continuità operativa in situazioni di emergenza.
7. **Identificazione e autenticazione:** identificare gli utenti del sistema, i processi che agiscono per conto degli utenti o dei dispositivi e autenticare (o verificare) le identità di tali utenti, processi o dispositivi, come prerequisito per consentire l'accesso ai sistemi.

8. **Risposta all'incidente:** stabilire una capacità operativa di gestione degli incidenti per le informazioni organizzative dei sistemi che includono un'adeguata preparazione, rilevamento, analisi, contenimento, recupero e un controllo delle attività di risposta dell'utente. Tracciare, documentare e segnalare gli incidenti ai funzionari appropriati e/o alle autorità.
9. **Manutenzione:** eseguire la manutenzione periodica e tempestiva dei sistemi, fornire controlli efficaci sugli strumenti, sulle tecniche, sui meccanismi e sul personale utilizzato per condurre una manutenzione del sistema informativo.
10. **Protezione dei media:** proteggere i media dei sistemi, sia cartacei che digitali, limitare l'accesso alle informazioni dei media agli utenti autorizzati e sanificare o distruggere prima i supporti del sistema informativo prima dello smaltimento o del rilascio per il riutilizzo.
11. **Protezione fisica e ambientale:** limitare l'accesso fisico dei soggetti autorizzati ai sistemi informativi, alle apparecchiature e ai rispettivi ambienti operativi. Proteggere l'impianto fisico e l'infrastruttura di supporto per i sistemi. Fornire utilità di supporto per i sistemi informativi e proteggere quest'ultimi dai rischi ambientali fornendo adeguati controlli ambientali alle strutture che li contengono.
12. **Pianificazione:** sviluppare, documentare, aggiornare periodicamente e implementare piani di sicurezza per le informazioni organizzative dei sistemi che descrivono i controlli di sicurezza esistenti o previsti e le regole di comportamento dei soggetti che accedono ai sistemi.
13. **Sicurezza del personale:** garantire che le persone che occupano posizioni di responsabilità all'interno delle organizzazioni (compresi i fornitori di servizi di terze parti) siano affidabili e soddisfino i criteri di sicurezza stabiliti per quelle posizioni. Garantire che le informazioni organizzative e i sistemi informativi siano protetti durante e dopo le azioni del personale quali licenziamenti e trasferimenti. Applicare sanzioni formali per il personale che non fa rispettare le politiche e le procedure di sicurezza dell'organizzazione.
14. **Valutazione del rischio:** valutare periodicamente il rischio per le operazioni organizzative (inclusi missioni, funzioni, immagine o reputazione), risorse organizzative e individui, risultanti dal funzionamento del sistema e il relativo trattamento, archiviazione o trasmissione di informazioni organizzative.

15. **Acquisizione di sistemi e servizi:** Allocare risorse sufficienti per proteggere adeguatamente l'organizzazione dei sistemi. Impiegare processi del ciclo di vita dello sviluppo del sistema che incorporano considerazioni sulla sicurezza. Imporre limitazioni all'utilizzo e all'installazione del software e garantire che i fornitori di terze parti adottino adeguate misure di sicurezza per proteggere le informazioni, le applicazioni e/o i servizi "esternalizzati" dall'organizzazione.
16. **Protezione del sistema e delle comunicazioni:** monitorare, controllare e proteggere le comunicazioni organizzative (vale a dire, le informazioni trasmesse o ricevute dai sistemi) ai confini esterni e interni. Impiegare progetti "architettonici", sviluppo software tecniche e principi di ingegneria dei sistemi che promuovono un'efficace sicurezza delle informazioni all'interno di dei sistemi organizzativi.
17. **Integrità del sistema e delle informazioni:** identificare, segnalare e correggere le informazioni e le falte del sistema in modo tempestivo. Fornire protezione da codice dannoso in posizioni appropriate all'interno del sistema organizzativo e monitorare gli avvisi di sicurezza del sistema informativo e adottare le azioni appropriate in risposta.

Capitolo 2

Capitolo 2

2.1 Riservatezza con la crittografia simmetrica

Una delle tecniche per garantire la riservatezza dei dati trasmessi o memorizzati è la crittografia simmetrica. Segue una panoramica dei due più importanti algoritmi di crittografia simmetrica:

- **Il Data Encryption Standard (Standard di Crittografia dei Dati) (DES)**
- **L'Advanced Encryption Standard (AES)**

Che sono algoritmi di crittografia a blocchi, ma esiste anche il concetto di crittografia a flusso.

2.1.1 Crittografia Simmetrica

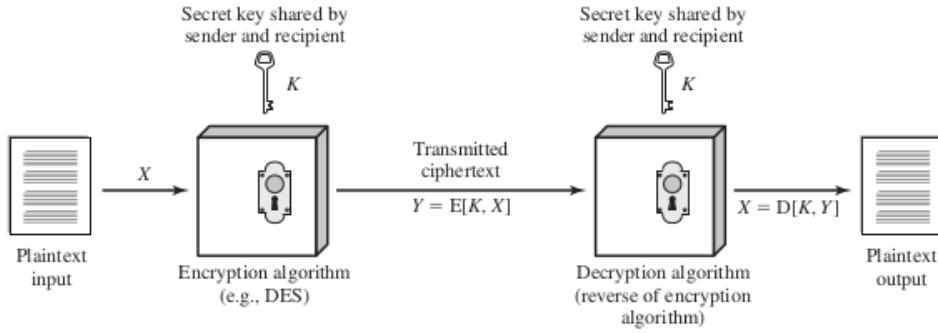


Figure 2.1 Simplified Model of Symmetric Encryption

Uno schema di crittografia simmetrica ha cinque ingredienti (vedi Figura 2.1):

- **Testo in chiaro:** Si tratta del messaggio o dei dati originali che vengono inseriti nell'algoritmo come in ingresso.
- **Algoritmo di crittografia:** L'algoritmo di crittografia esegue varie sostituzioni e trasformazioni sul testo in chiaro.
- **Chiave segreta:** La chiave segreta che è l'input dell'algoritmo di crittografia.
- **Testo cifrato:** È il messaggio criptato prodotto in uscita.

Dipende dal testo in chiaro e dalla chiave segreta. Per un dato messaggio, due chiavi diverse produrranno due cfrari diversi.

- **Algoritmo di decifrazione:** Si tratta essenzialmente dell'algoritmo di crittografia eseguito all'inverso.

Prende il testo cifrato e la chiave segreta e produce il testo in chiaro originale.

I requisiti per un **uso sicuro della crittografia simmetrica** sono due:

1. È necessario un algoritmo di crittografia forte.

Se un avversario conosce l'algoritmo di cifratura ed ha accesso a uno o più testi cifrati non deve comunque essere in grado di decifrarli.

2. Il mittente e il destinatario devono aver ottenuto copie della chiave segreta in modo sicuro e devono mantenerla protetta.

Se qualcuno scopre la chiave e conosce l'algoritmo, tutte le comunicazioni che utilizzano la medesima chiave sono leggibili.

Esistono due approcci generali per attaccare uno schema di crittografia simmetrica.

Il primo attacco è noto come crittoanalisi. Gli attacchi di crittoanalisi si basano sulla natura dell'algoritmo e sulla conoscenza delle caratteristiche generali del testo in chiaro e del testo cifrato.

Questo tipo di attacco sfrutta le caratteristiche dell'algoritmo per cercare di dedurre un testo in chiaro specifico o di dedurre la chiave utilizzata. Se l'attacco riesce a dedurre la chiave, l'effetto è catastrofico: **Tutti i messaggi futuri e passati crittografati con quella chiave sono compromessi.**

Il secondo metodo noto come attacco a forza bruta, consiste nel provare ogni possibile chiave su un pezzo di testo cifrato finché non si ottiene una traduzione comprensibile in testo in chiaro. In media, la metà di tutte le chiavi possibili deve essere provata per ottenere un successo. Cioè, se ci sono x chiavi diverse, in media un attaccante scoprerebbe la chiave effettiva dopo $x/2$ tentativi.

2.1.2 Algoritmi di crittografia a blocchi simmetrici

Gli algoritmi di crittografia simmetrica più utilizzati sono **i cifrari a blocchi**.

Un cifrario a blocchi elabora il testo in chiaro in blocchi di dimensioni fisse e produce un blocco di testo cifrato di dimensioni uguali per ogni blocco di testo in chiaro. L'algoritmo elabora quantità di testo in chiaro come una serie di blocchi di dimensioni fisse. Gli algoritmi simmetrici più importanti, che sono tutti cifrari a blocchi, sono il Data Encryption Standard (DES), il triplo DES e l'Advanced Encryption Standard (AES)

Table 2.1 Comparison of Three Popular Symmetric Encryption Algorithms

	DES	Triple DES	AES
Plaintext block size (bits)	64	64	128
Ciphertext block size (bits)	64	64	128
Key size (bits)	56	112 or 168	128, 192, or 256

DES = Data Encryption Standard

AES = Advanced Encryption Standard

Vedi tabella 2.1

DES Fino a poco tempo fa, lo schema di crittografia più diffuso era basato sul Data Encryption Standard (DES). Il DES utilizza un blocco di testo in chiaro di 64 bit e una chiave di 56 bit per produrre un blocco di testo cifrato di 64 bit. I problemi del DES si dividono in due categorie:

- Problemi sull'algoritmo stesso
- Problemi sull'uso di una chiave a 56 bit.

Il primo problema: si riferisce alla possibilità che la crittoanalisi sia possibile sfruttando le caratteristiche dell'algoritmo DES. Nel corso degli anni, ci sono stati numerosi tentativi di trovare e sfruttare le debolezze dell'algoritmo, rendendo il DES l'algoritmo di crittografia più studiato in assoluto. Nonostante i numerosi approcci, finora nessuno ha segnalato una debolezza "cruciale" nel DES.

IL secondo problema è la lunghezza della chiave. Con una lunghezza di 56 bit, ci sono 256 chiavi possibili, pari a circa $7,2 * 10^{16}$ chiavi. Considerata la velocità dei processori in commercio, questa lunghezza di chiave è tristemente inadeguata. Un documento di Seagate Technology suggerisce che una velocità di un miliardo (10⁹) di combinazioni di chiavi al secondo è ragionevole per gli attuali computer multicore. Le offerte recenti lo confermano. Sia Intel che AMD offrono ora istruzioni basate su hardware per accelerare l'uso di AES. I test condotti su una macchina Intel multicore contemporanea hanno dato come risultato una velocità di crittografia di circa mezzo miliardo di chiavi al secondo. Un'altra recente analisi suggerisce che, con la tecnologia contemporanea dei supercomputer, una velocità di 10¹³ crittografie/s è ragionevole.

Tenendo conto di questi risultati, la Tabella 2.2 mostra quanto tempo è necessario per compiere un attacco di forza bruta per chiavi di varie dimensioni. Come si può notare, un singolo PC può violare il DES in circa un anno. Se più PC lavorano in parallelo, il tempo si riduce drasticamente. E i supercomputer di oggi dovrebbero essere in grado di trovare una chiave in circa un'ora. Le chiavi di dimensioni 128 bit o superiori sono di fatto inviolabili con un semplice approccio di forza bruta. Anche se riuscissimo a velocizzare il

sistema di attacco di un fattore di 1 trilione (1012), ci vorrebbero comunque più di 100.000 anni per decifrare un codice che utilizza una chiave a 128 bit. Fortunatamente esistono diverse alternative al DES, le più importanti delle quali sono il triplo DES e l'AES.

TripleDes La vita del DES è stata prolungata dall'uso del triple DES (3DES), che prevede la **ripetizione dell'algoritmo DES di base per tre volte** utilizzando due o tre chiavi uniche, per una dimensione della chiave di 112 o 168 bit. Il 3DES è stato standardizzato per la prima volta per l'uso in applicazioni finanziarie nello standard ANSI X9.17 nel 1985. Il 3DES ha due caratteristiche che ne assicurano la diffusione nei prossimi anni.

- Grazie alla lunghezza della chiave di 168 bit, supera la vulnerabilità agli attacchi di forza bruta del DES. di attacchi brute-force del DES.
- L'algoritmo di crittografia sottostante al 3DES è lo stesso del DES.

Questo algoritmo è stato sottoposto a maggiori controlli rispetto a qualsiasi altro algoritmo di crittografia per un periodo di tempo più lungo e non è stato trovato alcun attacco di crittoanalisi efficace basato sull'algoritmo piuttosto che sulla forza bruta. Di conseguenza, c'è un'elevata livello di fiducia che il 3DES sia molto resistente alla crittoanalisi.

Il principale svantaggio del 3DES è che l'algoritmo è relativamente lento nel software. Il 3DES, che richiede un numero di calcoli tre volte i calcoli del DES, è di conseguenza più lento. Un altro svantaggio è che sia il DES che il 3DES utilizzano una dimensione di blocco di 64 bit. Per ragioni di efficienza e sicurezza, è auspicabile una dimensione di blocco maggiore.

AES A causa dei suoi svantaggi, il 3DES non è un candidato ottimale per un uso a lungo termine. Per sostituirlo, nel 1997 il NIST ha pubblicato un invito a presentare proposte per un nuovo Advanced Encryption Standard (AES), che dovrebbe avere una forza di sicurezza pari o superiore al 3DES e un'efficienza sicurezza pari o superiore al 3DES ma sicuramente un'efficienza significativamente migliorata.

Oltre a questi requisiti generali, il NIST ha specificato che AES deve essere un cifrario a blocchi simmetrico con un blocco simmetrico con una lunghezza di blocco di 128 bit e supporto per chiavi di 128, 192 e 256 bit. I criteri di valutazione includevano la **sicurezza, l'efficienza computazionale, requisiti di memoria, idoneità hardware e software e flessibilità**. In una prima fase di valutazione sono stati accettati 15 algoritmi proposti. In un secondo momento si è poi ristretto il campo a 5 algoritmi. Il NIST ha completato il processo di valutazione e ha pubblicato lo standard finale come FIPS PUB 197 (Advanced Encryption Standard, novembre 2001).

2.1.3 Cifrari a flusso

Un cifrario a blocchi elabora l'input un blocco di elementi alla volta, producendo un blocco di output per ogni blocco di input. Un cifrario a flusso elabora gli elementi in ingresso in modo continuo, producendo un elemento in uscita alla volta, man mano che procede. Sebbene i cifrari a blocchi siano molto più comuni, ci sono alcune applicazioni in cui un cifrario a flusso è più appropriato. Un tipico cifrario a flusso critta il testo in chiaro un byte alla volta, anche se un cifrario a flusso può essere progettato per operare su un bit alla volta o su unità di dimensioni piu' grandi di un byte alla volta.

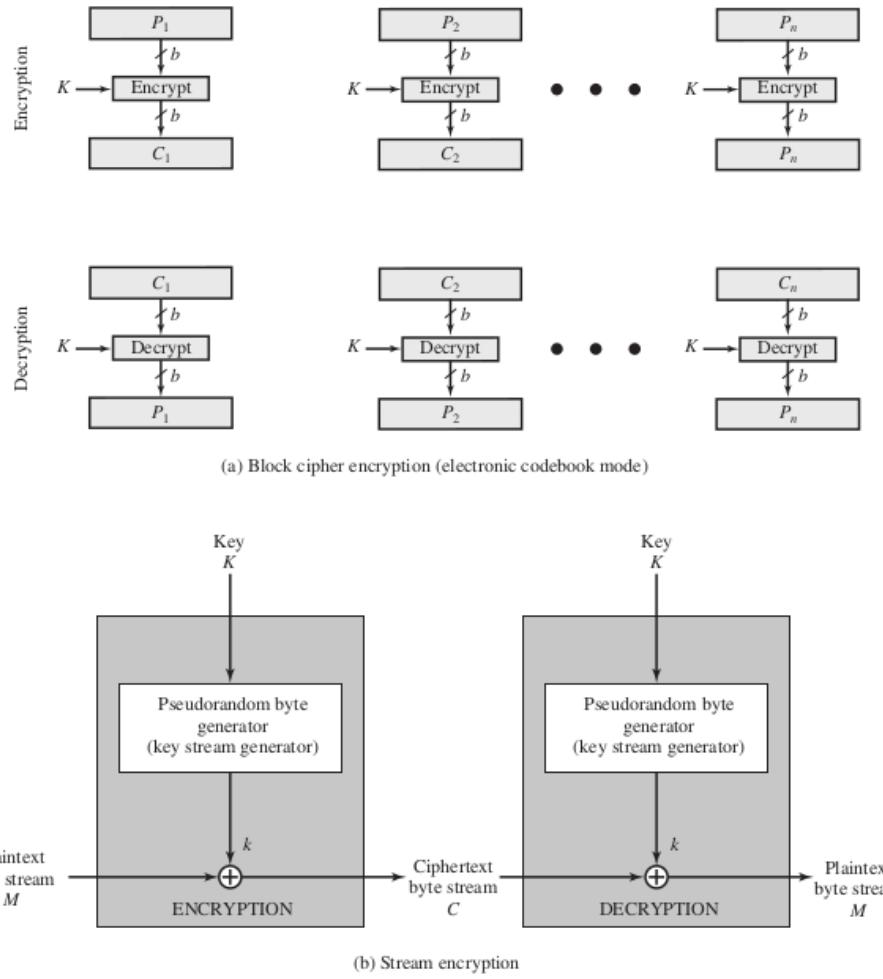


Figure 2.2 Types of Symmetric Encryption

La Figura 2.2b è un diagramma rappresentativo della struttura di un cifrario a flusso.

In questa struttura, una chiave viene immessa in un generatore di bit pseudorandom che produce un flusso di numeri a 8 bit che sono apparentemente casuali. Un flusso

pseudorandom è un flusso imprevedibile senza conoscere la chiave di ingresso e che ha un carattere apparentemente casuale. L'output del generatore, chiamato flusso di chiavi, viene combinato un byte alla volta con il flusso di testo in chiaro utilizzando l'operazione di OR esclusivo bitwize (XOR). Con un generatore di numeri pseudorandom correttamente progettato, un cifrario a flusso può essere sicuro quanto un cifrario a blocchi di lunghezza analoga. Il vantaggio principale di un cifrario a flusso è che i cifrari a flusso sono quasi sempre più veloci e utilizzano molto meno codice rispetto ai cifrari a blocco in piu' è possibile riutilizzare le chiavi. Per applicazioni che richiedono la crittografia/decrittografia di un flusso di dati, come ad esempio un canale di comunicazione dati o un browser/Web. Un cifrario a flusso potrebbe essere l'alternativa migliore. Per le applicazioni che trattano blocchi di dati, come il trasferimento di file, e-mail e database, i cifrari a blocchi possono essere più appropriati. Tuttavia, entrambi i tipi di di cifratura possono essere utilizzati praticamente in qualsiasi applicazione.

2.2 Autenticazione basata su messaggio e funzioni Hash

La crittografia protegge sia dagli **attacchi passivi** (intercettazioni) che dagli **attacchi attivi** (falsificazione di dati e transazioni).

La protezione contro tali attacchi è nota come autenticazione dei messaggi o dei dati. Un messaggio, un file, un documento o un'altra raccolta di dati si dice autentico quando è genuino e proviene dalla sua presunta fonte. L'autenticazione dei messaggi o dei dati è una procedura che consente alle parti comunicanti di verificare **che i messaggi ricevuti o memorizzati siano autentici**.

I due aspetti importanti per l'autenticazione sono:

1. Verificare che il contenuto del messaggio non sia stato alterato
2. La fonte deve essere autentica

Si può anche voler verificare la tempestività di un messaggio (che non sia stato artificialmente ritardato e riprodotto) rispetto ad altri messaggi che scorrono tra due parti.

2.2.1 Autenticazione usando la crittografia simmetrica

Sembra possibile eseguire l'autenticazione semplicemente utilizzando la crittografia simmetrica. Se assumiamo che solo il mittente e il destinatario condividano una chiave (come dovrebbe essere), allora solo il mittente autentico sarebbe in grado di criptare con successo un messaggio per l'altro partecipante, a patto che il destinatario sia in grado di riconoscere un messaggio valido.

Inoltre, se il messaggio include un codice di rilevazione degli errori e un numero di sequenza, il destinatario ha la certezza che non sono state apportate alterazioni e che il che la sequenza è corretta. Se il messaggio include anche un timestamp, il destinatario è sicuro che il messaggio non è stato alterato e che la sequenza è corretta ed anche che il messaggio non ha subito ritardi superiori a quelli normalmente previsti per il transito in rete.

In realtà, la crittografia simmetrica da sola non è uno strumento adatto per l'autenticazione dei dati.

Per fare un semplice esempio, nella modalità di crittografia BCE, se un aggressore riordina i blocchi di testo cifrato, ogni blocco verrà comunque decifrato con successo. Tuttavia, il riordino può alterare il significato della sequenza di dati complessiva. Anche se i numeri di sequenza possano essere utilizzati a un certo livello (ad esempio, ogni pacchetto IP), di solito non è il caso che un numero di sequenza separato sia associato a ciascun blocco b-bit di testo in chiaro. Pertanto, il riordino dei blocchi è una minaccia.

2.2.2 Autenticazione dei messaggi senza crittografia dei messaggi

In tutti questi approcci, un tag di autenticazione viene generato e aggiunto a ogni messaggio per la trasmissione. Il messaggio stesso non è criptato e può essere letto a destinazione indipendentemente dalla funzione di autenticazione. Poiché gli approcci discussi in questa sezione non cifrano il messaggio e non garantiscono la riservatezza del messaggio. Come già detto, la crittografia dei messaggi di per sé come si è detto, non fornisce una forma sicura di autenticazione. Tuttavia, è possibile combinare l'autenticazione e la riservatezza in un unico algoritmo crittografando un messaggio e il suo tag di autenticazione. In genere, tuttavia, l'autenticazione dei messaggi viene fornita come una funzione separata dalla crittografia del messaggio e suggerisce tre situazioni in cui in cui l'autenticazione dei messaggi senza riservatezza è migliore:

- Esistono diverse applicazioni in cui lo stesso messaggio viene trasmesso a più destinazioni.

Due esempi sono la notifica agli utenti che la rete non è più disponibile e la trasmissione di un messaggio a più destinazioni. Ovviamente è più economico e più affidabile avere una sola destinazione responsabile per il controllo dell'autenticità. Pertanto, il messaggio deve essere trasmesso in chiaro con un tag di autenticazione associato al messaggio. Il sistema responsabile esegue l'autenticazione. Se si verifica una violazione, gli altri sistemi di destinazione vengono avvisati da un allarme generale.

- Un altro scenario possibile è uno scambio in cui una parte ha un carico pesante e non può permettersi il tempo di decifrare tutti i messaggi in arrivo.

l'autenticazione viene effettuata in modo selettivo, scegliendo a caso i messaggi da controllare.

- L'autenticazione in chiaro è un servizio interessante.

Il programma può essere eseguito senza doverlo decifrare ogni volta, che sarebbe uno spreco di risorse del processore. Tuttavia, se un'etichetta di autenticazione del messaggio al programma, questo potrebbe essere controllato ogni volta che si richiede una garanzia di integrità del programma.

Pertanto, sia l'autenticazione che la crittografia possono essere utilizzate per soddisfare le esigenze di sicurezza.

Codice di autenticazione del messaggio E' una tecnica di autenticazione che prevede l'uso di una chiave segreta per generare un piccolo blocco di dati, noto come codice di autenticazione del messaggio, che viene aggiunto al messaggio. Questa tecnica presuppone che due parti comunicanti, ad esempio A e B, condividano una chiave segreta comune K_{AB} . Quando A ha un messaggio da inviare a B, calcola il codice di autenticazione del messaggio come funzione complessa del messaggio e della chiave: $MAC_M = F(K_{AB}, M)$.³ Il messaggio più il codice vengono trasmessi al destinatario previsto.

Il destinatario esegue lo stesso calcolo sul messaggio ricevuto, utilizzando la stessa chiave segreta, per generare un nuovo codice di autenticazione del messaggio. Il codice ricevuto viene confrontato con quello calcolato (cfr. Figura 2.3). Se assumiamo che solo il destinatario e il mittente conoscano l'identità della chiave segreta, e se il codice ricevuto corrisponde a quello calcolato, allora il codice ricevuto corrisponde al codice calcolato allora:

1. Il destinatario ha la certezza che il messaggio non è stato alterato. Se un aggressore altera il messaggio ma non il codice, il calcolo del codice da parte del destinatario sarà diverso da quello ricevuto. Poiché si presume che l'aggressore non conosca la chiave segreta, l'aggressore non può alterare il codice in modo da far corrispondere alle alterazioni del messaggio.
2. Il destinatario ha la certezza che il messaggio provenga dal presunto mittente. Poiché nessun altro conosce la chiave segreta, nessun altro potrebbe preparare un messaggio con un codice adeguato.
3. Se il messaggio include un numero di sequenza (come nel caso di X.25, HDLC e TCP), il destinatario può essere certo che il messaggio provenga dal presunto mittente perché un aggressore non può alterare il numero di sequenza.

Il DES o l'AES vengono utilizzati per generare una versione crittografata del messaggio, e alcuni di questi algoritmi vengono utilizzati per la creazione di un codice.

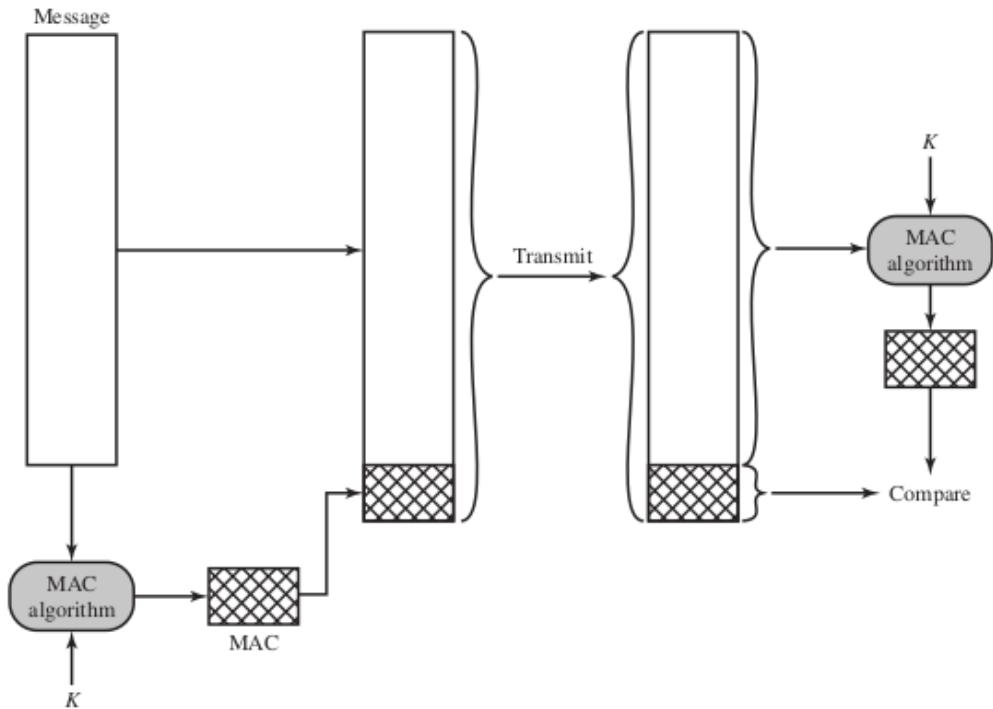


Figure 2.3 Message Authentication Using a Message Authentication Code (MAC)

Funzione di hash a una via Un’alternativa al codice di autenticazione dei messaggi è la funzione hash unidirezionale. Come nel caso del codice di autenticazione dei messaggi, una funzione di hash accetta in ingresso un messaggio M di dimensioni variabili e produce in uscita un digest del messaggio $H(M)$ di dimensioni fisse (vedere Figura 2.4). In genere, il messaggio viene imbottito fino a un multiplo intero di una certa lunghezza fissa (ad esempio, 1024 bit) e l’imbottitura include il valore della lunghezza del messaggio originale in bit. Il campo della lunghezza è una misura di sicurezza per di sicurezza per aumentare la difficoltà per un aggressore di produrre un messaggio alternativo con lo stesso valore di hash.

A differenza del MAC, una funzione di hash non riceve in ingresso una chiave segreta. La Figura 2.5 illustra tre modi in cui il messaggio può essere autenticato utilizzando una funzione hash. Il digest del messaggio può essere crittografato utilizzando la crittografia simmetrica (vedere Figura 2.5a); se si presume che solo il mittente e il destinatario condividono la chiave di crittografia, l’autenticità è garantita. Il message digest può anche essere crittografato con la crittografia a chiave pubblica (cfr. Figura 2.5b), come illustrato nella Sezione 2.3. L’approccio a chiave pubblica chiave pubblica presenta due vantaggi:

1. Fornisce una firma digitale
2. L’autenticazione del messaggio e non richiede la distribuzione delle chiavi alle parti comunicanti

Questi due approcci hanno un vantaggio rispetto a quelli che criptano l'intero messaggio, in quanto richiedono meno calcoli. Ma un approccio ancora più comune è l'uso di una tecnica che evita del tutto la crittografia. Diverse ragioni per questo interesse sono evidenziate in:

- 1. Il software di crittografia è piuttosto lento.**

Anche se la quantità di dati da crittografia per ogni messaggio è piccola, può esserci un flusso costante di messaggi in entrata e in uscita da un sistema.

- 2. I costi dell'hardware di crittografia non sono trascurabili.**

Sono disponibili implementazioni su chip a basso costo di DES e AES, ma il costo aumenta se tutti i nodi di una rete devono avere questa capacità.

- 3. L'hardware di crittografia è ottimizzato per le grandi dimensioni dei dati.**

Per piccoli blocchi di dati, un'alta percentuale del tempo viene spesa per l'overhead di inizializzazione/invocazione.

- 4. Un algoritmo di crittografia può essere protetto da un brevetto.**

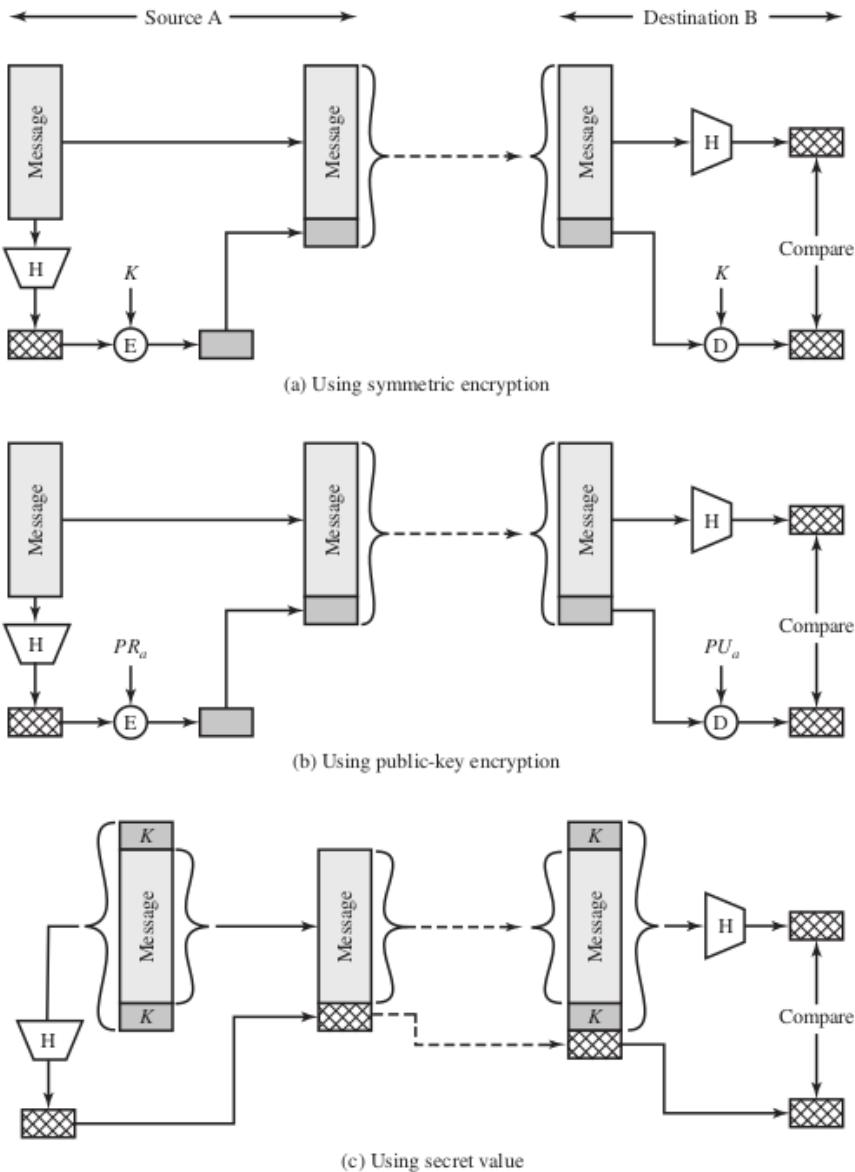


Figure 2.5 Message Authentication Using a One-Way Hash Function

La Figura 2.5c mostra una tecnica che utilizza una funzione hash ma non la crittografia per l'autenticazione dei messaggi. Questa tecnica, nota come MAC hash a chiave, presuppone che due parti comunicanti, ad esempio A e B, condividano una chiave segreta comune K. Questa chiave segreta viene incorporata nel processo di generazione di un codice hash. Nell'approccio illustrato nella Figura 2.5c, quando A ha un messaggio da inviare a B, calcola la funzione di hash sulla concatenazione della chiave segreta e del messaggio: $M_{DM} = H(K||M||K)$. Quindi invia $[M M_{DM}]$ a B. Poiché B possiede K, può ricalcolare $H(K||M||K)$ e verificare M_{DM} . Dato che la chiave segreta non viene inviata,

non dovrebbe essere possibile per un attaccante un aggressore di modificare un messaggio intercettato. Finché la chiave segreta rimane segreta, non dovrebbe essere possibile per un aggressore generare un messaggio falso. Si noti che la chiave segreta viene utilizzata sia come prefisso che come suffisso del messaggio. Se la chiave segreta è usata solo come prefisso o solo come suffisso, lo schema è meno sicuro.

2.2.3 Altre applicazioni delle funzioni hash

Ecco altri due esempi di applicazioni sicure delle funzioni hash:

- **Password**

In parole povere, quando un utente inserisce una password, l'hash di tale password viene confrontato con il valore di hash memorizzato per la verifica. Questa applicazione richiede resistenza alla preimmagine e forse una seconda resistenza alla preimmagine.

- **Rilevamento delle intrusioni**

Memorizzare il valore di hash di un file, $H(F)$, per ogni file su un sistema e proteggere i valori di hash ($H(F)$). Un sistema e proteggere i valori di hash (ad esempio, su un'unità bloccata in scrittura o su un disco ottico disco ottico write-once che viene tenuto al sicuro). È possibile determinare in seguito se un file è stato modificato calcolando nuovamente $H(F)$. Un intruso dovrebbe modificare F senza modificare $H(F)$.

2.3 Crittografia a chiave pubblica

2.3.1 Struttura della crittografia a chiave pubblica

Gli algoritmi a chiave pubblica sono basati su funzioni matematiche piuttosto che su semplici operazioni su schemi di bit, come quelle utilizzate negli algoritmi di crittografia simmetrica. Inoltre, la crittografia a chiave pubblica è asimmetrica e prevede l'uso di due chiavi separate, a differenza della crittografia simmetrica che utilizza una sola chiave.

L'uso di due chiavi ha profonde conseguenze nelle aree della riservatezza, della distribuzione delle chiavi e dell'autenticazione.

Prima di procedere, è necessario menzionare alcune idee sbagliate comuni sulla crittografia a chiave pubblica. Una di queste è che la crittografia a chiave pubblica sia più sicura rispetto alla crittografia simmetrica.

In realtà, la sicurezza di qualsiasi schema di crittografia dipende da:

1. **La lunghezza della chiave**
2. **il lavoro computazionale necessario per di calcolo necessario per decifrare un cifrario**

In linea di principio non c'è nulla nella crittografia simmetrica o della crittografia a chiave pubblica che ne renda una superiore all'altra dal punto di vista di resistenza agli attacchi. Una seconda convinzione errata è che la crittografia a chiave pubblica sia una tecnica di uso generale che ha reso obsoleta la crittografia simmetrica. Al contrario, a causa dell'overhead computazionale degli attuali schemi, non sembra che la crittografia simmetrica venga abbandonata. Infine, si ha l'impressione che la distribuzione delle chiavi

sia banale quando si usa la crittografia a chiave pubblica, rispetto a quella a chiave simmetrica. Per la distribuzione delle chiavi a chiave pubblica, è necessaria una qualche forma di protocollo, che spesso coinvolge un agente centrale, e le procedure non sono più semplici o più efficienti di quelle richieste per la crittografia simmetrica.

Uno schema di crittografia a chiave pubblica ha sei ingredienti (vedi Figura 2.6a):

- **Testo in chiaro**

Si tratta del messaggio o dei dati leggibili che vengono inseriti nell'algoritmo come input.

- **Algoritmo di crittografia**

L'algoritmo di crittografia esegue varie trasformazioni sul testo in chiaro.

- **Chiave Pubblica e Privata**

Si tratta di una coppia di chiavi selezionate in modo che se una viene usata per la crittografia, l'altra viene usata per la decrittografia. Le trasformazioni esatte eseguite dall'algoritmo di crittografia dipendono dalla chiave pubblica o privata fornita in ingresso.

- **Testo cifrato**

È il messaggio criptato prodotto in uscita. Dipende dal testo in chiaro e dalla chiave. Per un dato messaggio, due chiavi diverse produrranno due cifrari diversi.

- **Algoritmo di decrittazione**

Questo algoritmo accetta il testo in chiaro e la chiave corrispondente e produce il testo in chiaro originale.

Come suggeriscono i nomi, la chiave pubblica della coppia è resa pubblica e può essere utilizzata da altri, mentre la chiave privata è nota solo al suo proprietario. Un algoritmo crittografico di uso generale si basa su una chiave per la crittografia e su una chiave diversa ma correlata per la per la decifrazione.

Le fasi essenziali sono le seguenti:

1. Ogni utente genera una coppia di chiavi da utilizzare per la crittografia e la decrittografia dei messaggi.
2. Ogni utente inserisce una delle due chiavi in un registro pubblico o in un altro file accessibile.
3. Se Bob desidera inviare un messaggio privato ad Alice, Bob critta il messaggio utilizzando la chiave pubblica di Alice.
4. Quando Alice riceve il messaggio, lo decifra utilizzando la sua chiave privata. Nessun altro può decifrare il messaggio perché solo Alice conosce la chiave privata di Alice.

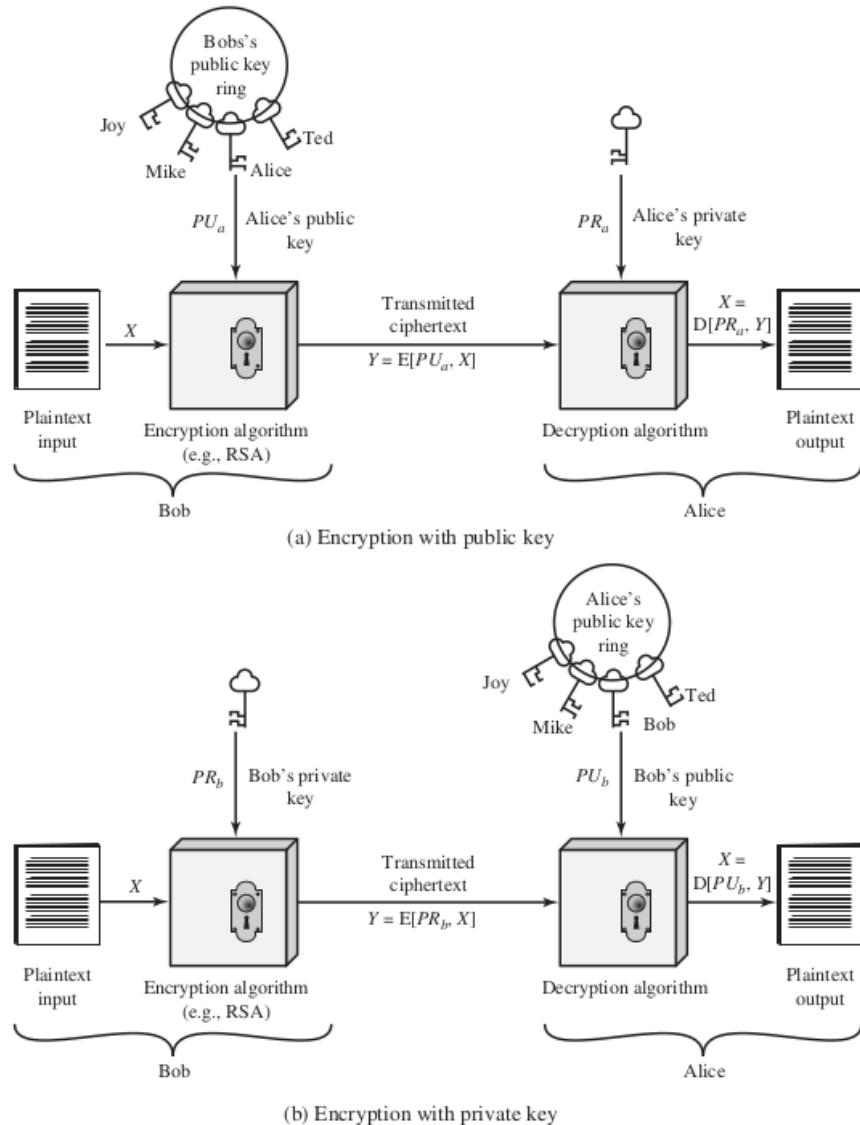


Figure 2.6 **Public-Key Cryptography**

Con questo approccio, tutti i partecipanti hanno accesso alle chiavi pubbliche, mentre le chiavi private sono generate localmente da ogni partecipante e quindi non devono mai essere distribuite. Finché un utente protegge la propria chiave privata, la comunicazione in entrata è sicura. In qualsiasi momento, un utente può cambiare la chiave privata e pubblicare la nuova chiave pubblica che sostituisce la vecchia chiave pubblica.

In questo schema, un utente critta i dati utilizzando la propria chiave privata. Chiunque conosca la corrispondente chiave pubblica sarà in grado di decifrare il messaggio. Lo schema della Figura 2.6a è finalizzato a garantire la riservatezza. Solo il destinatario previsto deve essere in grado di decifrare il testo cifrato perché solo il destinatario previsto

è in possesso di una chiave privata richiesta.

Lo schema della Figura 2.6b è finalizzato a fornire l'autenticazione e/o l'integrità dei dati. Se un utente è in grado di recuperare il testo in chiaro dal testo cifrato di Bob utilizzando la chiave pubblica di Bob, ciò indica che solo Bob può aver cifrato il testo in chiaro, fornendo così l'autenticazione. Inoltre, solo Bob sarebbe in grado di modificare il testo in chiaro perché solo Bob può cifrare il testo in chiaro con la chiave privata di Bob.

2.3.2 Applicazioni dei sistemi crittografici a chiave pubblica

I sistemi a chiave pubblica sono caratterizzati dall'uso di un algoritmo di tipo crittografico con due chiavi, una privata e una pubblica. A seconda dell'applicazione, il mittente utilizza la chiave privata del mittente o la chiave pubblica del destinatario, o entrambe, per eseguire un tipo di funzione crittografica. In termini generali, possiamo classificare l'uso dei sistemi crittografici a chiave pubblica in tre categorie: firma digitale, distribuzione di chiavi simmetriche e crittografia di chiavi segrete.

2.3.3 Requisiti per la crittografia a chiave pubblica

Il sistema crittografico illustrato nella Figura 2.6 dipende da un algoritmo crittografico basato su due chiavi correlate. Diffie e Hellman hanno postulato questo sistema senza dimostrare l'esistenza di tali algoritmi.

Tuttavia, hanno definito le condizioni che tali algoritmi devono soddisfare:

1. È computazionalmente facile per una parte B generare una coppia (chiave pubblica PU_b , chiave privata PR_b).
2. È computazionalmente facile per un mittente A, conoscendo la chiave pubblica e il messaggio da crittografare, M, per generare il corrispondente testo cifrato:

$$C = E(PU_b, M)$$

3. È computazionalmente facile per il destinatario B decrittare il testo cifrato risultante utilizzando la chiave privata per recuperare il messaggio originale:

$$M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$$

4. È computazionalmente impossibile per un avversario, conoscendo la chiave pubblica, PU_b , determinare la chiave privata, PR_b .
5. È computazionalmente impossibile per un avversario, conoscendo la chiave pubblica, PU_b , e un testo cifrato, C, per recuperare il messaggio originale, M. Possiamo aggiungere un sesto requisito che, sebbene utile, non è necessario per tutte le applicazioni a chiave pubblica.
6. Una delle due chiavi correlate può essere utilizzata per la crittografia e l'altra per la decrittografia.

$$M = D[PU_b, E(PR_b, M)] = D[PR_b, E(PU_b, M)]$$

Table 2.3 Applications for Public-Key Cryptosystems

Algorithm	Digital Signature	Symmetric Key Distribution	Encryption of Secret Keys
RSA	Yes	Yes	Yes
Diffie–Hellman	No	Yes	No
DSS	Yes	No	No
Elliptic Curve	Yes	Yes	Yes

2.3.4 Algoritmi di crittografia asimmetrica

RSA Lo schema RSA da allora regna sovrano come l'approccio alla crittografia a chiave pubblica. RSA è un cifrario a blocchi in cui il plaintext e il testo in chiaro e il testo cifrato sono numeri interi compresi tra 0 e $n - 1$ per qualche n . Per l'uso di RSA, è necessario utilizzare chiavi di dimensioni maggiori. Attualmente, una chiave di 1024 bit (circa 300 cifre decimali) è considerata abbastanza forte per quasi tutte le applicazioni.

Accordo di chiave Diffie-Hellman Il primo algoritmo a chiave pubblica pubblicato è apparso nell'articolo di Diffie e Hellman che ha definito la crittografia a chiave pubblica e viene generalmente chiamato scambio di chiavi Diffie-Hellman, o accordo di chiavi.

Numerosi prodotti commerciali utilizzano questa tecnica di scambio di chiavi. Lo scopo dell'algoritmo è quello di permettere a due utenti di raggiungere in modo sicuro un accordo su un segreto condiviso che può essere usato come chiave segreta per la successiva crittografia simmetrica dei messaggi. L'algoritmo stesso si limita allo scambio delle chiavi.

Standard di firma digitale L'Istituto nazionale per gli standard e la tecnologia (NIST) ha pubblicato questo algoritmo come FIPS PUB 186 (Digital Signature Standard (DSS)). Il DSS fa uso di SHA-1 e presenta una nuova tecnica di firma digitale, l'algoritmo di firma digitale (DSA). Il DSS utilizza un algoritmo progettato per fornire solo i dati di accesso al sistema. A differenza dell'RSA, non può essere utilizzato per la crittografia o lo scambio di chiavi.

Crittografia a curva ellittica La stragrande maggioranza dei prodotti e degli standard che utilizzano la crittografia a chiave pubblica per la crittografia e la firma digitale utilizzano RSA. La lunghezza dei bit per l'utilizzo sicuro di RSA è aumentata negli ultimi anni e ciò ha comportato un carico di elaborazione più pesante. Questo ha comportato un carico di elaborazione maggiore per le applicazioni che utilizzano RSA. Recentemente, un sistema concorrente ha iniziato a sfidare RSA: la crittografia a curve ellittiche (ECC). L'ECC è già presente nelle iniziative di standardizzazione, tra cui l'IEEE (Istituto Elettrico). L'attrattiva principale dell'ECC rispetto all'RSA è che sembra offrire la stessa sicurezza per una dimensione di bit molto più piccola riducendo così l'overhead di elaborazione. D'altra parte, sebbene la teoria dell'ECC esista da tempo, è solo di recente che hanno cominciato ad apparire dei prodotti e che c'è stato un interesse crittoanalitico per la ricerca di punti deboli. Pertanto, il livello di fiducia nell'ECC non è ancora così alto come quello di RSA.

2.4 Firme digitali e gestione delle chiavi

2.4.1 Firme digitali

La crittografia a chiave pubblica può essere utilizzata per l'autenticazione con una tecnica nota come firma digitale. Digital Signature Standard (DSS), definiscono la firma digitale come segue: Il risultato di una trasformazione crittografica di dati che, se correttamente implementata, fornisce un meccanismo per la verifica dell'origine l'autenticazione, l'integrità dei dati e il non ripudio del firmatario. Pertanto, una firma digitale è un modello di bit dipendente dai dati, generato da un agente in funzione di un file, di un messaggio o di un'altra forma di blocco di dati.

Il FIPS 186-4 specifica l'uso di uno dei tre algoritmi di firma digitale:

- Algoritmo di firma digitale (DSA): L'algoritmo originale approvato dal NIST, che si basa sulla difficoltà di calcolo dei logaritmi discreti.
- Algoritmo di firma digitale RSA: Basato sull'algoritmo a chiave pubblica RSA.
- Algoritmo di firma digitale a curva ellittica (ECDSA): basato sulla crittografia a curva ellittica.

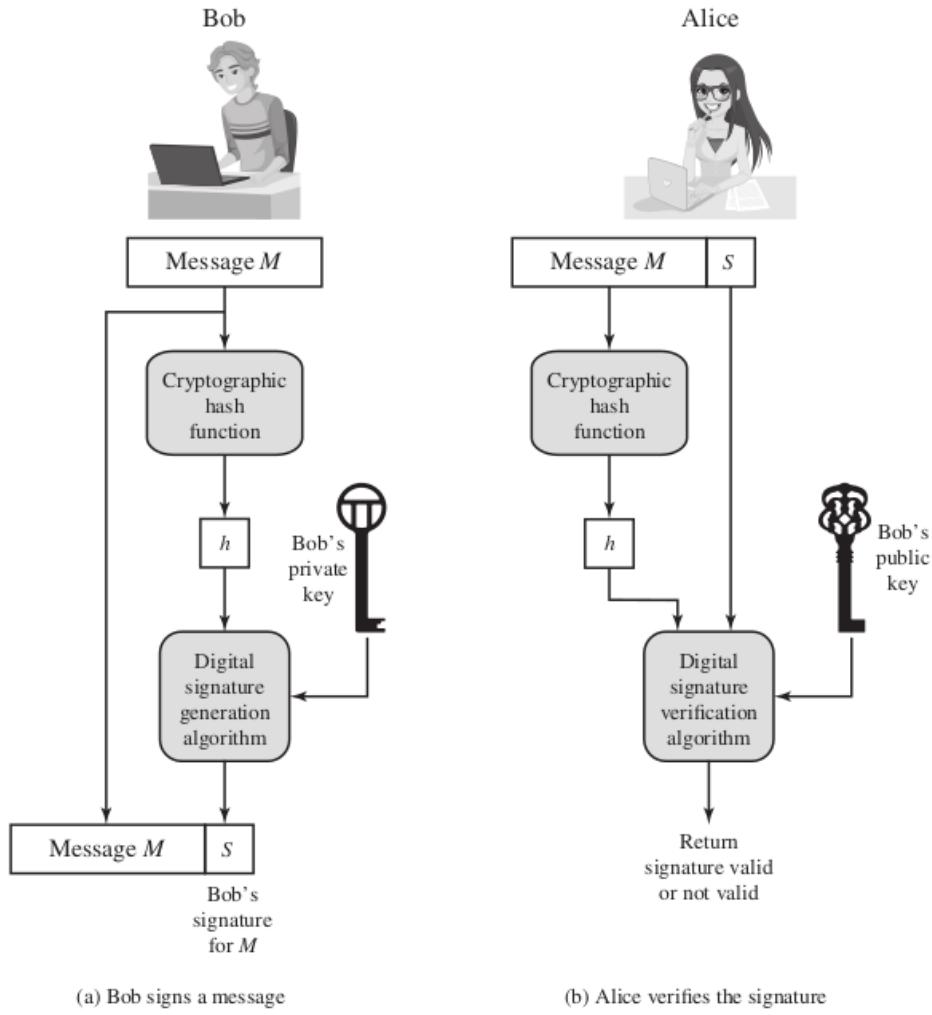


Figure 2.7 Simplified Depiction of Essential Elements of Digital Signature Process

La Figura 2.7 è un modello generico del processo di creazione e utilizzo delle firme digitali.

Tutti gli schemi di firma digitale previsti dal FIPS 186-4 hanno questa struttura.

Supponiamo che Bob voglia inviare un messaggio ad Alice. Sebbene non sia importante che il messaggio segreto, Bob vuole che Alice abbia la certezza che il messaggio provenga da lui. A tal fine, Bob utilizza una funzione di hash sicura, come SHA-512, per generare un valore di hash per il messaggio. Questo valore di hash, insieme alla chiave privata di Bob, serve come input per un algoritmo di generazione della firma digitale che produce un breve blocco che funge da firma digitale. Bob invia il messaggio con la firma allegata.

Quando Alice riceve il messaggio e la firma:

1. calcola un valore di hash per il messaggio
 2. fornisce il valore di hash per il messaggio e la chiave pubblica di Bob come input a un algoritmo di verifica della firma digitale.

Se l'algoritmo restituisce il risultato che la firma è valida, Alice ha la certezza che il messaggio è stato firmato da Bob. Nessun altro possiede la chiave privata di Bob e quindi nessun altro può aver creato una firma che possa essere verificata per questo messaggio con la chiave pubblica di Bob. Inoltre, è impossibile alterare il messaggio senza accedere alla chiave privata di Bob, quindi il messaggio è autenticato sia in termini di origine che di integrità dei dati. La firma digitale non garantisce la riservatezza. Cioè, il messaggio inviato è al sicuro da alterazioni, ma non da intercettazioni. Questo è evidente nel caso di una firma basata su una parte del messaggio, perché il resto del messaggio viene trasmesso in chiaro. Anche nel caso di una crittografia completa, non c'è alcuna protezione della riservatezza, perché qualsiasi osservatore può decifrare il messaggio utilizzando la chiave pubblica del mittente.

2.4.2 Certificati a chiave pubblica

Esiste un algoritmo a chiave pubblica ampiamente accettato, come RSA, ogni partecipante può inviare la propria chiave pubblica a qualsiasi altro partecipante o trasmettere la chiave alla comunità in generale. Sebbene questo approccio sia conveniente, ha una grande debolezza. Chiunque può falsificare tale annuncio pubblico. Cioè, un utente potrebbe fingere di essere Bob e inviare una chiave pubblica a un altro partecipante o trasmettere tale chiave pubblica. Finché Bob non scopre la falsificazione e avvisa gli altri partecipanti, il falsario è in grado di leggere tutti i messaggi cifrati destinati a Bob e può utilizzare le chiavi falsificate per l'autenticazione.

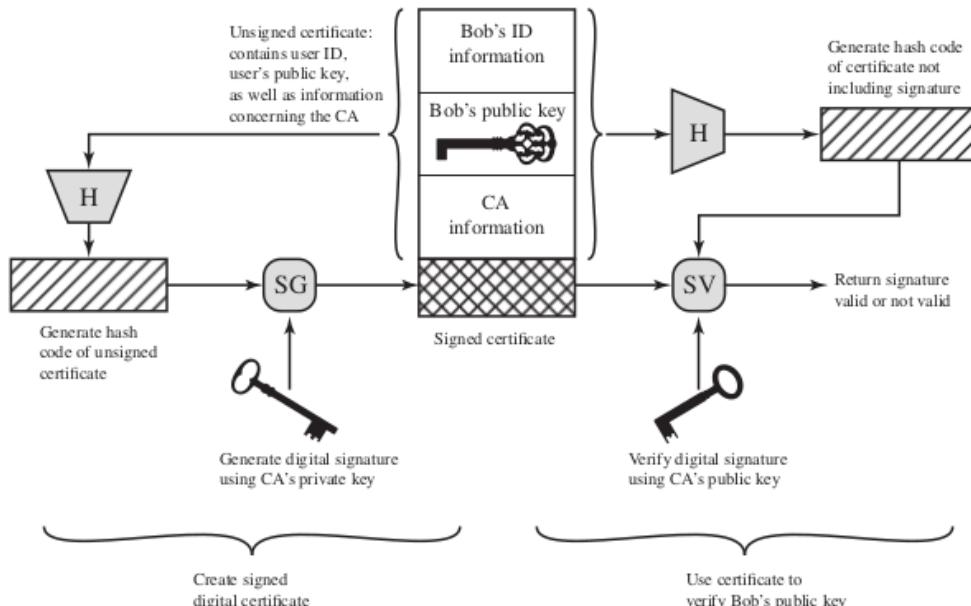


Figure 2.8 Public-Key Certificate Use

La soluzione a questo problema è il certificato a chiave pubblica. In sostanza, un certificato è costituito da una chiave pubblica più un ID utente del proprietario della chiave, con l'intero blocco firmato da una terza parte fidata. Il certificato include anche alcune informazioni sulla terza parte e un'indicazione del periodo di validità del certificato. In genere, la terza parte è un'autorità di certificazione (CA) di cui si fida la comunità degli utenti, come un'agenzia governativa o un'istituzione finanziaria. Un utente può presentare la propria chiave pubblica all'autorità in modo sicuro e ottenere un certificato firmato. L'utente può quindi pubblicare il certificato. Chiunque abbia bisogno della chiave pubblica di questo utente può ottenere il certificato e verificarne la validità attraverso la firma di fiducia allegata. La Figura 2.8 illustra il processo. Le fasi principali possono essere riassunte come segue:

1. Il software utente (client) crea una coppia di chiavi: una pubblica e una privata.
2. Il client prepara un certificato non firmato che include l'ID utente e la chiave pubblica dell'utente.
3. L'utente fornisce il certificato non firmato a una CA in modo sicuro. Questo potrebbe un incontro faccia a faccia, l'uso di un'e-mail registrata o un modulo Web con verifica via e-mail.
4. La CA crea una firma come segue:
 - La CA utilizza una funzione hash per calcolare il codice hash del certificato non firmato.
Una funzione di hash è una funzione che mappa un blocco di dati o un messaggio di lunghezza variabile in un valore di lunghezza fissa, chiamato codice hash.
 - La CA genera la firma digitale utilizzando la chiave privata della CA e un algoritmo di generazione della firma.
5. La CA appone la firma al certificato non firmato per creare un certificato firmato.
6. La CA restituisce al cliente il certificato firmato.
7. Ogni utente può verificare la validità del certificato come segue:
 - L'utente calcola il codice hash del certificato (esclusa la firma).
 - L'utente verifica la firma digitale utilizzando la chiave pubblica della CA e l'algoritmo di verifica della firma. L'algoritmo restituisce un risultato di firma valida o non valida.

Uno schema è diventato universalmente accettato per la formattazione dei certificati a chiave pubblica, lo standard X.509. I certificati X.509 sono utilizzati nella maggior parte delle applicazioni per la sicurezza di rete, tra cui applicazioni di sicurezza di rete, tra cui IP Security (IPsec), Transport Layer Security (TLS), Secure Shell (SSH) e Secure Shell e Secure/Multipurpose Internet Mail Extension (S/MIME). La maggior parte di queste esamineremo la maggior parte di queste applicazioni nella quinta parte.

2.4.3 Scambio di chiavi simmetriche con crittografia a chiave pubblica

Con la crittografia simmetrica, un requisito fondamentale affinché due parti possano comunicare in modo sicuro è la condivisione di una chiave segreta. Supponiamo che Bob voglia creare un'applicazione di messaggistica che gli consenta di scambiare e-mail in modo sicuro con chiunque abbia accesso a Internet o a un'altra rete condivisa da entrambi.

Supponiamo che Bob voglia farlo utilizzando la crittografia simmetrica. Con la crittografia simmetrica, Bob e la sua corrispondente, diciamo Alice, devono trovare un modo per condividere una chiave segreta unica che nessun altro conosce. Come possono farlo? Se

Alice si trova nella stanza accanto a Bob, quest'ultimo potrebbe generare una chiave e scriverla su un foglio di carta o memorizzarla su un disco o una chiavetta e consegnarla ad Alice. Ma se Alice si trova dall'altra parte del continente o del mondo, cosa può fare Bob? Potrebbe criptare la chiave con la crittografia simmetrica e inviarla via e-mail ad Alice, ma ciò significa che Bob e Alice devono condividere una chiave segreta per criptare la nuova chiave segreta. Inoltre, Bob e chiunque altro utilizzi questo nuovo pacchetto di e-mail si trova ad affrontare lo stesso problema con ogni potenziale corrispondente: Ogni coppia di corrispondenti deve condividere una chiave segreta unica. Un approccio è l'uso dello scambio di chiavi Diffie-Hellman. Questo approccio è in effetti ampiamente utilizzato. Tuttavia, soffre dell'inconveniente che, nella sua forma più semplice, Diffie-Hellman non prevede l'autenticazione dei due interlocutori. Esistono esistono varianti di Diffie-Hellman che superano questo problema. Inoltre, esistono protocolli che utilizzano altri algoritmi a chiave pubblica che raggiungono lo stesso obiettivo.

2.4.4 Buste Digitali

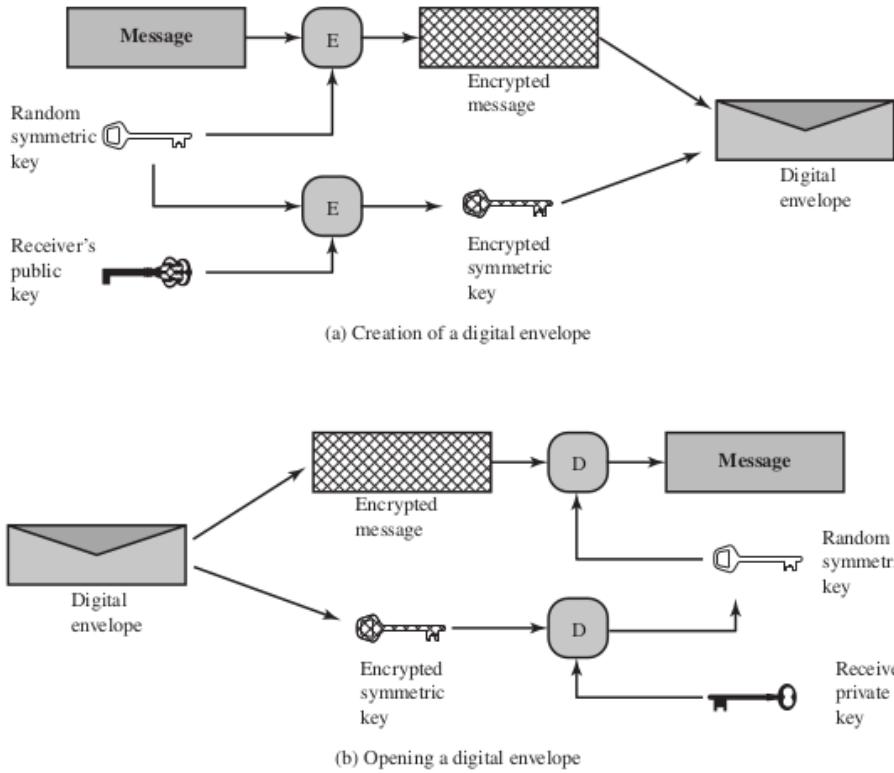


Figure 2.9 Digital Envelopes

Un'altra applicazione in cui la crittografia a chiave pubblica viene utilizzata per proteggere una chiave simmetrica è la busta digitale, che può essere usata per proteggere un messaggio senza dover prima fare in modo che mittente e destinatario abbiano la stessa chiave segreta. La tecnica è denominata busta digitale, che è l'equivalente di una busta sigillata contenente una lettera non firmata. L'approccio generale è illustrato nella Figura 2.9. Supponiamo che Bob voglia inviare un messaggio riservato ad Alice, ma che non condivida una chiave segreta simmetrica.

Bob esegue le seguenti operazioni:

1. Preparare un messaggio.
2. Generare una chiave simmetrica casuale che verrà utilizzata una sola volta.
3. Crittografare il messaggio utilizzando la crittografia simmetrica della chiave unica.
4. Crittografare la chiave una tantum utilizzando la crittografia a chiave pubblica con la chiave pubblica di Alice.

5. Allegare la chiave monouso crittografata al messaggio crittografato e inviarlo ad Alice.

Solo Alice è in grado di decifrare la chiave a tempo unico e quindi di recuperare il messaggio originale. Se Bob ha ottenuto la chiave pubblica di Alice per mezzo del certificato di certificato a chiave pubblica di Alice, Bob ha la certezza che si tratta di una chiave valida.

2.5 Applicazione pratica: Crittografia dei dati memorizzati

Uno dei principali requisiti di sicurezza di un sistema informatico è la protezione dei dati memorizzati. I meccanismi di sicurezza per garantire tale protezione includono il controllo degli accessi, di rilevamento delle intrusioni e schemi di prevenzione delle intrusioni. Ma oltre agli approcci tecnici, questi approcci possono diventare vulnerabili a causa di fattori umani. Ne elenchiamo qui alcuni esempi, basati su:

- Nel dicembre 2004, alcuni dipendenti della Bank of America hanno eseguito il backup e poi inviato al proprio centro dati di backup nastri contenenti i nomi dei clienti, gli indirizzi, i numeri di conto corrente e i numeri di previdenza sociale di 1,2 milioni di lavoratori pubblici iscritti a un conto con carta di credito. Nessuno dei dati era criptato.

I nastri non sono mai arrivati e non sono mai stati ritrovati. Purtroppo, questo metodo di backup e spedizione dei dati è fin troppo comune. Per fare un altro esempio, nell'aprile del 2005 Ameritrade ha incolpato il suo fornitore di spedizioni per aver perso un nastro di backup contenente informazioni non crittografate su 200.000 clienti.

- Nell'aprile del 2005, il San Jose Medical Group ha annunciato che qualcuno aveva fisicamente rubato uno dei suoi computer e un'altra parte di esso.
- Ci sono stati innumerevoli esempi di computer portatili persi negli aeroporti, rubati da una auto parcheggiata o presi mentre l'utente è lontano dalla sua scrivania. Se i dati sul disco non sono criptati, tutti i dati sono a disposizione del ladro.

Finché l'utente protegge la propria e non usa una password facilmente indovinabile, i file sono completamente protetti a protetti quando sono a riposo.

Alcuni approcci più recenti sono elencati in:

- **Dispositivo back-end**

Si tratta di un dispositivo hardware che si colloca tra i server e i sistemi di archiviazione e critta tutti i file. Questi dispositivi criptano i dati a una velocità alla velocità del cavo, con una latenza molto ridotta. Al contrario, il software di crittografia sui server e sui sistemi di archiviazione rallenta i backup.

- **Crittografia dei nastri basata su libreria**

È fornita da una scheda co-processore incorporata nell'unità nastro e integrato nell'unità nastro e nell'hardware della libreria a nastro. Il co-processore critta i dati utilizzando una chiave non leggibile configurata nella scheda. I nastri possono essere inviati fuori sede a una struttura che dispone dello stesso hardware dell'unità a nastro. La chiave può essere esportata tramite un' e-mail sicura o una piccola unità flash che viene trasportata in modo sicuro.

- **Crittografia di base dei dati di laptop e PC**

Diversi fornitori offrono prodotti software che forniscono una crittografia trasparente all'applicazione e all'utente. Alcuni prodotti crittografano tutti i file e le cartelle o una parte di essi. Altri prodotti, come altri prodotti, come BitLocker di Windows e FileVault di MacOS, crittografano un intero disco o un'immagine del disco o un'immagine del disco che si trova sul disco rigido dell'utente o su un dispositivo di archiviazione di rete.

Capitolo 3

Capitolo3

3.1 Principi dell'autenticazione digitale

L'autenticazione dell'utente comprende due funzioni.

1. L'utente si identifica al sistema presentando una credenziale, come l'ID utente.
2. Il sistema verifica l'utente attraverso lo scambio di informazioni di autenticazione.

Esempio. L'utente Alice Toklas potrebbe avere l'identificatore utente ABTOKLAS. Queste informazioni devono essere memorizzate su qualsiasi server o sistema di computer che Alice desidera utilizzare, e potrebbero essere note agli amministratori di sistema e ad altri utenti. Una tipica informazione di autenticazione associata a questo ID utente è una password, che è tenuta segreta (nota solo ad Alice e al sistema). Se nessuno è in grado di ottenere o indovinare la password di Alice, allora la combinazione di ID utente e password di Alice permette agli amministratori di impostare i permessi di accesso di Alice e di controllare la sua attività. Poiché l'ID di Alice non è segreto, gli utenti del sistema possono inviarle e-mail, ma poiché la sua password è segreta, nessuno può fingere di essere Alice.

- **L'identificazione** è il mezzo con cui un utente fornisce un'identità dichiarata al sistema.
- **L'autenticazione** dell'utente è il mezzo per stabilire la validità della dichiarazione.

NIST SP 800-63-3 (Digital Authentication Guideline, ottobre 2016). Definisce l'autenticazione digitale degli utenti come il processo per stabilire la fiducia nelle identità degli utenti che sono presentate elettronicamente a un sistema informativo. I sistemi possono usare l'identità autenticata per determinare se l'individuo autenticato è autorizzato ad eseguire particolari funzioni, come le transazioni su database o l'accesso alle risorse del sistema. In molti casi, l'autenticazione e la transazione, o altre funzioni autorizzate, avvengono attraverso una rete aperta come Internet.

3.1.1 Un modello per l'autenticazione digitale degli utenti

NIST SP 800-63-3 Definisce un modello generale per l'autenticazione dell'utente che coinvolge una serie di entità e procedure.

Il requisito iniziale per eseguire l'autenticazione dell'utente è che l'utente deve essere registrato nel sistema. La seguente è una tipica sequenza per la registrazione.

- **Un richiedente** si rivolge a un'autorità di registrazione (RA) per diventare un abbonato di un fornitore di servizi di credenziali (CSP).
- **La RA** è un'entità fidata che stabilisce e garantisce l'identità di un richiedente a un CSP
- **Il CSP** poi si impegna in uno scambio con l'abbonato. A seconda dei dettagli del sistema di autenticazione globale, il CSP rilascia una sorta di credenziale elettronica al abbonato.
- **La credenziale** è una struttura di dati che lega autorevolmente un'identità e attributi aggiuntivi a un token posseduto da un abbonato, e può essere verificata quando viene presentata al verificatore in una transazione di autenticazione.
- **Il token** potrebbe essere una chiave di crittografia o una password criptata che identifica l'abbonato. Il token può essere emesso dal CSP, generato direttamente dall'abbonato o fornito da una terza parte.

Il token e la credenziale possono essere usati in successivi eventi di autenticazione.

Table 3.1 Identification and Authentication Security Requirements (NIST SP 800-171)

Basic Security Requirements:	
1	Identify information system users, processes acting on behalf of users, or devices.
2	Authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.
Derived Security Requirements:	
3	Use multifactor authentication for local and network access to privileged accounts and for network access to non-privileged accounts.
4	Employ replay-resistant authentication mechanisms for network access to privileged and non-privileged accounts.
5	Prevent reuse of identifiers for a defined period.
6	Disable identifiers after a defined period of inactivity.
7	Enforce a minimum password complexity and change of characters when new passwords are created.
8	Prohibit password reuse for a specified number of generations.
9	Allow temporary password use for system logons with an immediate change to a permanent password.
10	Store and transmit only cryptographically-protected passwords.
11	Obscure feedback of authentication information.

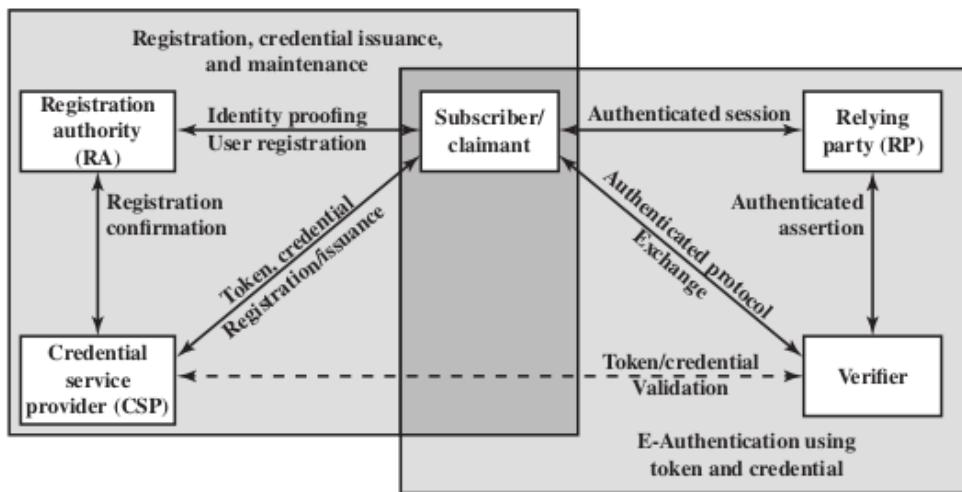


Figure 3.1 The NIST SP 800-63-3 E-Authentication Architectural Model

Una volta che un utente è registrato come abbonato, l'effettivo processo di autenticazione può avvenire tra l'abbonato e uno o più sistemi che eseguono l'autenticazione e, successivamente, l'autorizzazione. La parte che deve essere autenticata è chiamata richiedente e la parte che verifica tale identità è chiamata verificatore. Quando un richiedente dimostra con successo il possesso e il controllo di un token a un verificatore attraverso un protocollo di autenticazione, il verificatore può verificare che il richiedente sia il sottoscrittore indicato nella credenziale corrispondente. Il verificatore passa un'asserzione sull'identità del sottoscrittore alla parte fidata (RP).

3.1.2 Mezzi di Autenticazione

Ci sono quattro mezzi generali per autenticare l'identità di un utente, che possono essere usati da soli o con una combinazione di esse:

1. **Qualcosa che l'individuo conosce**

Come una password, un numero di identificazione personale (PIN), o le risposte a una serie di domande prestabilite.

2. **Qualcosa che l'individuo possiede**

Come le keycard elettroniche, smart card e chiavi fisiche. Questo tipo di autenticatore è chiamato token.

3. **Qualcosa che l'individuo è (biometria statica)**

Come il riconoscimento per impronta digitale, retina e faccia.

4. **Qualcosa che l'individuo fa (biometria dinamica)**

Come il riconoscimento tramite il modello di voce, le caratteristiche della scrittura a mano e il ritmo di battitura.

Tutti questi metodi, correttamente implementati e utilizzati, possono fornire l'autenticazione dell'utente. Tuttavia, ogni metodo ha dei problemi. Un avversario può essere in grado di indovinare o rubare una password. Allo stesso modo, un avversario può essere in grado di falsificare o rubare un token. Un utente può dimenticare una password o perdere un token.

L'autenticazione a più fattori si riferisce all'uso di più di uno dei mezzi di autenticazione nella lista precedente. La forza dei sistemi di autenticazione è ampiamente determinata dal numero di fattori incorporati dal sistema. Le implementazioni che usano due fattori sono considerate più forti di quelle che usano un solo fattore. I sistemi che incorporano tre fattori sono più forti di quelli che ne incorporano solo due, e così via.

3.1.3 Valutazione dei rischi per l'autenticazione degli utenti

Ci sono tre concetti separati che vogliamo mettere in relazione l'uno con l'altro: livello di sicurezza, il potenziale d'impatto e le aree di rischio.

- **Livello di sicurezza**

Un livello di sicurezza descrive il grado di certezza di un'organizzazione che un utente ha presentato una credenziale che si riferisce alla sua identità. Più specificamente, la sicurezza è definita come:

- **Il grado di fiducia nel processo di controllo** utilizzato per stabilire l'identità dell'individuo a cui la credenziale è stata rilasciata.
- **Il grado di fiducia dell'individuo** che utilizza la credenziale sia l'individuo a cui la credenziale è stata rilasciata.

SP 800-63-3 riconosce quattro livelli di sicurezza:

1. **Livello:** poca o nessuna fiducia nella validità dell'identità asserita.

Un esempio in cui questo livello è appropriato è un consumatore che si registra per partecipare a una discussione sul sito web di un'azienda. La tipica tecnica di autenticazione a questo livello sarebbe un ID e una password forniti dall'utente al momento della transazione.

2. **Livello:** una certa fiducia nella validità dell'identità asserita.

Le credenziali di livello 2 sono appropriate per un'ampia gamma di affari con il pubblico dove le organizzazioni che richiedono un'affermazione iniziale dell'identità (i cui dettagli sono verificati indipendentemente prima di qualsiasi azione). A questo livello, deve essere usato un qualche tipo di protocollo di autenticazione sicura deve essere usato, insieme a uno dei mezzi di autenticazione riassunti in precedenza e discussi nelle sezioni successive.

3. **Livello:** Alta fiducia nella validità dell'identità asserita Questo livello è appropriato per permettere ai clienti o agli impiegati di accedere a servizi limitati di alto valore ma non al valore più alto.

4. **Livello:** Fiducia molto alta nella validità dell'identità asserita Questo livello è appropriato per permettere ai clienti o agli impiegati di accedere a servizi limitati di alto valore o per i quali un accesso improprio è molto dannoso.

Un concetto strettamente legato a quello di livello di sicurezza è **il potenziale d'impatto**, definisce tre livelli di impatto potenziale sulle organizzazioni o sugli individui in caso di violazione della sicurezza (nel nostro contesto, un errore nell'autenticazione autenticazione dell'utente):

Potenziale D'impatto

- **Low** Un errore di autenticazione potrebbe avere un effetto negativo limitato sulle operazioni organizzative, sulle risorse organizzative o sugli individui. Più specificamente, possiamo dire che l'errore potrebbe:
 - Provocare un danno minore ai beni dell'organizzazione
 - Provocare una perdita finanziaria minore per l'organizzazione o gli individui
 - Risultare in un danno minore agli individui.
- **Moderate** Un errore di autenticazione potrebbe avere un serio effetto negativo effetto. Più specificamente, l'errore potrebbe:
 - Provocare un danno significativo ai beni dell'organizzazione
 - Provocare comporti una perdita finanziaria significativa
 - Comporti un danno significativo alle persone che non che non implica la perdita di vite umane o lesioni gravi in pericolo di vita.
- **High** Un errore di autenticazione potrebbe avere un effetto negativo grave o catastrofico. L'errore potrebbe:
 - Causare un grave danno ai beni dell'organizzazione.
 - Causare una grave perdita finanziaria all'organizzazione o agli individui.
 - Causare un danno grave o catastrofico agli individui che comporta la perdita della vita o lesioni gravi che mettono in pericolo la vita.

La tabella 3.2 mostra una possibile mappatura per vari rischi a cui un'organizzazione può essere esposta. Questa tabella suggerisce una tecnica per fare la valutazione dei rischi. Per un dato sistema informativo o asset di servizio di un'organizzazione, l'organizzazione deve determinare il livello di impatto se si verifica un errore di autenticazione, usando le categorie di impatto, o aree di rischio, che sono preoccupanti.

Per esempio, considerate il potenziale di perdita finanziaria se c'è un errore di autenticazione che risulta in un accesso non autorizzato a un database. A seconda della natura del database, l'impatto potrebbe essere:

Area di Rischio

- **Low** Nel peggior dei casi, una perdita finanziaria insignificante per un'irrilevante responsabilità dell'organizzazione.
- **Moderate** Nel peggior dei casi, una grave perdita finanziaria irrecuperabile per qualsiasi parte, per una grave responsabilità dell'organizzazione.
- **High** Grave o catastrofica perdita finanziaria irrecuperabile per qualsiasi parte per grave una catastrofica responsabilità dell'organizzazione.

3.2 Autenticazione basata su password

Una linea di difesa molto usata contro gli intrusi è il sistema di password. Praticamente tutti i sistemi multiutente, server basati sulla rete, siti di e-commerce basati sul web e altri servizi simili richiedono che un utente fornisca non solo un nome o un identificatore (ID) ma anche una password. Il sistema confronta la password con una password precedentemente memorizzata per quell'ID utente, conservata in un file di password di sistema. La password serve ad autenticare l'ID dell'individuo che accede al sistema. A sua volta, l'ID fornisce sicurezza nei seguenti modi:

- **L'ID determina se l'utente è autorizzato ad accedere ad un sistema.**

In alcuni sistemi, solo coloro che hanno già un ID depositato nel sistema sono permesso di accedere.

- **L'ID determina i privilegi accordati all'utente.**

Alcuni utenti possono avere lo stato di amministratore o "superutente" che permette loro di leggere file ed eseguire funzioni che sono particolarmente protette dal sistema operativo. Alcuni sistemi hanno account ospiti o anonimi, e gli utenti di questi account hanno privilegi più limitati degli altri.

- **L'ID è usato in quello che viene chiamato controllo di accesso discrezionale.**

Per esempio esempio, elencando gli ID degli altri utenti, un utente può concedere loro il permesso di leggere i file di proprietà di quell'utente.

3.2.1 Vulnerabilità delle password

- **Attacco a dizionario offline:**

L'attaccante ottiene il file delle password di sistema e confronta gli hash delle password con gli hash delle password comunemente usate. Se viene trovata una corrispondenza, l'attaccante può ottenere l'accesso con quella combinazione ID/password.

- **Contromisure**

Includono controlli per prevenire l'accesso non autorizzato al file delle password, misure di rilevamento delle intrusioni per identificare una compromissione, e una rapida riemissione delle password se il file delle password viene compromesso.

- **Attacco all'account specifico**

L'attaccante prende di mira un account specifico e presenta password mirate finché non viene scoperta la password corretta.

- **Contromisure**

è un meccanismo di blocco dell'account, che blocca l'accesso all'account dopo un certo numero di tentativi di accesso falliti. La pratica tipica è non più di di cinque tentativi di accesso.

- **Attacco con password popolare**

Una variante dell'attacco precedente consiste nell'utilizzare una password popolare e provarla contro una vasta gamma di ID utente. La tendenza di un utente è quella di scegliere una password che sia facilmente ricordabile; questo purtroppo rende la password facile da indovinare.

- **Contromisure**

Includono politiche per inibire la selezione da parte degli utenti di password comuni e la scansione degli indirizzi IP delle richieste di autenticazione e dei cookie del client per i modelli di invio.

- **Indovinare la password contro un singolo utente**

L'attaccante tenta di ottenere la conoscenza del titolare dell'account e delle politiche di password del sistema e usa tale conoscenza per indovinare la password.

- **Contromisure**

Includono la formazione e l'applicazione di politiche sulle password che rendono le password difficili da indovinare.

- **Dirottamento della stazione di lavoro**

L'attaccante aspetta fino a quando una stazione di lavoro loggata non è sorvegliata.

– **Contromisure**

è la registrazione automatica della workstation dopo un periodo di inattività. Gli schemi di rilevamento delle intrusioni possono essere usati per rilevare i cambiamenti nel comportamento dell’utente.

• **Sfruttare gli errori dell’utente**

Se il sistema assegna una password, allora l’utente è più probabile che la scriva perché è difficile da ricordare. Questa situazione crea il potenziale per un avversario di leggere la password scritta. Un utente può condividere intenzionalmente una password, per permettere ad un collega di condividere i file, per esempio. Inoltre, gli aggressori hanno spesso successo nell’ottenere le password utilizzando tattiche di ingegneria sociale che ingannano l’utente o un account manager a rivelare una password. Molti sistemi informatici sono forniti con password preconfigurate per gli amministratori di sistema. A meno che queste password preconfigurate non vengano cambiate, sono facilmente indovinate.

– **Contromisure**

la formazione degli utenti, il rilevamento delle intrusioni e password più semplici combinate con un altro meccanismo di autenticazione.

• **Sfruttare l’uso di password multiple**

Gli attacchi possono anche diventare molto più efficaci o dannosi se diversi dispositivi di rete condividono la stessa password o una password simile per un dato utente.

– **Contromisure**

includono una politica che proibisce la stessa o password simili su particolari dispositivi di rete.

- **Monitoraggio elettronico**

Se una password viene comunicata attraverso una rete per accedere ad un sistema remoto, è vulnerabile alle intercettazioni. La semplice crittografia non risolve questo problema, perché la password criptata è, in effetti, la password e può essere osservata e riutilizzata da un avversario.

3.2.2 Uso di password con hash

Una tecnica di sicurezza delle password molto diffusa è l'uso di password con hash e di un valore di sale. Questo schema è presente in quasi tutte le varianti di UNIX e in numerosi altri sistemi operativi. Si utilizza la seguente procedura (vedi Figura 3.3). Per caricare una nuova password nel sistema, l'utente sceglie o gli viene assegnata una password. Questa password viene combinata con un valore di "sale" a lunghezza fissa. Nelle vecchie implementazioni, questo valore è legato al momento in cui la password è stata assegnata all'utente.

Le implementazioni più recenti utilizzano un numero pseudorandom o casuale. La password e il sale servono come input a un algoritmo di hashing per produrre un codice hash di lunghezza fissa.

L'algoritmo di hash è progettato per essere lento nell'esecuzione al fine di contrastare gli attacchi. La password hash viene memorizzata, insieme a una copia in chiaro del sale, nel file delle password dell'ID utente corrispondente. file delle password per l'ID utente corrispondente. È stato dimostrato che il metodo della password hash è sicuro contro una serie di attacchi crittoanalitici.

Quando un utente tenta di accedere a un sistema UNIX, fornisce un ID e una password (vedi Figura 3.1). e una password (vedi Figura 3.3b). Il sistema operativo utilizza l'ID per indicizzare il file delle password e recuperare "il sale" in chiaro e la password crittografata.

"Il sale" e la password forniti dall'utente vengono utilizzati come input per la routine di crittografia. Se il risultato corrisponde al valore memorizzato, la password viene accettata.

Il "sale" ha tre funzioni:

1. Impedisce che le password duplicate siano visibili nel file delle password.

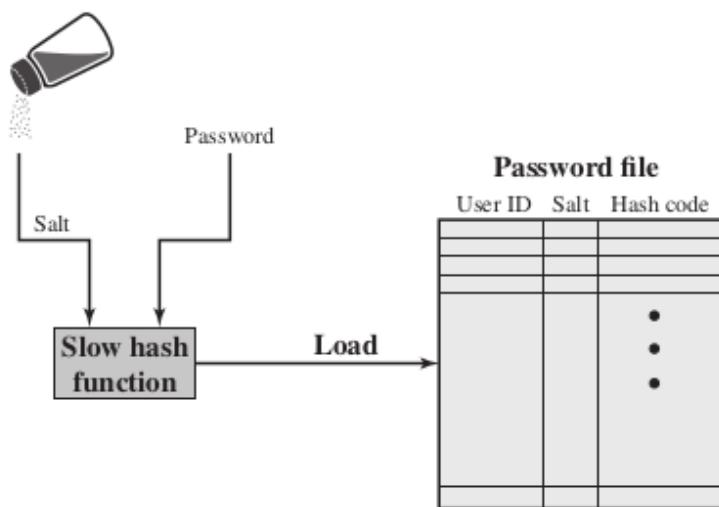
Anche se due utenti scelgono la stessa password, a queste password saranno assegnati valori di valori di sale diversi. Di conseguenza, le password con hash dei due utenti saranno diverse.

2. Questo aumenta notevolmente la difficoltà degli attacchi a dizionario offline.

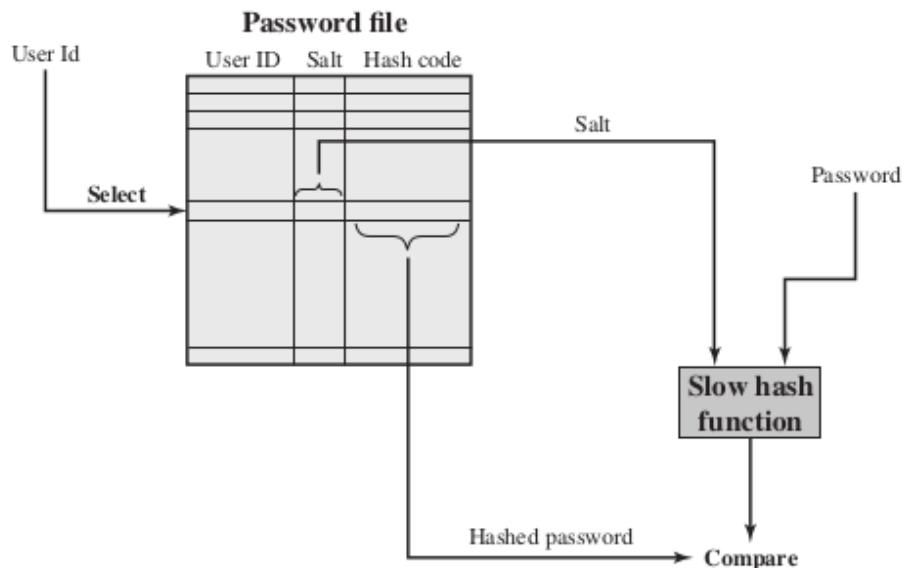
Per un sale di lunghezza bits, il numero di password possibili aumenta di un fattore 2^b , aumentando la difficoltà di indovinare una password la difficoltà di indovinare una password in un attacco a dizionario.

3. Diventa quasi impossibile scoprire se una persona che ha password su due o più sistemi due o più sistemi abbia usato la stessa password su tutti.

Per capire il secondo punto, considerate il modo in cui funzionerebbe un attacco a dizionario offline. L'attaccante ottiene una copia del file delle password. Supponiamo innanzitutto che il sale non venga utilizzato. L'obiettivo dell'attaccante è indovinare una singola password. A tal fine, l'attaccante sottopone alla funzione di hashing un gran numero di password probabili. Se una delle ipotesi corrisponde a uno degli hash del file, l'attaccante ha trovato una password che si trova nel file. Ma con lo schema UNIX, l'aggressore deve prendere ogni ipotesi e sottoporla alla funzione di hash una volta per ogni valore di sale nel file del dizionario, moltiplicando il numero di ipotesi da verificare. Lo schema di password UNIX è soggetto a due minacce. In primo luogo, un utente può ottenere l'accesso a una macchina utilizzando un account ospite o con altri mezzi e poi eseguire un programma per indovinare le password, chiamato password cracker, su quella macchina. L'attaccante dovrebbe essere in grado di controllare molte migliaia di possibili password con un consumo minimo di risorse. Inoltre, se l'avversario è in grado di ottenere una copia del file della password, il programma di cracking può essere eseguito a piacere su un altro computer. In questo modo, l'avversario può esaminare milioni di possibili password in un periodo di tempo ragionevole.



(a) Loading a new password



(b) Verifying a password

Figure 3.3 UNIX Password Scheme

3.2.3 Cracking delle password scelte dall'utente

Approccio Tradizionale L'approccio tradizionale all'indovinare le password, è quello di sviluppare un grande dizionario di possibili password e di provare ognuna di queste con il file delle password. Questo significa che ogni password deve essere sottoposta a un hash usando ogni valore di sale disponibile e poi confrontata con i valori di hash memorizzati. Se non viene trovata alcuna corrispondenza, il programma di cracking prova variazioni su tutte le parole del suo dizionario di password probabili. Tali variazioni includono l'ortografia a ritroso delle parole, numeri aggiuntivi o caratteri speciali, o sequenze di caratteri. Un'alternativa è quella di barattare lo spazio con il tempo precompilando i potenziali valori di hash. In questo approccio l'attaccante genera un grande dizionario di possibili password. Per ogni password, l'attaccante genera i valori di hash associati ad ogni possibile valore di "sale". Il risultato è una mastodontica tabella di valori di hash nota come tabella arcobaleno.

Approccio Moderno Purtroppo, questo tipo di vulnerabilità non è diminuita negli ultimi 25 anni o giù di lì. Gli utenti stanno facendo un lavoro migliore nel selezionare le password e le organizzazioni stanno facendo un lavoro migliore nel costringere gli utenti a scegliere password più forti, un concetto noto come politica delle password complesse.

Tuttavia, le tecniche di cracking delle password sono migliorate per tenere il passo. I miglioramenti sono di due tipi. In primo luogo, la capacità di elaborazione disponibile per il cracking delle password è aumentata drammaticamente. Ora utilizzati sempre più per il calcolo, i processori grafici permettono ai programmi di cracking delle password di lavorare migliaia di volte più velocemente di quanto non facessero solo un dieci anni fa su PC di prezzo simile che usavano solo CPU tradizionali. Un PC che esegue una singola GPU AMD Radeon HD7970, per esempio, può provare in media una $8,2 \times 10^9$ combinazioni di password ogni secondo, a seconda dell'algoritmo utilizzato.

La seconda area di miglioramento nel cracking delle password è l'uso di algoritmi sofisticati per generare potenziali password. I migliori risultati sono stati raggiunti studiando esempi di parole in uso. Per sviluppare tecniche che siano più efficienti ed efficaci dei semplici dizionario e degli attacchi bruteforce, ricercatori e hacker hanno studiato la struttura struttura delle password. Per fare questo, gli analisti hanno bisogno di un grande pool di password di parole reali da studiare, cosa che ora hanno. La prima grande svolta è avvenuta alla fine del 2009, quando un attacco SQL injection contro il servizio di giochi online RockYou.com ha esposto 32 milioni di password in chiaro usate dai suoi membri per accedere ai loro account. Da da allora, numerosi set di file di password trapelate sono diventati disponibili per l'analisi.

3.2.4 Controllo dell'accesso ai file di password

Un modo per contrastare un attacco con password è negare all'avversario l'accesso al file delle password. Se la porzione di password hash del file è accessibile solo da un utente privilegiato, allora l'avversario non può leggerla senza conoscere già la password di un utente privilegiato. Spesso, le password hash sono tenute in un file separato dagli ID utente, indicato come un file di password ombra.

Si presta particolare attenzione a rendere il file shadow password protetto da accessi non autorizzati. Anche se la protezione del file delle password sia certamente utile, rimangono delle vulnerabilità:

- Molti sistemi, compresa la maggior parte dei sistemi UNIX, sono suscettibili di intrusioni.

Un hacker potrebbe essere in grado di sfruttare una vulnerabilità del software nel sistema operativo per bypassare il sistema di controllo degli accessi abbastanza a lungo da estrarre il file di password. In alternativa, l'hacker può trovare una debolezza nel file system o nel sistema di gestione del database che permette l'accesso al file.

- Un incidente di protezione potrebbe rendere il file delle password leggibile, rendendo così compromessi tutti gli account.
- Alcuni utenti hanno account su altre macchine in altri domini di protezione, e usano la stessa password. Quindi, se le password potrebbero essere lette da chiunque su una macchina, una macchina in un'altra posizione potrebbe essere compromessa.
- Una mancanza o una debolezza nella sicurezza fisica può fornire opportunità per un hacker.

A volte, c'è un backup del file delle password su un disco di riparazione disco di riparazione di emergenza o un disco di archiviazione. L'accesso a questo backup permette all'attaccante di leggere il file della password. In alternativa, un utente può avviare da un disco che esegue un altro sistema operativo come Linux e accedere al file da questo sistema operativo.

- Invece di catturare il file delle password di sistema, un altro approccio per raccogliere ID utente e password è attraverso lo sniffing del traffico di rete.

Quindi, una politica di protezione delle password deve integrare le misure di controllo dell'accesso con tecniche per forzare gli utenti a scegliere password difficili da indovinare.

3.2.5 Strategie selezione password

Quando non sono costretti, molti utenti scelgono una password troppo corta o troppo facile da indovinare. All'altro estremo, se agli utenti vengono assegnate password che consistono di otto caratteri stampabili scelti a caso, il cracking della password è effettivamente impossibile. Il nostro obiettivo, quindi, è quello di eliminare le password indovinabili mentre permettendo all'utente di scegliere una password che sia memorizzabile. Quattro tecniche di base sono in uso:

1. Educazione dell'utente
2. Password generate dal computer
3. Controllo reattivo delle password
4. Politica delle password complesse

Gli utenti possono essere informati dell'importanza di usare password difficili da indovinare e possono essere fornire delle linee guida per la selezione di password forti. Questa strategia di educazione degli utenti è improbabile che abbia successo nella maggior parte delle installazioni, in particolare dove c'è una grande una vasta popolazione di utenti o molto turnover. Molti utenti semplicemente ignoreranno le linee guida. Altri possono non essere buoni giudici di ciò che è una password forte. Per esempio, molti utenti (erroneamente) credono che invertire una parola o scrivere in maiuscolo l'ultima lettera renda una password indovinabile.

3.3 Autenticazione Basata sui token

Gli oggetti che un utente possiede ai fini dell'autenticazione sono chiamati token.

3.3.1 Memory Cards

Le **Memory Cards** possono immagazzinare ma non elaborare dati.

La più comune di queste carte è la carta bancaria con una banda magnetica sul retro. Una banda magnetica può memorizzare solo un semplice codice di sicurezza, che può essere letto (e sfortunatamente riprogrammato) da un economico lettore di carte. Ci sono anche schede di memoria che includono una memoria elettronica interna. Le carte di memoria possono essere usate da sole per

- l'accesso fisico, come ad esempio in una stanza d'albergo.
- Per autenticazione, un utente fornisce sia la scheda di memoria che una qualche forma di password o numero di identificazione personale (PIN).

Un'applicazione tipica è uno sportello automatico automatico (ATM). La scheda di memoria, se combinata con un PIN o una password, fornisce una sicurezza significativamente maggiore di una password da sola. Un avversario deve ottenere il possesso fisico possesso fisico della carta (o essere in grado di duplicarla) e in più deve ottenere la conoscenza del PIN.

Tra i potenziali inconvenienti NIST SP 800-12 (An Introduction to Computer Sicurezza: The NIST Handbook, ottobre 1995) nota quanto segue:

– **Richiede un lettore speciale**

Questo aumenta il costo di utilizzo del token e crea l'obbligo di mantenere la sicurezza dell'hardware e del software del lettore.

– **Perdita dei token**

Un token perso impedisce temporaneamente al suo proprietario di ottenere l'accesso al sistema. Quindi, c'è un costo amministrativo nella sostituzione del token perso. Inoltre, se il token viene trovato, rubato o falsificato, allora un avversario deve solo determinare il PIN per ottenere un accesso non autorizzato.

– **Insoddisfazione dell'utente**

Anche se gli utenti possono non avere difficoltà ad accettare l'uso di una scheda di memoria per l'accesso al bancomat, il suo uso per l'accesso al computer può essere considerato scomodo.

3.3.2 Smart Cards

Un'ampia varietà di dispositivi si qualificano come token intelligenti. Questi possono essere classificati lungo quattro dimensioni che non si escludono a vicenda:

- **Caratteristiche fisiche**

I token intelligenti includono un microprocessore incorporato. Un token intelligente che assomiglia a una carta bancaria è chiamato smart card. Altri smart token possono assomigliare a calcolatrici, chiavi o altri piccoli oggetti portatili.

- **Interfaccia Utente**

Le interfacce manuali includono una tastiera e un display per l'interazione uomo / interazione tra uomo e token.

- **Interfaccia Elettronica**

Una smart card o un altro token richiede un'interfaccia elettronica elettronica per comunicare con un lettore/scrittore compatibile.

Una carta può avere uno o entrambi i seguenti tipi di interfaccia:

- **Contatto**

Una smart card a contatto deve essere inserita in un lettore di smart card con una connessione diretta a una piastra di contatto conduttiva sulla superficie della carta (tipicamente placcata in oro).

La trasmissione di comandi, dati e stato della carta avviene attraverso questi punti di contatto fisico.

- **Senza Contatto**

Una carta senza contatto richiede solo la vicinanza di un lettore.

Sia il lettore che la carta hanno un'antenna, e i due comunicano utilizzando frequenze radio. La maggior parte delle carte senza contatto derivano anche l'energia per il chip interno da questo segnale elettromagnetico. La portata è tipicamente da un mezzo a tre pollici per le carte non alimentate a batteria, ideale per applicazioni come l'ingresso in un edificio e il pagamento che richiedono un'interfaccia della carta molto veloce.

- **Protocollo di autenticazione**

Lo scopo di un token intelligente è quello di fornire un mezzo per l'autenticazione dell'utente.

Possiamo classificare i protocolli di autenticazione usati con token intelligenti in tre categorie:

1. Statico

Con un protocollo statico, l'utente si autentica con il token, poi il token autentica l'utente al computer. L'ultima metà di questo protocollo è simile al funzionamento di un token di memoria.

2. Generatore dinamico di password

In questo caso, il token genera una password unica periodicamente (ad esempio, ogni minuto). Questa password viene poi inserita nel sistema informatico per l'autenticazione, sia manualmente dall'utente o elettronicamente tramite il token. Il token e il sistema informatico devono essere inizializzati e mantenuti sincronizzati in modo che il computer conosca la password che è corrente per questo token.

3. Sfida-risposta

In questo caso, il sistema informatico genera una sfida, come una stringa casuale di numeri. Il token intelligente genera una risposta basata sulla sfida. Per esempio, si potrebbe usare la crittografia a chiave pubblica e il token potrebbe criptare la stringa di sfida con la chiave privata del token.

3.3.3 Elettronic Identity Cards

Un'applicazione di crescente importanza è l'uso di una smart card come carta d'identità nazionale per i cittadini. Una carta d'identità elettronica nazionale (eID) può servire agli stessi scopi di altre carte d'identità nazionali, e carte simili come la patente di guida, per l'accesso ai servizi governativi e commerciali. Inoltre, una carta eID può fornire una prova di identità più forte ed essere usata in una più ampia varietà di applicazioni. In effetti, una carta eID è una smart card che è stata verificata dal governo nazionale come valida e autentica.

L'identity cards conterrà:

- Dati personali
Come nome, data di nascita e indirizzo
- Numero del documento
Un identificatore alfanumerico di nove caratteri unico per ogni carta.
- Numero di accesso alla carta (CAN)
Un numero decimale casuale di sei cifre stampato sulla faccia della carta.
- Zona a lettura ottica (MRZ)
Tre righe di testo leggibile dall'uomo e dalla macchina sul retro della carta. Anche questo può essere usato come password.

Le funzioni dell'identity cards sono:

- **ePass**
Questa funzione è riservata all'uso governativo e memorizza una rappresentazione digitale dell'identità del titolare della carta. Questa funzione è simile a, e può essere usata per, un passaporto elettronico. Anche altri servizi governativi possono usare ePass. La funzione ePass deve essere implementata sulla carta.
- **eID**
Questa funzione è per uso generale in una varietà di applicazioni governative e commerciali. applicazioni commerciali. La funzione eID memorizza un record di identità a cui i servizi autorizzati possono accedere con il permesso del titolare della carta. I cittadini scelgono se vogliono attivare questa funzione.
- **eSign**
Questa funzione opzionale memorizza una chiave privata e un certificato che verifica la chiave, è usata per generare una firma digitale. Un centro di fiducia del settore privato emette il certificato.

3.4 Autenticazione Biometrica

Un sistema di autenticazione biometrica tenta di autenticare un individuo sulla base di le sue caratteristiche fisiche uniche. Queste includono:

- **Caratteristiche statiche**

come le impronte digitali, la geometria della mano, le caratteristiche del viso e i modelli della retina e dell'iride

- **Caratteristiche dinamiche**

come l'impronta vocale e la firma.

In sostanza, la biometria si basa sul riconoscimento dei modelli. Rispetto alle password e ai token, l'autenticazione è tecnicamente più complessa e costosa. Sebbene sia usata in un numero di applicazioni specifiche, la biometria deve ancora maturare come strumento standard per l'autenticazione degli utenti ai sistemi informatici.

3.4.1 Caratteristiche fisiche utilizzate nelle applicazioni biometriche

Un certo numero di diversi tipi di caratteristiche fisiche sono in uso o in fase di studio per l'autenticazione dell'utente. Le più comuni sono le seguenti:

- **Caratteristiche facciali**

Le caratteristiche facciali sono il mezzo più comune per l'identificazione da uomo a uomo, quindi è naturale considerarle per l'identificazione tramite computer. L'approccio più comune è quello di definire le caratteristiche basate sulla posizione relativa e la forma delle caratteristiche facciali chiave, come occhi, sopracciglia, naso, labbra e forma del mento.

- **Impronte digitali**

Le impronte digitali sono state usate come mezzo di identificazione per secoli, e il processo è stato sistematizzato e automatizzato in particolare per l'applicazione della legge. Un'impronta digitale è il modello di creste e solchi sulla superficie del polpastrello. Si ritiene che le impronte digitali siano uniche in l'intera popolazione umana. In pratica, il riconoscimento automatico delle impronte digitali è un sistema di corrispondenza che estrae un certo numero di caratteristiche dall'impronta digitale per memorizzarle come surrogato numerico del modello completo dell'impronta digitale.

- **Geometria della mano**

I sistemi di geometria della mano identificano le caratteristiche della mano, compresa la forma, la lunghezza e la larghezza delle dita.

- **Modello della retina**

Il modello formato dalle vene sotto la superficie della retina è unico e quindi adatto all'identificazione. Un sistema biometrico retinico ottiene un'immagine digitale del modello retinico proiettando un fascio di luce visiva o infrarossa a bassa intensità nell'occhio.

- **Iride**

Un'altra caratteristica fisica unica è la struttura dettagliata dell'iride.

- **Firma**

Ogni individuo ha uno stile unico di scrittura e questo si riflette specialmente nella firma, che è tipicamente una sequenza scritta frequentemente. Tuttavia, più campioni di firma di un singolo individuo non saranno identici.

- **La voce**

Mentre lo stile della firma di un individuo riflette non solo gli unici attributi fisici dello scrittore ma anche l'abitudine alla scrittura che si è sviluppata, i modelli di voce sono più strettamente legati alle caratteristiche fisiche e anatomiche del parlante. Ciononostante, c'è ancora una variazione da campione a campione nel tempo dallo stesso parlante, complicando il compito di riconoscimento biometrico.

3.4.2 Funzionamento di un sistema di autenticazione biometrica

Ogni individuo che deve essere incluso nel database degli utenti autorizzati deve prima essere iscritto al sistema. Questo è analogo all'assegnazione di una password a un utente. Per un sistema biometrico, l'utente presenta al sistema un nome e, tipicamente, un qualche tipo di password o PIN. Allo stesso tempo, il sistema rileva alcune caratteristiche biometriche di questo utente (ad esempio, l'impronta digitale del dito indice destro). Il sistema digitalizza l'input e poi estrae un insieme di caratteristiche che possono essere memorizzate come un numero o un insieme di numeri che rappresentano questa caratteristica biometrica unica. Questo insieme di numeri viene chiamato template dell'utente. L'utente è ora iscritto al sistema, che mantiene per l'utente un nome (ID), forse un PIN o una password, e il valore biometrico.

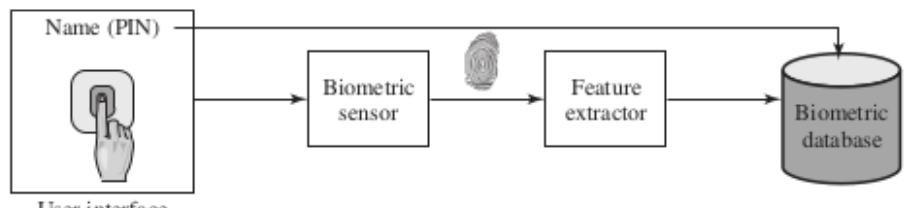
A seconda dell'applicazione, l'autenticazione dell'utente su un sistema biometrico comporta la verifica o l'identificazione.

- **Verifica**

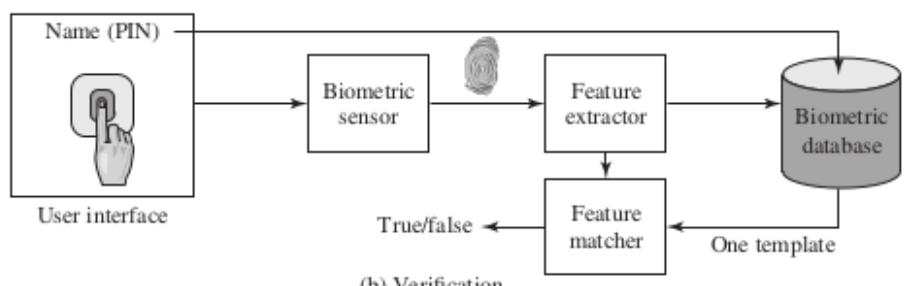
è analoga a quella di un utente che accede a un sistema usando una scheda di memoria o una smart card accoppiata a una password o a un PIN. Per la verifica biometrica, l'utente inserisce un PIN e utilizza anche un sensore biometrico. Il sistema estrae la caratteristica corrispondente e la confronta con il modello memorizzato per questo utente. Se c'è una corrispondenza, allora il sistema autentica questo utente.

- **Identificazione**

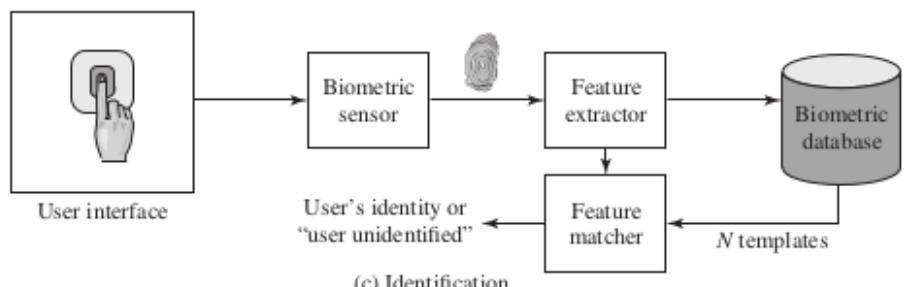
l'individuo usa il sensore biometrico ma non presenta informazioni aggiuntive. Il sistema quindi confronta il modello presentato con l'insieme dei modelli memorizzati. Se c'è una corrispondenza, l'utente viene identificato. Altrimenti, l'utente viene rifiutato.



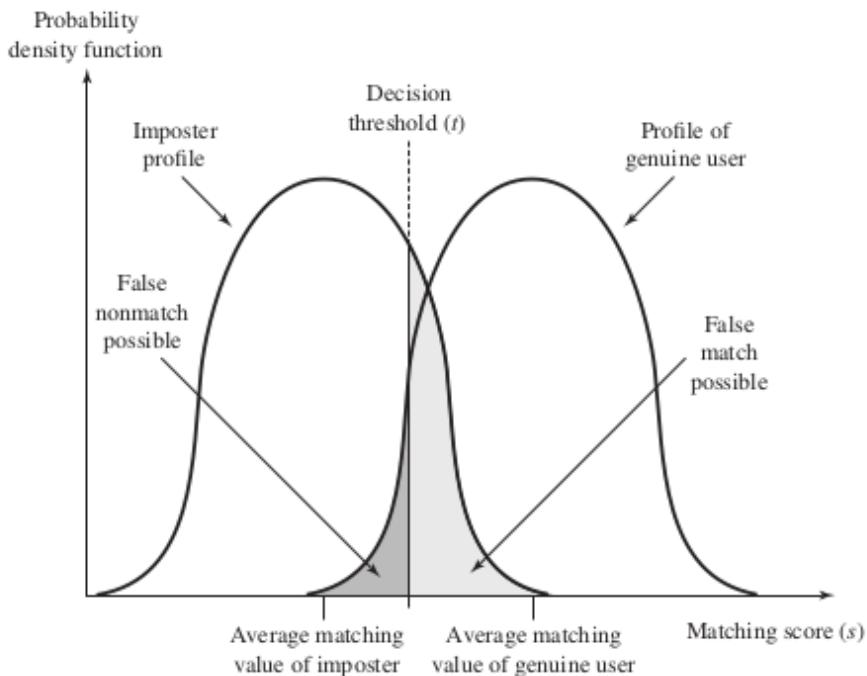
(a) Enrollment



(b) Verification



(c) Identification



3.4.3 Precisione Biometrica

In qualsiasi schema biometrico, alcune caratteristiche fisiche dell'individuo sono mappate in una rappresentazione digitale. Per ogni individuo, una singola rappresentazione digitale, o template, è memorizzata nel computer. Quando l'utente deve essere autenticato, il sistema confronta il modello memorizzato con il modello presentato. Data la complessità delle caratteristiche fisiche, non ci si può aspettare che ci sia una corrispondenza esatta tra i due modelli. Piuttosto, il sistema usa un algoritmo per generare un punteggio (tipicamente un singolo numero) che quantifica la somiglianza tra l'input e il modello memorizzato. Per procedere con la discussione, definiamo i seguenti termini.

- Il tasso di falsa corrispondenza

è la frequenza con cui i campioni biometrici provenienti da diverse fonti diverse sono erroneamente valutati come provenienti dalla stessa fonte. Il tasso di falsa non corrispondenza è la frequenza con cui i campioni della stessa fonte sono erroneamente valutati essere di fonti diverse.

- La funzione di densità
tipicamente forma una curva a campana.

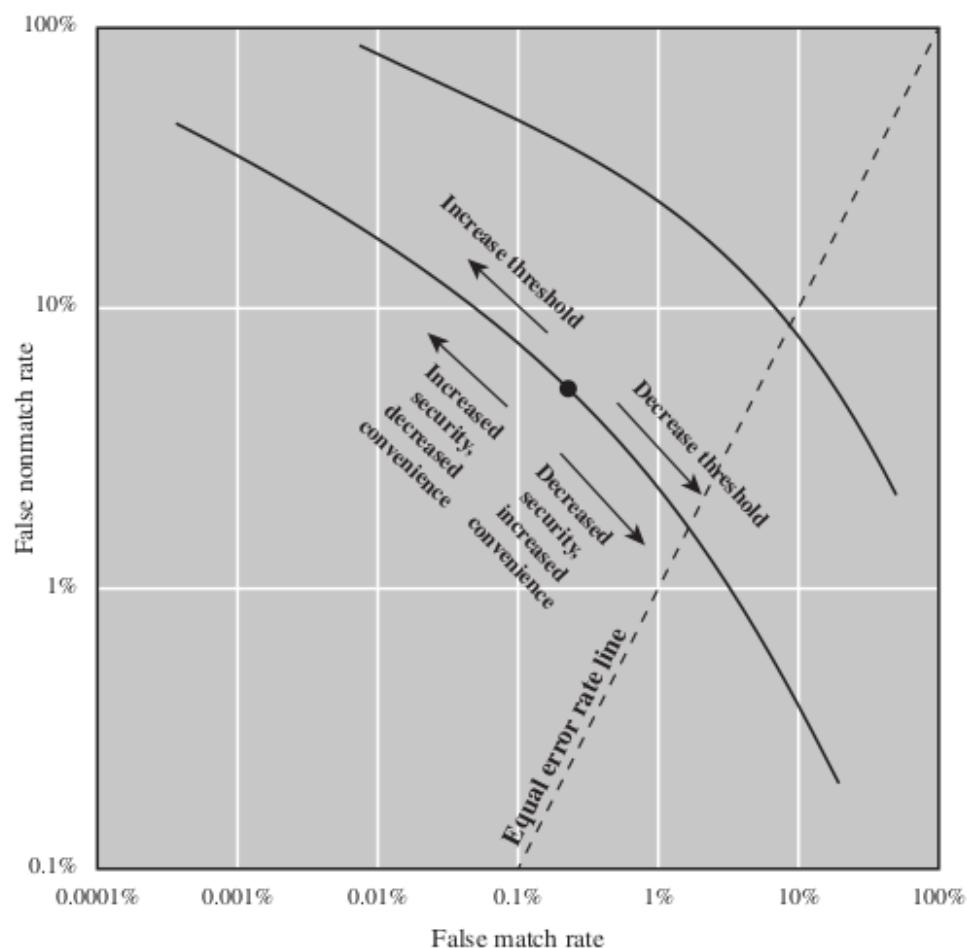


Figure 3.11 Idealized Biometric Measurement Operating Characteristic Curves (log-log scale)

3.4.4 Protocollo delle password

In questo esempio, un utente trasmette prima la sua identità a all'host remoto. L'host genera un numero casuale r , spesso chiamato nonce, restituisce questo nonce all'utente.

Inoltre, l'host specifica due funzioni, $h()$ e $f()$, da utilizzare nella risposta. Questa trasmissione dall'host all'utente è la sfida. La risposta dell'utente è la quantità $f(r', h(P'))$, dove $r' = r$ e P' è la password dell'utente.

La funzione h è una funzione hash, quindi la risposta consiste nella funzione hash della password dell'utente combinata con il numero casuale utilizzando la funzione.

L'host memorizza la funzione hash della password di ogni utente registrato, rappresentata come $h(P(U))$ per l'utente U . Quando arriva la risposta, l'host confronta $f(r', h(P'))$ con la $f(r, h(P(U)))$ calcolata.) Se le quantità corrispondono, l'utente è autenticato. Questo schema difende da diverse forme di attacco. L'host memorizza non la password ma un codice hash della password. Questo protegge la password dagli intrusi nel sistema host. Inoltre, nemmeno l'hash della password viene trasmesso direttamente, ma piuttosto una funzione in cui l'hash della password è uno degli argomenti. Così, per una funzione f adatta, l'hash della password non può essere catturato durante la trasmissione.

Infine, l'uso di un numero casuale tenta di difendere da un attacco di replay, in cui un avversario cattura la trasmissione dell'utente e tenta di accedere a un sistema ritrasmettendo i messaggi.

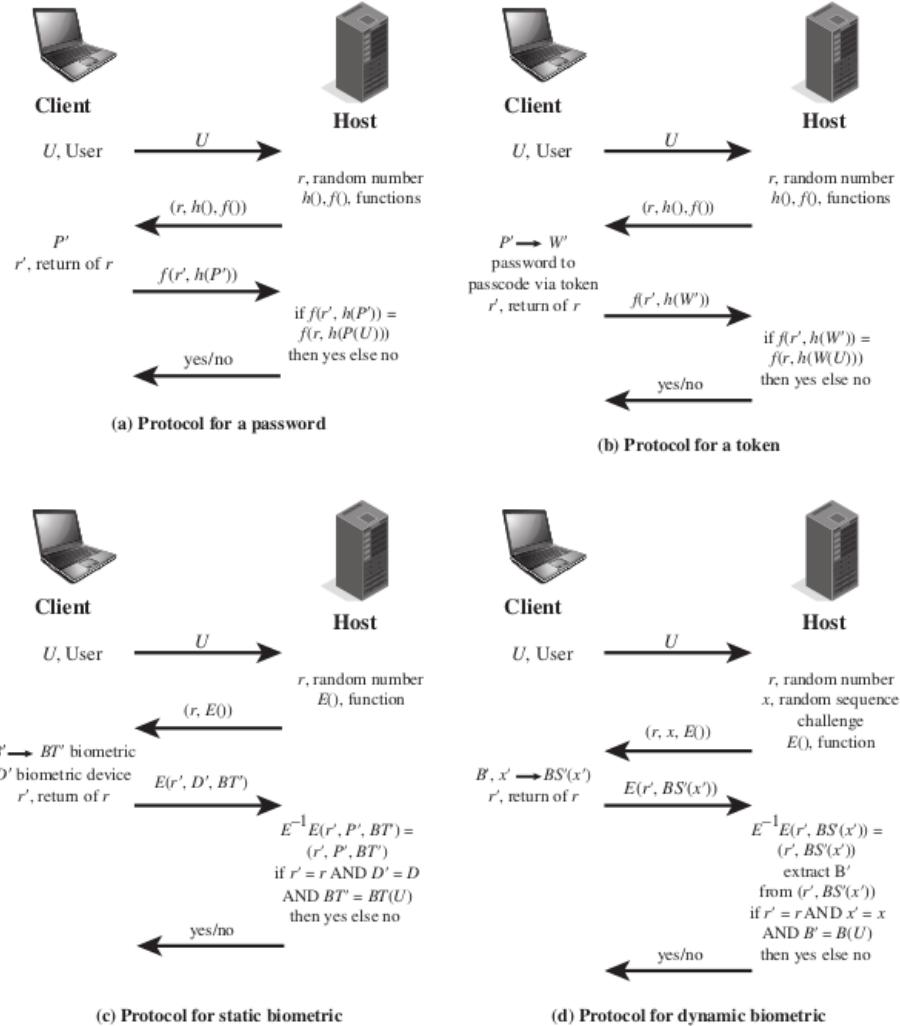


Figure 3.13 Basic Challenge-Response Protocols for Remote User Authentication
Source: Based on [OGOR03].

3.4.5 Protocollo di Token

Come prima, un utente trasmette prima la sua identità all'host remoto. L'host restituisce un numero casuale e gli identificatori delle funzioni $f()$ e $h()$ da utilizzare nella risposta.

Alla fine dell'utente, il token fornisce un codice di accesso W' . Il token memorizza un codice statico statico o genera un codice casuale una tantum. Per un codice casuale una tantum, il token deve essere sincronizzato. Per un codice casuale una tantum, il token deve essere sincronizzato in qualche modo con l'host. In entrambi i casi, l'utente attiva il codice inserendo una password P' . Questa password è condivisa solo tra l'utente e il token e non coinvolge l'host remoto. Il token risponde all'host con la quantità $f(r', h(W'))$. Per un codice di accesso statico, l'host memorizza il valore hashed $h(W(U))$; per un passcode dinamico, l'host genera un passcode una tantum (sincronizzato con quello generato dal

token) e prende il suo hash. L'autenticazione procede poi nello stesso modo del protocollo della password.

3.4.6 Protocollo biometrico statico

Come prima cosa, l'utente trasmette un ID all'host, che risponde con un numero casuale r , in questo caso, l'identificatore di una crittografia $E()$. Sul lato utente c'è un sistema client che controlla un dispositivo biometrico. Il sistema genera un template biometrico BT' dal biometrico dell'utente B' e restituisce il testo cifrato $E(r', D', BT')$, dove D' identifica questo particolare dispositivo biometrico. L'host decifra il messaggio in arrivo per recuperare i tre parametri trasmessi e li confronta con i valori memorizzati localmente.

Per una corrispondenza, l'host deve trovare $r' = r$. Inoltre, il punteggio di corrispondenza tra BT' e il modello memorizzato deve superare una soglia predefinita. Infine, l'host fornisce una semplice autenticazione del dispositivo di cattura biometrica confrontando l'ID del dispositivo in entrata con un elenco di dispositivi registrati nel database dell'host.

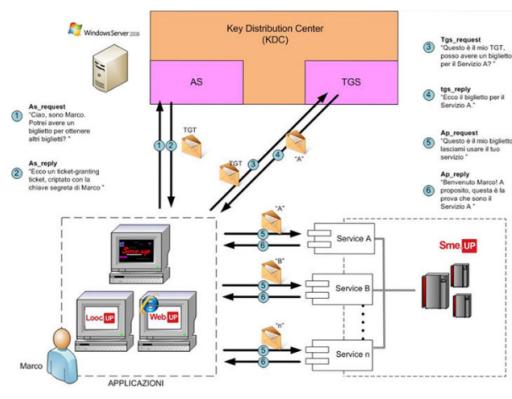
3.4.7 Protocollo Biometrico Dinamico

La principale differenza rispetto al caso di una biometria stabile è che l'host fornisce una sequenza casuale fornisce una sequenza casuale e un numero casuale come sfida. La sfida è una sequenza di numeri, caratteri o parole. L'utente umano all'estremità del client deve quindi vocalizzare (verifica con altoparlante), digitare (verifica dinamica della tastiera) o scrivere (verifica a mano) o scrivere (verifica della scrittura) la sequenza per generare un segnale biometrico $BS'(x')$. Il lato client critta il segnale biometrico e il numero casuale. All'indirizzo lato host, il messaggio in arrivo viene decifrato. Il numero casuale in arrivo r' deve essere una corrispondenza esatta con il numero casuale che è stato originariamente utilizzato come sfida (r). Inoltre, l'host genera un confronto basato sul segnale biometrico in entrata $BS'(x')$, il template memorizzato $BT(U)$ per questo utente e il segnale originale x . Se il valore di confronto supera una soglia predefinita, l'utente viene autenticato.

Capitolo 4

Keberos

Imitologia Greca: cane a tre teste guardiano delle porte dell'ade. Sviluppato a fine anni '80, il goal: segretezza, autentica (ad accesso singolo), temporalità o Le chiavi usate hanno validità limitata onde prevenire replay attack. Usa i timespamp, che richiedono macchine sincronizzate, contro replay attack.



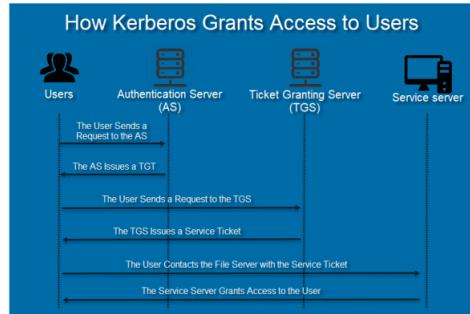


Figura 67: Kerberos Schema 2

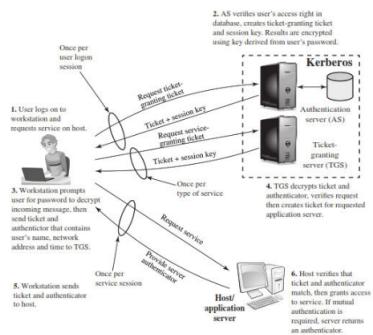


Figura 68: Kerberos schema lezione

4.0.1 Caratteristiche

- 3 fasi: autenticazione, autorizzazione, servizio
- Ultime 2 opzionali e trasparenti all'utente
- Ognuna fornisce credenziali per successiva
 - Fase 1 fornisce authKey e authTicket per la 2
 - Fase 2 fornisce servKey e servTicket per la fase 3
- Ogni tipo di chiave si sessione ha sua durata (in genere quelle per i servizi dura meno, es. sessione 1 giorno, servizio 1 ora)
- Una authKey può criptare diverse servKey

Nota: Chiave pubblica e privata per autenticazione, chiave di sessione per la sessione (authKey e servKey) perché ha una durata limitata e servono a dare confidentiality/confidencialità a questa comunicazione in modo da cifrarla.

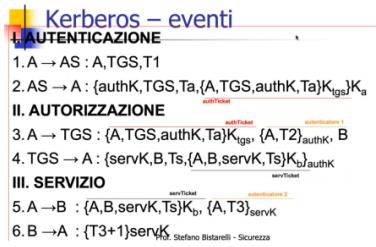


Figura 69: Kerberos eventi Slide

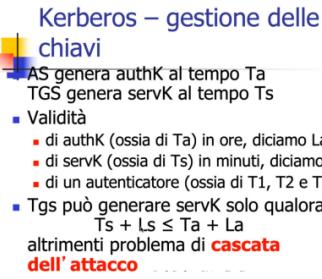


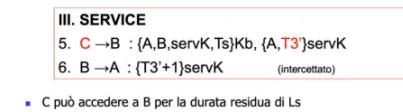
Figura 70: Kerberos - gestione delle chiavi slide

- ## Discussion
- Replay attack su N-S simmetrico nell'ipotesi che chiavi di sessione **vecchie** siano insicure
 - Vecchio: genericamente, del passato – non esiste temporalità
 - consequential attack su Kerberos nell'ipotesi che chiavi di sessione **scadute** siano insicure
 - Scaduto: specificatamente, il cui intervallo di validità sia scaduto – esiste temporalità

Attacchi consequenziali

Def. Un attacco ne provoca altri direttamente

- Supponiamo che C abbia violato una chiave di sessione (di autorizzazione) scaduta authK che B condivide con A
- Con semplice decodifica ottiene servK ancora valida se non si impone $Ts + Ls \leq Ta + La$



- C può accedere a B per la durata residua di Ls

- **Versione IV:** ristretta a un singolo realm
- **Versione V:** funzionamento inter-realm
- Altre differenze: scelta dell'algoritmo di crittografia impossibile in IV (DES); scelta del lifetime impossibile in IV
- Versione V è uno standard di vastissimo utilizzo (specificato in RFC1510)

Figura 71: Differenza ultime due versioni Kerberos

Usare Kerberos

- Serve un KDC sul proprio dominio
- Tutte le applicazioni partecipanti devono essere servizi Kerberos
- Problema: gli algoritmi crittografici americani non possono essere esportati
 - I sorgenti di Kerberos non possono lasciare gli USA
 - Il crittosistema va implementato localmente

4.0.2 Descrizione del protocollo

Il client si autentica **sull'Authentication Server (AS)** che inoltra il nome utente a un centro di distribuzione delle chiavi (KDC) . Il KDC emette un **ticket-granting ticket (TGT)** , che è timestamp e lo crittografa utilizzando la chiave segreta del **ticket-granting service (TGS)** e restituisce il risultato crittografato alla workstation dell'utente. Questa operazione viene eseguita raramente, in genere all'accesso dell'utente; il TGT scade a un certo punto sebbene possa essere rinnovato in modo trasparente dal gestore della sessione dell'utente mentre è connesso. Quando il client ha bisogno di comunicare con un servizio su un altro nodo (un "principal", nel gergo Kerberos), il client invia il **TGT** al **TGS**, che di solito condivide lo stesso host del KDC. Il servizio deve essere già stato registrato con il TGS con un nome dell'entità servizio (SPN) . Il client utilizza l'SPN per richiedere l'accesso a questo servizio. Dopo aver verificato che il TGT è valido e che l'utente è autorizzato ad accedere al servizio richiesto, il TGS emette il ticket e le chiavi di sessione al client. Il client invia quindi il ticket al **server del servizio (SS)** insieme alla sua richiesta di servizio.

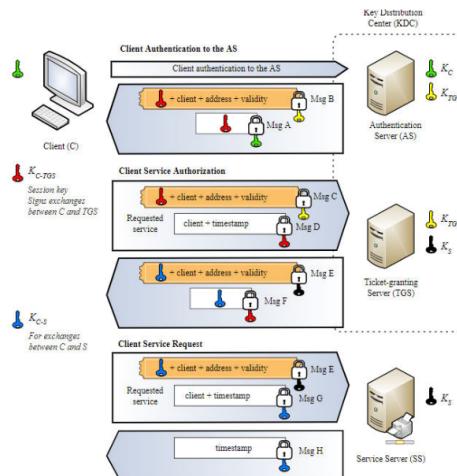


Figura 72: it.qaz.wiki Kerberos Schema

4.0.3 Accesso basato su client utente

Un utente immette un nome utente e una password sul computer client . Altri meccanismi di credenziali come pkinit (RFC 4556) consentono l'uso di chiavi pubbliche al posto di una password. Il client trasforma la password nella chiave di un cifrario simmetrico. Questo utilizza la pianificazione della chiave incorporata o un hash unidirezionale , a seconda della suite di crittografia utilizzata.

4.0.4 Autenticazione client

1. Il client invia un messaggio di testo in chiaro dell'ID utente all'AS (Authentication Server) richiedendo servizi per conto dell'utente. (Nota: né la chiave segreta né la password vengono inviate all'AS.)
2. L'AS verifica se il client è nel suo database. In tal caso, l'AS genera la chiave segreta eseguendo l'hashing della password dell'utente trovata nel database (ad esempio, Active Directory in Windows Server) e restituisce i seguenti due messaggi al client:
 - Messaggio A: chiave di sessione client / TGS crittografata utilizzando la chiave segreta del client/ utente.
 - Messaggio B: Ticket-Granting-Ticket (TGT, che include l'ID client, l'indirizzo di rete del client , il periodo di validità del ticket e la chiave di sessione client / TGS) crittografato utilizzando la chiave segreta del TGS.
3. Una volta che il client riceve i messaggi A e B, tenta di decrittografare il messaggio A con la chiave segreta generata dalla password inserita dall'utente. Se la password inserita dall'utente non corrisponde alla password nel database AS, la chiave segreta del client sarà diversa e quindi non sarà in grado di decifrare il messaggio A. Con una password e una chiave segreta valide il client decrittografa il messaggio A per ottenere la chiave di sessione client / TGS . Questa chiave di sessione viene utilizzata per ulteriori comunicazioni con il TGS. (Nota: il client non può decrittografare il messaggio B, poiché è crittografato utilizzando la chiave segreta del TGS.) A questo punto, il client dispone di informazioni sufficienti per autenticarsi al TGS.

4.0.5 Autenticazione del servizio clienti

1. Quando richiede servizi, il client invia i seguenti messaggi al TGS:
 - Messaggio C: composto dal messaggio B (il TGT crittografato che utilizza la chiave segreta TGS) e dall'ID del servizio richiesto.
 - Messaggio D: Autenticatore (composto dall'ID client e dal timestamp), crittografato utilizzando la chiave di sessione client / TGS

2. Alla ricezione dei messaggi C e D, il TGS recupera il messaggio B dal messaggio C. Decrittografa il messaggio B utilizzando la chiave segreta TGS. Questo fornisce la "chiave di sessione client / TGS" e l'ID client (entrambi sono nel TGT). Utilizzando questa "chiave di sessione client / TGS", il TGS decrittografa il messaggio D (Authenticator) e confronta gli ID client dai messaggi B e D; se corrispondono, il server invia i seguenti due messaggi al client:

- Messaggio E: ticket da client a server (che include ID client, indirizzo di rete client, periodo di validità e chiave di sessione client / server) crittografato utilizzando la chiave segreta del servizio.
- Messaggio F: chiave di sessione client / server crittografata con chiave di sessione client / TGS.

4.0.6 Richiesta di servizio clienti

1. Dopo aver ricevuto i messaggi E ed F da TGS, il client dispone di informazioni sufficienti per autenticarsi al Service Server (SS). Il client si connette alla SS e invia i seguenti due messaggi:
 - Messaggio E: dal passaggio precedente (il ticket da client a server , crittografato utilizzando la chiave segreta del servizio).
 - Messaggio G: un nuovo autenticatore, che include l'ID client, il timestamp ed è crittografato utilizzando la chiave di sessione client / server.
2. L'SS decrittografa il ticket (messaggio E) utilizzando la propria chiave segreta per recuperare la chiave di sessione client / server . Utilizzando la chiave delle sessioni, SS decrittografa l'Autenticatore e confronta l'ID client dai messaggi E e G, se corrispondono, il server invia il seguente messaggio al client per confermare la sua vera identità e disponibilità a servire il client:
 - Messaggio H: il timestamp trovato nell'autenticatore del client (più 1 nella versione 4, ma non necessario nella versione 5), crittografato utilizzando la chiave di sessione client / server.
3. Il client decrittografa la conferma (messaggio H) utilizzando la chiave di sessione client / server e controlla se il timestamp è corretto. In tal caso, il client può considerare attendibile il server e può iniziare a inviare richieste di servizio al server.
4. Il server fornisce i servizi richiesti al client.

4.0.7 Inconvenienti e limitazioni

- Kerberos ha requisiti di tempo rigorosi, il che significa che gli orologi degli host coinvolti devono esseresincronizzati entro i limiti configurati. I ticket hanno un periodo

di disponibilità temporale e se l'orologio dell'host non è sincronizzato con l'orologio del server Kerberos, l'autenticazione fallirà. La configurazione predefinita per MIT richiede che gli orari dell'orologio non siano distanti più di cinque minuti. In pratica, i demoni Network Time Protocol vengono solitamente utilizzati per mantenere sincronizzati gli orologi host. Si noti che alcuni server (l'implementazione di Microsoft è uno di questi) possono restituire un risultato KRB_AP_ERR_SKEW contenente l'ora del server crittografata nel caso in cui entrambi gli orologi abbiano un offset maggiore del valore massimo configurato. In tal caso, il client potrebbe riprovare calcolando l'ora utilizzando l'ora del server fornita per trovare l'offset. Questo comportamento è documentato in RFC 4430.

- Il protocollo di amministrazione non è standardizzato e differisce tra le implementazioni del server. Le modifiche alla password sono descritte in RFC 3244
- In caso di adozione della crittografia simmetrica (Kerberos può funzionare utilizzando la crittografia simmetrica o asimmetrica (chiave pubblica)), poiché tutte le autenticazioni sono controllate da un centro di distribuzione delle chiavi centralizzato (KDC), la compromissione di questa infrastruttura di autenticazione consentirà a un utente malintenzionato di impersonare qualsiasi utente
- Ogni servizio di rete che richiede un nome host diverso avrà bisogno del proprio set di chiavi Kerberos. Ciò complica l'hosting virtuale e i cluster.
- Kerberos richiede che gli account utente e i servizi abbiano una relazione affidabile con il server di token Kerberos.
- La fiducia del client richiesta rende difficile la creazione di ambienti a fasi (ad esempio, domini separati per l'ambiente di test, l'ambiente di pre-produzione e l'ambiente di produzione): è necessario creare relazioni di fiducia del dominio che impediscono una netta separazione dei domini dell'ambiente o che siano necessari client utente aggiuntivi previsto per ogni ambiente.

4.0.8 Vulnerabilità

La crittografia Data Encryption Standard (DES) può essere utilizzata in combinazione con Kerberos, ma non è più uno standard Internet perché è debole. Esistono vulnerabilità di sicurezza in molti prodotti legacy che implementano Kerberos perché non sono stati aggiornati per utilizzare crittografie più recenti come AES invece di DES. Nel novembre 2014, Microsoft ha rilasciato una patch (MS14-068) per correggere una vulnerabilità sfruttabile nell'implementazione Windows del Kerberos Key Distribution Center (KDC). La vulnerabilità presumibilmente consente agli utenti di "elevare" (e abusare) i propri privilegi, fino al livello di dominio.

4.0.9 Altra descrizione del protocollo (wikipedia)

In informatica e telecomunicazioni Kerberos è un protocollo di rete per l'autenticazione forte che permette a diversi terminali di comunicare su una rete informatica insicura provando la propria identità mediante l'utilizzo di tecniche di crittografia simmetrica. Kerberos previene attacchi quali l'intercettazione e i replay attack ed assicura l'integrità dei dati. I suoi progettisti mirarono soprattutto ad un modello client-server, e fornisce una mutua autenticazione cioè sia l'utente sia il fornitore del servizio possono verificare l'identità dell'altro.

Descrizione

Kerberos si basa sul protocollo di Needham-Schroeder. Utilizza una terza parte affidabile per centralizzare la distribuzione delle chiavi detta Key Distribution Center (KDC), che consiste di due parti separate logicamente: l'Authentication Server (AS) e il Ticket Granting Server (TGS). Kerberos funziona utilizzando dei "biglietti" (detti ticket) che servono per provare l'identità degli utenti.

L'AS mantiene un database delle chiavi segrete; ogni entità sulla rete — che sia un client o un server — condivide la chiave segreta solo con l'AS. La conoscenza di questa chiave serve per provare l'identità di un'entità. Per comunicazioni tra due entità, Kerberos genera una chiave di sessione, che può essere utilizzata dai due terminali per comunicare.

Il protocollo

Il protocollo può essere definito come segue utilizzando la notazione per protocolli di sicurezza, dove Alice (A) si autentica presso Bob (B) usando il server S:

$$\begin{aligned} A &\rightarrow S : A, B \\ S &\rightarrow A : \{T_S, L, K_{AB}, B, \{T_S, L, K_{AB}, A\}_{K_{BS}}\}_{K_{AS}} \\ A &\rightarrow B : \{T_S, L, K_{AB}, A\}_{K_{BS}}, \{A, T_A\}_{K_{AB}} \\ B &\rightarrow A : \{T_A + 1\}_{K_{AB}} \end{aligned}$$

La sicurezza del protocollo si basa fortemente sui timestamp T e sui tempi di vita L come indicatori affidabili della creazione recente della comunicazione per evitare replay attack (vedi logica BAN).

È importante notare come il server S stia sia come Authentication Service (AS) sia come Ticket Granting Service (TGS).

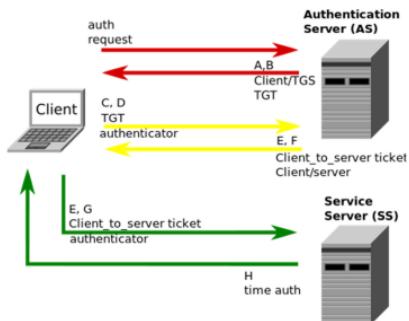


Figura 73: Kerberos schema wikipedia

Operazioni di Kerberos

Quella che segue è una descrizione semplificata del protocollo. Saranno utilizzate le seguenti abbreviazioni: AS = Authentication Server, TGS = Ticket Granting Server, SS = Service Server. In breve: il client si autentica presso AS che gli fornisce un ticket di sessione per accedere a TGS, si autentica presso TGS e riceve il ticket per aprire una sessione di comunicazione con SS. In dettaglio

Utente: autenticazione di base

1. Un utente inserisce username e password sul client

Client: Autenticazione AS

1. Il client manda un messaggio non criptato all'AS richiedendo i servizi per l'utente. ("L'utente XYZ vorrebbe richiedere dei servizi"). Né la chiave segreta né la password vengono inviate all'AS.
2. L'AS controlla se il client è nel suo database. Se lo è invia due messaggi al client:
 - Messaggio A: Chiave di sessione client-TGS criptata usando la chiave segreta dell'utente.
 - Messaggio B: Ticket-Granting Ticket (che include l'identificativo del client, l'indirizzo di rete, il tempo di validità del ticket e la chiave di sessione client-TGS). Il Ticket-Granting Ticket è criptato utilizzando la chiave segreta di TGS.
3. Quando il client riceve i messaggi A e B, decripta il messaggio A ottenendo la chiave di sessione client-TGS. Questa chiave è utilizzata per le successive comunicazioni con TGS. (Nota: il client non può decriptare il Messaggio B, che è stato criptato con la chiave segreta di TGS). A questo punto il client possiede i mezzi per autenticarsi presso TGS.

Client: Autenticazione TGS

1. Quando richiede dei servizi, il client invia i seguenti due messaggi a TGS:
 - Messaggio C: composto dal Ticket-Granting Ticket (mandatogli dal AS nel messaggio B) e dall'identificativo del servizio richiesto
 - Messaggio D: autenticatore (Authenticator) (che è formato da identificativo del client e imestamp), criptato usando la chiave di sessione client—TGS.
2. Ricevendo i messaggi C e D, TGS decripta il messaggio C con la propria chiave e dal messaggio estrae la chiave di sessione client—TGS che utilizza per decriptare il messaggio D (autenticatore). A questo punto invia i seguenti due messaggi al client:
 - Messaggio E: Ticket client-server (che include l'identificativo del client, l'indirizzo di rete del client, il periodo di validità e la chiave di sessione client-server) criptato utilizzando la chiave segreta del server che offre il servizio.
 - Messaggio F: Chiave di sessione client-server criptato usando la chiave di sessione client-TGS

Client: Autenticazione SS

1. Ricevendo i messaggi E e F dal TGS, il client può autenticarsi presso il SS. Il client si connette al SS e invia i seguenti due messaggi:
 - Messaggio E: Ticket client-server criptato usando la chiave segreta di SS.
 - Messaggio G: un nuovo autenticatore, che include l'identificativo del client, il timestamp ed è
2. Il server decripta il ticket usando la sua chiave segreta e invia il seguente messaggio al client per confermare la propria identità e la volontà di fornire il servizio al client:
 - Messaggio H: il timestamp trovato nell'autenticatore incrementato di uno, criptato utilizzando la chiave di sessione client-server.
3. Il client decripta la conferma usando la chiave di sessione client-server e controlla che il timestamp sia correttamente aggiornato. Se lo è, il client può considerare affidabile il server e iniziare a effettuare le richieste di servizio.
4. Il server fornisce i servizi al client.

Capitolo 5

Capitolo4

5.1 Principi di controllo dell'accesso

Due definizioni di controllo dell'accesso sono utili per capire la sua portata.

1. **NISTIR 7298** (Glossario dei termini chiave della sicurezza delle informazioni, maggio 2013), definisce **il controllo dell'accesso** come il processo di concessione o rifiuto di richieste specifiche a:
 - Ottenere e utilizzare le informazioni e i relativi servizi di elaborazione delle informazioni
 - Entrare in specifiche strutture fisiche.
2. **RFC 4949**, Internet Security Glossary, definisce **il controllo dell'accesso** come un processo con cui l'uso delle risorse del sistema è regolato secondo una politica di sicurezza è permesso solo alle entità autorizzate (utenti, programmi, processi o altri sistemi) secondo tale politica.

Possiamo considerare il controllo degli accessi come un elemento centrale della sicurezza informatica. Gli obiettivi principali della sicurezza informatica sono di impedire agli utenti non autorizzati di accesso alle risorse, impedire agli utenti legittimi di accedere alle risorse in modo non autorizzato, e permettere agli utenti legittimi di accedere alle risorse in modo autorizzato.

Table 4.1 Access Control Security Requirements (SP 800-171)

Basic Security Requirements
1 Limit information system access to authorized users, processes acting on behalf of authorized users, or devices (including other information systems).
2 Limit information system access to the types of transactions and functions that authorized users are permitted to execute.
Derived Security Requirements
3 Control the flow of CUI in accordance with approved authorizations.
4 Separate the duties of individuals to reduce the risk of malevolent activity without collusion.
5 Employ the principle of least privilege, including for specific security functions and privileged accounts.
6 Use non-privileged accounts or roles when accessing nonsecurity functions.
7 Prevent non-privileged users from executing privileged functions and audit the execution of such functions.
8 Limit unsuccessful logon attempts.
9 Provide privacy and security notices consistent with applicable CUI rules.
10 Use session lock with pattern-hiding displays to prevent access and viewing of data after period of inactivity.
11 Terminate (automatically) a user session after a defined condition.
12 Monitor and control remote access sessions.
13 Employ cryptographic mechanisms to protect the confidentiality of remote access sessions.
14 Route remote access via managed access control points.
15 Authorize remote execution of privileged commands and remote access to security-relevant information.
16 Authorize wireless access prior to allowing such connections.
17 Protect wireless access using authentication and encryption.
18 Control connection of mobile devices.
19 Encrypt CUI on mobile devices.
20 Verify and control/limit connections to and use of external information systems.
21 Limit use of organizational portable storage devices on external information systems.
22 Control CUI posted or processed on publicly accessible information systems.

CUI = controlled unclassified information

Source: From NIST SP 800-171 Protecting Controlled Unclassified Information in Nonfederal Information Systems and Organizations, December 2016 National Institute of Standards and Technology (NIST), United States Department of Commerce.

In senso lato, tutta la sicurezza informatica riguarda il controllo degli accessi. Infatti, RFC 4949 definisce la sicurezza informatica come segue:

Misure che implementano e assicurano servizi di sicurezza in un sistema informatico, in particolare quelle che assicurano il servizio di controllo degli accessi.

5.1.1 Contesto del controllo dell'accesso

Oltre al controllo d'accesso, questo contesto coinvolge le seguenti entità e funzioni:

- **Autenticazione**

Verifica che le credenziali di un utente o di un'altra entità del sistema siano valide.

- **Autorizzazione**

La concessione di un diritto o di un permesso ad un'entità di sistema per accedere una risorsa del sistema. Questa funzione determina chi è affidabile per un determinato scopo.

- **Audit**

Una revisione ed esame indipendente delle registrazioni e delle attività del sistema al fine di verificare l'adeguatezza dei controlli del sistema, di assicurare la conformità con la politica stabilita e le procedure operative, per rilevare le violazioni della sicurezza, e per raccomandare qualsiasi cambiamento indicato nel controllo, nella politica e nelle procedure.

Un meccanismo di controllo dell'accesso media tra un utente (o un processo che esegue per conto di un utente) e le risorse di sistema, come applicazioni, sistemi operativi, firewall, router, file e database. Il sistema deve prima autenticare un'entità che cerca l'accesso.

Tipicamente, la funzione di autenticazione determina se l'utente è permesso di accedere al sistema. Poi la funzione di controllo dell'accesso determina se l'accesso specifico richiesto da questo utente è permesso. Un amministratore di sicurezza mantiene un database un database di autorizzazioni che specifica quale tipo di accesso a quali risorse è permesso a questo utente. La funzione di controllo degli accessi consulta questo database per se concedere l'accesso. Una funzione di auditing monitora e tiene un registro degli accessi degli utenti alle risorse del sistema.

In pratica, un certo numero di componenti può cooperare per condividere la funzione di controllo la funzione di controllo degli accessi. Tutti i sistemi operativi hanno almeno un rudimentale, e in molti casi un componente di controllo degli accessi abbastanza robusto. I pacchetti di sicurezza aggiuntivi possono integrare le capacità di controllo d'accesso native del sistema operativo. Applicazioni particolari o utilità, come un sistema di gestione di database, incorporano anche funzioni di controllo degli accessi funzioni di controllo degli accessi. Dispositivi esterni, come i firewall, possono anche fornire servizi di controllo dell'accesso.

5.1.2 Politiche di controllo dell'accesso

Una politica di controllo degli accessi, che può essere incorporata in un database di autorizzazioni, quali tipi di accesso sono permessi, in quali circostanze e da chi.

Le politiche di controllo dell'accesso sono generalmente raggruppate nelle seguenti categorie:

- **Controllo dell'accesso discrezionale (DAC)**

Controlla l'accesso in base all'identità del richiedente e su regole di accesso (autorizzazioni) che stabiliscono cosa i richiedenti sono (o non sono) autorizzati a fare. Questa politica è definita discrezionale perché un'entità può avere diritti di accesso che permettono all'entità, di sua spontanea volontà, di un'altra entità di accedere a qualche risorsa.

- **Controllo di accesso obbligatorio (MAC)**

Controlla l'accesso basandosi sul confronto delle etichette di sicurezza (che indicano quanto sono sensibili o critiche le risorse del sistema), con le autorizzazioni di sicurezza (che indicano che le entità del sistema sono autorizzate ad accedere a certe risorse). Questa politica è definita obbligatoria perché un'entità che ha l'autorizzazione di accedere a una risorsa non può, solo per sua volontà, permettere a un'altra entità di accedere a quella risorsa.

- **Controllo di accesso basato sui ruoli (RBAC)**

Controlla l'accesso in base ai ruoli che gli utenti hanno all'interno del sistema e su regole che stabiliscono quali accessi sono permessi agli utenti in determinati ruoli.

- **Controllo di accesso basato sugli attributi (ABAC)**

Controlla l'accesso in base agli attributi dell'utente, della risorsa a cui accedere e delle condizioni ambientali correnti.

Queste quattro politiche non si escludono a vicenda. Un meccanismo di controllo degli accessi può impiegare due o anche tutte e tre queste politiche per coprire diverse classi di risorse di sistema.

5.1.3 Soggetti oggetti e diritti d'accesso

Gli elementi di base del controllo d'accesso sono: soggetto, oggetto e diritto d'accesso.

Un soggetto è un'entità capace di accedere agli oggetti. In generale, il concetto di soggetto equivale a quello di processo. Qualsiasi utente o applicazione ottiene effettivamente l'accesso a un oggetto per mezzo di un processo che rappresenta quell'utente o applicazione. Il processo assume gli attributi dell'utente, come i diritti di accesso.

Un soggetto è tipicamente ritenuto responsabile delle azioni che ha iniziato, e un audit trail può essere usato per registrare l'associazione di un soggetto con azioni rilevanti per la sicurezza eseguite su un oggetto dal soggetto. I sistemi di controllo dell'accesso di base definiscono tipicamente tre classi di soggetti, con diritti di accesso diversi per ogni classe:

- **Proprietario:** può essere il creatore di una risorsa, come un file. Per le risorse di sistema, la proprietà può appartenere ad un amministratore di sistema. Per le risorse del progetto, l'amministratore o il leader di progetto può essere assegnato la proprietà.
- **Gruppo:** In aggiunta ai privilegi assegnati ad un proprietario, ad un gruppo nominato di utenti possono anche essere concessi diritti di accesso, in modo tale che l'appartenenza al gruppo sufficiente per esercitare questi diritti di accesso. Nella maggior parte degli schemi, un utente può appartenere a più gruppi.
- **Mondo:** Il minimo di accesso è concesso agli utenti che sono in grado di accedere al sistema ma non sono inclusi nelle categorie proprietario e gruppo per questa risorsa.

Un oggetto è una risorsa il cui accesso è controllato. In generale, un oggetto è un'entità usata per contenere e/o ricevere informazioni.

Il numero e i tipi di oggetti da proteggere con un sistema di controllo degli accessi dipende dall'ambiente in cui opera il controllo degli accessi e dal compromesso tra sicurezza da un lato e complessità, carico di lavoro e facilità d'uso dall'altro.

Un diritto di accesso descrive il modo in cui un soggetto può accedere a un oggetto. I diritti di accesso potrebbero includere quanto segue:

- **Leggere:** L'utente può visualizzare le informazioni in una risorsa di sistema (ad esempio, un file, record selezionati in un file, campi selezionati all'interno di un record, o qualche combinazione). Lettura l'accesso include la possibilità di copiare o stampare.
- **Scrittura:** L'utente può aggiungere, modificare o cancellare dati in una risorsa di sistema (ad esempio, file, record, programmi). L'accesso in scrittura include l'accesso in lettura.
- **Eseguire:** L'utente può eseguire programmi specifici.

- **Cancellare:** L'utente può cancellare certe risorse di sistema, come file o record.
- **Creare:** L'utente può creare nuovi file, record o campi.
- **Cercare:** L'utente può elencare i file in una directory o altrimenti cercare nella directory.

5.1.4 Controllo dell'accesso discrezionale

Come si è detto in precedenza, uno schema di controllo di accesso discrezionale è uno schema in cui un'entità può ricevere diritti di accesso che le permettono, per sua volontà, di permettere a un'altra entità di accedere a qualche risorsa. Un approccio generale al DAC, come esercitato da un sistema operativo o da un sistema di gestione di database, è quello di una matrice di accesso.

Una dimensione della matrice consiste in soggetti identificati che possono tentare l'accesso alle risorse.

Tipicamente, questa lista consisterà di singoli utenti o gruppi di utenti anche se l'accesso potrebbe essere controllato per terminali, apparecchiature di rete, host, o applicazioni invece di o in aggiunta agli utenti. L'altra dimensione elenca gli oggetti a cui si può accedere. Al massimo livello di dettaglio, gli oggetti possono essere singoli dati campi di dati. Raggruppamenti più aggregati, come record, file o anche l'intero database, possono anche essere oggetti nella matrice. Ogni voce nella matrice indica i diritti di accesso di un particolare soggetto per un particolare oggetto.

		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

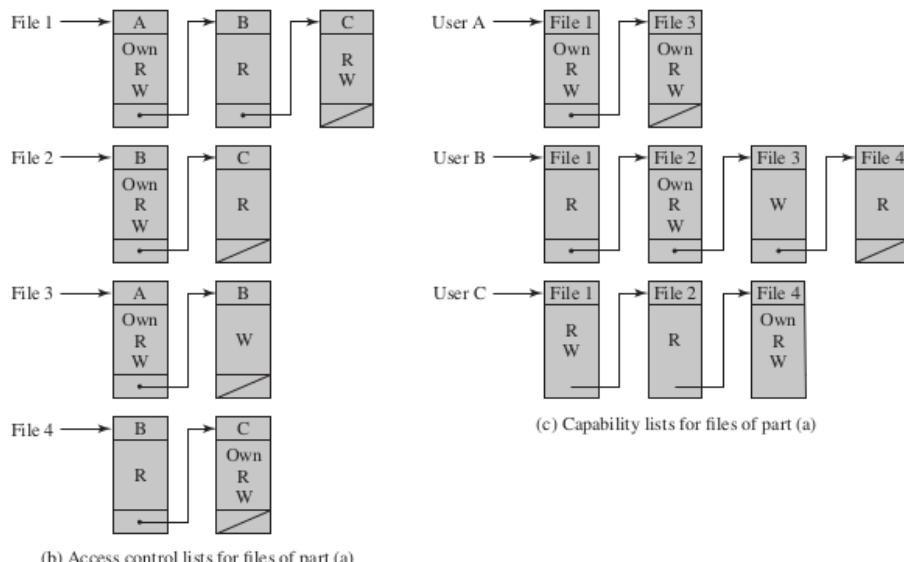


Figure 4.2 Example of Access Control Structures

Così, l'utente A possiede i file 1 e 3 e ha diritti di accesso in lettura e scrittura a questi file.

L'utente B ha diritti di accesso in lettura al file 1, e così via.

In pratica, una matrice di accesso è di solito sparsa e viene implementata con la decomposizione in uno dei due modi. La matrice può essere decomposta per colonne, ottenendo liste di controllo degli accessi (ACL) (vedi Figura 4.2b). Per ogni oggetto, una ACL elenca gli utenti e i loro diritti di accesso consentiti. L'ACL può contenere una voce di default, o pubblica. Questo permette agli utenti che non sono esplicitamente elencati come aventi diritti speciali di avere un set di diritti. L'insieme predefinito di diritti dovrebbe sempre seguire la regola del minimo privilegio o accesso in sola lettura, a seconda del caso. Gli elementi della lista possono includere singoli utenti così come gruppi di utenti.

Quando si vuole determinare quali soggetti hanno quali diritti di accesso ad una particolare risorsa, le ACL sono convenienti, perché ogni ACL fornisce le informazioni per

una data risorsa. Tuttavia, questa struttura di dati non è conveniente per determinare i diritti di accesso disponibili per uno specifico utente.

La decomposizione per righe produce i capability ticket (vedi Figura 4.2c).

Un capability è un ticket di capacità specifica gli oggetti e le operazioni autorizzate per un particolare utente. Ogni utente ha un certo numero di ticket e può essere autorizzato a prestarli o darli ad altri. Poiché i ticket possono essere dispersi nel sistema, presentano un problema di sicurezza maggiore rispetto alle liste di controllo degli accessi. L'integrità del ticket deve essere protetta e garantita (di solito dal sistema operativo). In particolare, il ticket deve essere non falsificabile. Un modo per ottenere ciò è quello di avere il sistema operativo che tiene tutti i ticket per conto degli utenti. Questi biglietti dovrebbero essere tenuti in una regione di memoria inaccessibile agli utenti. Un'altra alternativa è includere un token non falsificabile nella capacità. Questo potrebbe essere una grande password casuale, o un codice crittografico di autenticazione del messaggio. Questo valore è verificato dalla risorsa ogni volta che viene richiesto l'accesso. □

Una tabella di autorizzazione contiene una riga per un diritto di accesso di un soggetto ad una risorsa. Ordinare o accedere alla tabella per soggetto è equivalente a una lista di capacità. Ordinare o l'accesso alla tabella per oggetto è equivalente ad una ACL. Un database relazionale può facilmente implementare una tabella di autorizzazione di questo tipo.

Table 4.2 Authorization Table for Files in Figure 4.2

Subject	Access Mode	Object
A	Own	File 1
A	Read	File 1
A	Write	File 1
A	Own	File 3
A	Read	File 3
A	Write	File 3
B	Read	File 1
B	Own	File 2
B	Read	File 2
B	Write	File 2
B	Write	File 3
B	Read	File 4
C	Read	File 1
C	Write	File 1
C	Read	File 2
C	Own	File 4
C	Read	File 4
C	Write	File 4

5.1.5 Un modello di controllo d'accesso

Questa sezione introduce un modello generale per DAC sviluppato da Lampson, Graham, e Denning. Il modello presuppone un insieme di soggetti, un insieme di oggetti e un insieme di regole che governano l'accesso dei soggetti agli oggetti. Definiamo lo stato di protezione di un sistema come l'insieme di informazioni, in un dato momento, che specifica i diritti di accesso per ogni soggetto rispetto a ogni oggetto.

Noi Possiamo identificare tre requisiti: rappresentare lo stato di protezione, far rispettare i diritti di accesso e permettere ai soggetti di alterare lo stato di protezione in certi modi. Il modello affronta tutti e tre i requisiti, dando una descrizione generale e logica di un sistema DAC.

Per rappresentare lo stato di protezione, estendiamo l'universo di oggetti nella matrice di controllo degli accessi per includere quanto segue:

- **Processi:** I diritti di accesso includono la capacità di cancellare un processo, fermare (bloccare) e svegliare un processo.
- **Dispositivi:** I diritti di accesso includono la capacità di leggere/scrivere il dispositivo, di controllare il suo funzionamento (ad esempio, una ricerca su disco), e di bloccare/sbloccare il dispositivo per l'uso.
- **Luoghi o regioni di memoria:** I diritti di accesso includono la capacità di leggere/scrivere certe regioni di memoria che sono protette in modo tale che il default è di disabilitare l'accesso.
- **Soggetti:** I diritti di accesso rispetto ad un soggetto hanno a che fare con la capacità di concedere o cancellare i diritti di accesso di quel soggetto ad altri oggetti, come spiegato successivamente.

La figura 4.3 è un esempio. Per una matrice di controllo di accesso A, ogni voce $A[S, X]$ contiene stringhe, chiamate attributi di accesso, che specificano i diritti di accesso del soggetto S all per l'oggetto X. Per esempio, nella figura 4.3, S1 può leggere il file F1, perché 'read' appare in $A[S1, F1]$. Da un punto di vista logico o funzionale, un modulo di controllo degli accessi separato è associato ad ogni tipo di oggetto (vedi figura 4.4). Il modulo valuta ogni richiesta di un soggetto di accedere a un oggetto per determinare se il diritto di accesso esiste. Un tentativo di accesso innesca i seguenti passi:

1. Un soggetto S0 emette una richiesta di tipo a per l'oggetto X.
2. La richiesta fa sì che il sistema (il sistema operativo o un modulo di interfaccia di controllo degli accessi di qualche tipo) a generare un messaggio della forma $(S0, a, X)$ al controllore per X.
3. Il controllore interroga la matrice di accesso A per determinare se a è in $A[S0, X]$.

		OBJECTS								
		Subjects			Files		Processes		Disk drives	
		<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>F</i> ₁	<i>F</i> ₂	<i>P</i> ₁	<i>P</i> ₂	<i>D</i> ₁	<i>D</i> ₂
SUBJECTS	<i>S</i> ₁	control	owner	owner control	read*	read owner	wakeup	wakeup	seek	owner
	<i>S</i> ₂		control		write*	execute			owner	seek*
	<i>S</i> ₃			control		write	stop			

* = copy flag set

Figure 4.3 Extended Access Control Matrix

In caso affermativo, l'accesso è permesso; in caso contrario, l'accesso è negato e si verifica una violazione della protezione si verifica una violazione della protezione. La violazione dovrebbe innescare un avvertimento e un'azione appropriata.

La figura 4.4 suggerisce che ogni accesso di un soggetto ad un oggetto è mediato dal controllore per quell'oggetto, e che la decisione del controllore è basata sul contenuto attuale della matrice. Inoltre, alcuni soggetti hanno l'autorità di apportare modifiche specifiche alla matrice di accesso. Una richiesta di modifica della matrice di accesso è trattata come un accesso alla matrice, con le singole voci della matrice trattate come oggetti.

Tali accessi sono mediati da un controllore della matrice di accesso, che controlla gli aggiornamenti alla matrice. Il modello include anche un insieme di regole che governano le modifiche alla matrice di accesso come mostrato nella tabella 4.3. A questo scopo, introduciamo i diritti di accesso "proprietario e 'controllo' e il concetto di flag di copia, come spiegato nei paragrafi successivi.

Le prime tre regole riguardano il trasferimento, la concessione e la cancellazione dei diritti di accesso. Supponiamo che la voce a* esista in A[S0, X]. Questo significa che S0 ha il diritto di accesso a al soggetto X e, a causa della presenza del flag di copia, può trasferire questo diritto, con o senza con o senza flag di copia, ad un altro soggetto. La regola R1 esprime questa capacità. Un soggetto potrebbe trasferire il diritto di accesso senza il flag di copia se ci fosse la preoccupazione che il nuovo soggetto potrebbe trasferire maliziosamente il diritto ad un altro soggetto che non dovrebbe avere quel diritto di accesso. Per esempio, S1 può mettere 'read' o 'read*' in qualsiasi voce della matrice in nella colonna F1. La regola R2 afferma che se S0 è designato come proprietario dell'oggetto X, allora S0 può concedere un diritto di accesso a quell'oggetto per qualsiasi altro soggetto. La regola R2

afferma che se S0 è designato come proprietario dell'oggetto X, allora S0 può concedere un diritto di accesso a quell'oggetto per qualsiasi altro soggetto.

La regola R2 afferma che S0 può aggiungere qualsiasi diritto di accesso ad A[S, X] per qualsiasi S, se S0 ha accesso "proprietario" a X. La regola R3 permette a S0 di cancellare qualsiasi diritto di accesso da qualsiasi voce della matrice in una riga per la quale S0 controlla il soggetto, e per qualsiasi voce della matrice in una colonna per la quale S0 possiede l'oggetto.

La regola R4 permette ad un soggetto di leggere quella porzione di matrice che possiede o controlla.

Le restanti regole della tabella 4.3 regolano la creazione e la cancellazione di soggetti e oggetti.

La regola R5 afferma che ogni soggetto può creare un nuovo oggetto, che possiede, e può quindi concedere e cancellare l'accesso all'oggetto. Secondo la regola R6, il proprietario di un oggetto può distruggere l'oggetto, con la conseguente cancellazione della colonna corrispondente della matrice di accesso.

La regola R7 permette a qualsiasi soggetto di creare un nuovo soggetto; il creatore è proprietario del nuovo soggetto e il nuovo soggetto ha accesso di controllo su se stesso.

La regola R8 permette al proprietario di un soggetto di cancellare la riga e la colonna (se ci sono colonne di soggetti) della matrice di accesso designata da quel soggetto.

L'insieme di regole nella Tabella 4.3 è un esempio dell'insieme di regole che potrebbero essere definite per un sistema di controllo degli accessi. I seguenti sono esempi di regole aggiuntive o alternative

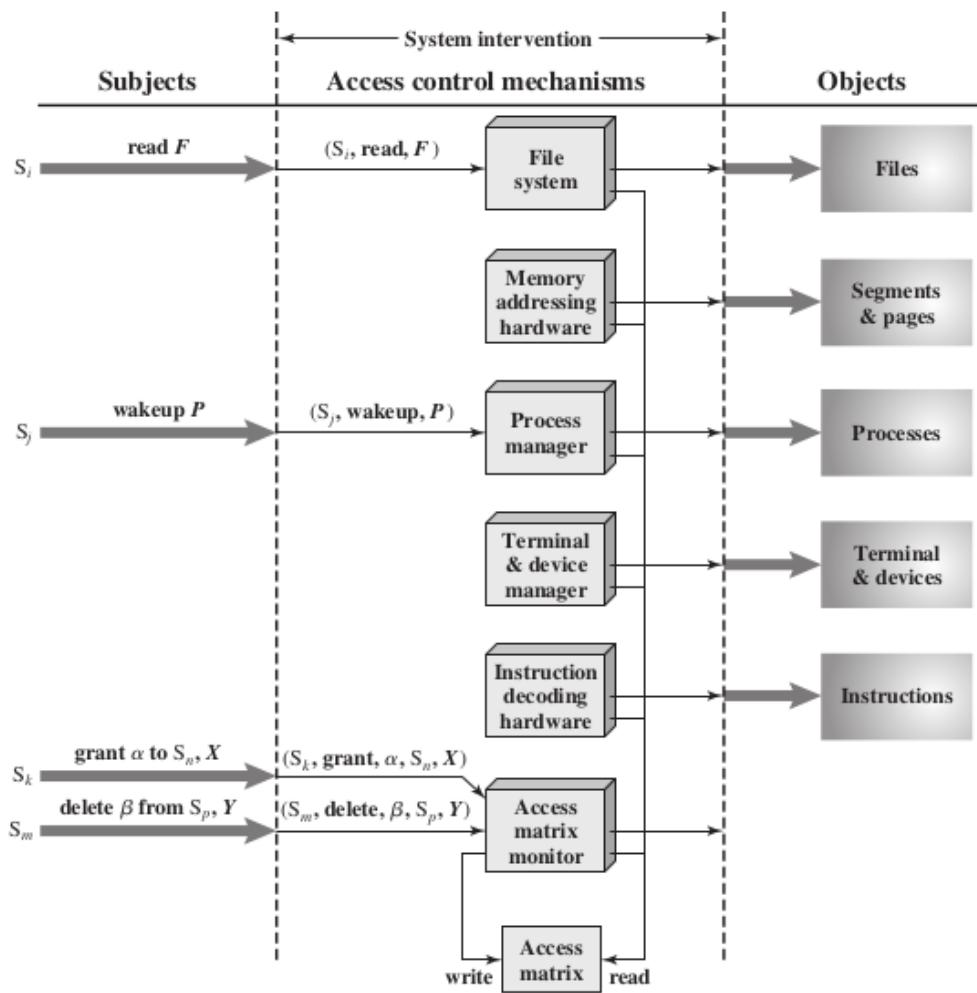


Figure 4.4 An Organization of the Access Control Function

Table 4.3 Access Control System Commands

Rule	Command (by S_0)	Authorization	Operation
R1	transfer $\left\{ \begin{array}{l} \alpha^* \\ \alpha \end{array} \right\}$ to S, X	‘ α^* ’ in $A[S_0, X]$	store $\left\{ \begin{array}{l} \alpha^* \\ \alpha \end{array} \right\}$ in $A[S, X]$
R2	grant $\left\{ \begin{array}{l} \alpha^* \\ \alpha \end{array} \right\}$ to S, X	‘owner’ in $A[S_0, X]$	store $\left\{ \begin{array}{l} \alpha^* \\ \alpha \end{array} \right\}$ in $A[S, X]$
R3	delete α from S, X	‘control’ in $A[S_0, S]$ or ‘owner’ in $A[S_0, X]$	delete α from $A[S, X]$
R4	$w \leftarrow \text{read } S, X$	‘control’ in $A[S_0, S]$ or ‘owner’ in $A[S_0, X]$	copy $A[S, X]$ into w
R5	create object X	None	add column for X to A ; store ‘owner’ in $A[S_0, X]$
R6	destroy object X	‘owner’ in $A[S_0, X]$	delete column for X from A
R7	create subject S	none	add row for S to A ; execute create object S ; store ‘control’ in $A[S, S]$
R8	destroy subject S	‘owner’ in $A[S_0, S]$	delete row for S from A ; execute destroy object S

5.2 Esempio: Controllo di accesso ai file Unix

Tutti i tipi di file UNIX sono amministrati dal sistema operativo per mezzo di inode.

Un **inode** (nodo indice) è una struttura di controllo che contiene le informazioni chiave necessarie al sistema operativo per un particolare file. Diversi nomi di file possono essere associati ad un singolo inode, ma un inode attivo è associato esattamente ad un file, e ogni file è controllato esattamente da un inode. Gli attributi del file così come i suoi permessi e altre informazioni di controllo sono memorizzati nell'inode. Sul disco, c'è una tabella di inode, o lista di inode, che contiene gli inode di tutti i file nel file sistema. Quando un file viene aperto, il suo inode viene portato nella memoria principale e memorizzato in una tabella di inode residente in memoria.

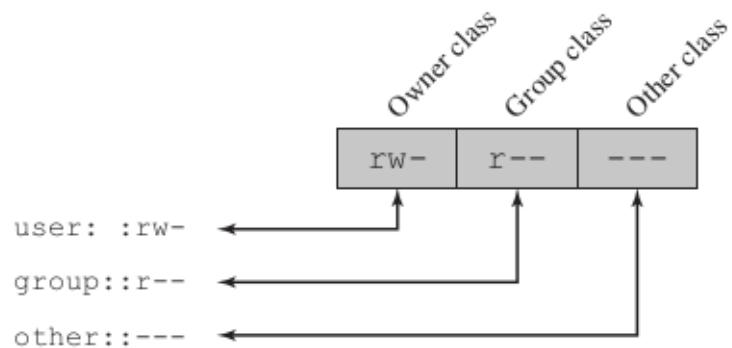
Le directory sono strutturate in un albero gerarchico. Ogni directory può contenere file e/o altre directory. Una directory che si trova all'interno di un'altra directory viene una sottodirectory. Una directory è semplicemente un file che contiene una lista di nomi di file più puntatori agli inode associati. Così, associato ad ogni directory c'è il proprio inode.

5.2.1 Controllo di accesso ai file UNIX tradizionale

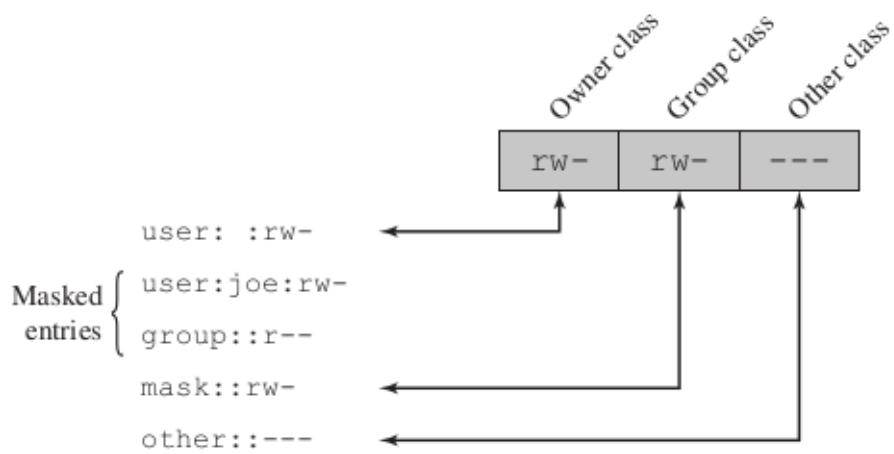
La maggior parte dei sistemi UNIX dipende, o almeno si basa, sullo schema di controllo dell'accesso ai file introdotto con le prime versioni di UNIX. Ad ogni utente UNIX viene assegnato un unico numero di identificazione utente (ID utente). Un utente è anche membro di un gruppo primario, e possibilmente di un certo numero di altri gruppi, ciascuno identificato da un ID di gruppo. Quando un file viene creato, è designato come di proprietà di un particolare utente e contrassegnato dall'ID di quell'utente **ID DI QUELL'UTENTE.**

Appartiene anche a un gruppo specifico, che inizialmente è o il gruppo primario del suo creatore, o il gruppo della sua directory madre se questa ha il permesso SetGID impostato. Associato ad ogni file c'è un insieme di 12 bit di protezione. L'ID del proprietario, l'ID del gruppo e i bit di protezione fanno parte dell'inode del file.

Nove dei bit di protezione specificano i permessi di lettura, scrittura ed esecuzione per il proprietario del file, gli altri membri del gruppo a cui questo file appartiene e tutti gli altri utenti. Questi formano una gerarchia di proprietario, gruppo e tutti gli altri, con l'insieme di permessi più alto che viene usato. La figura 4.5a mostra un esempio in cui il proprietario del file ha accesso in lettura e scrittura; tutti gli altri membri del gruppo del file hanno accesso in lettura; e gli utenti esterni al gruppo non hanno diritti di accesso al file. Quando sono applicati ad una directory, i bit di lettura e scrittura garantiscono il diritto di elencare e creare/rinominare/cancellare file nella directory. Il bit di esecuzione garantisce il diritto di scendere nella directory o di cercare un nome di file.



(a) Traditional UNIX approach (minimal access control list)



(b) Extended access control list

Figure 4.5 UNIX File Access Control

I tre bit rimanenti definiscono uno speciale comportamento aggiuntivo per i file o le directory. Due di questi sono i permessi "set user ID" (SetUID) e "set group ID" (SetGID) permessi. Se questi sono impostati su un file eseguibile, il sistema operativo funziona come segue.

Quando un utente (con privilegi di esecuzione per questo file) esegue il file, il sistema alloca temporaneamente i diritti dell'ID dell'utente del creatore del file, o del gruppo del file, rispettivamente, a quelli dell'utente che esegue il file. Questi sono conosciuti come "ID utente effettivo" e "ID gruppo effettivo" e sono usati in aggiunta all'"ID utente reale" e all'"ID gruppo reale". "ID" e "ID gruppo reale" dell'utente in esecuzione quando si prendono decisioni di controllo dell'accesso per questo programma. Questa modifica è efficace solo mentre il programma è in esecuzione.

Questa caratteristica permette la creazione e l'uso di programmi privilegiati che possono utilizzare file normalmente inaccessibili agli altri utenti. Permette agli utenti di accedere a certi file in modo controllato. In alternativa, quando applicato ad una directory, il permesso SetGID indica che i file appena creati erediteranno il gruppo di questa directory.

Il permesso SetUID viene ignorata.

L'ultimo bit di permesso è il bit "appiccicoso". Quando è impostato su un file, originariamente indica che il sistema dovrebbe mantenere il contenuto del file in memoria dopo l'esecuzione. Questo non è più usato. Quando è applicato ad una directory, però, specifica che solo il proprietario di qualsiasi file nella directory può rinominare, spostare o cancellare quel file. Questo è utile per gestire i file nelle directory temporanee condivise.

Un particolare ID utente è designato come "superutente". Il superutente è esente dai soliti vincoli di controllo dell'accesso ai file e ha accesso a tutto il sistema. Qualsiasi programma che è posseduto da, e SetUID a, il "superutente" garantisce potenzialmente un accesso illimitato al sistema accesso al sistema a qualsiasi utente che esegua quel programma.

Quindi è necessaria una grande attenzione quando si scrivono tali programmi.

Questo schema di accesso è adeguato quando i requisiti di accesso ai file si allineano con gli utenti e un numero modesto di gruppi di utenti. Per esempio, supponiamo che un utente voglia dare l'accesso in lettura per il file X agli utenti A e B, e l'accesso in lettura per il file Y agli utenti B e C.

Avremmo bisogno di almeno due gruppi di utenti, e l'utente B dovrebbe appartenere ad entrambi i gruppi per poter accedere ai due file. Tuttavia, se c'è un gran numero di diversi raggruppamenti di utenti che richiedono una serie di diritti di accesso a diversi file, allora potrebbe essere necessario un numero molto grande di gruppi può essere necessario per fornire questo. Questo diventa rapidamente ingombrante e difficile da gestire, se possibile.

Un modo per superare questo problema è usare le liste di controllo degli accessi, che sono fornite nella maggior parte dei moderni sistemi UNIX. Un ultimo punto da notare è che il tradizionale schema di controllo di accesso ai file UNIX implementa una semplice struttura di dominio di protezione. Un dominio è associato con l'utente, e cambiare il dominio corrisponde a cambiare temporaneamente l'ID dell'utente.

5.2.2 Liste di controllo d'accesso in UNIX

Molti moderni sistemi operativi UNIX e basati su UNIX supportano le liste di controllo degli accessi, inclusi FreeBSD, OpenBSD, Linux e Solaris. In questa sezione, descriviamo FreeBSD, ma altre implementazioni hanno essenzialmente le stesse caratteristiche e la stessa interfaccia. La caratteristica è indicata come lista di controllo di accesso estesa, mentre il tradizionale approccio UNIX tradizionale è indicato come lista di controllo dell'accesso minima.

FreeBSD permette all'amministratore di assegnare una lista di ID utente UNIX e ad un file usando il comando setfacl. Qualsiasi numero di utenti e gruppi può essere associato ad un file, ognuno con tre bit di protezione (lettura, scrittura, esecuzione), offrendo un meccanismo flessibile per l'assegnazione dei diritti di accesso. Un file non ha bisogno di avere un ACL ma può essere protetto solo dal tradizionale meccanismo di accesso ai file UNIX. I file FreeBSD includono un ulteriore bit di protezione che indica se il file ha una ACL estesa.

FreeBSD e la maggior parte delle implementazioni UNIX che supportano le ACL estese usano la seguente strategia (ad esempio, Figura 4.5b):

1. La classe del proprietario e le altre voci di classe nel campo dei permessi a 9 bit hanno lo stesso stesso significato che nel caso dell'ACL minima.
2. La voce group class specifica i permessi per il gruppo proprietario di questo file.

Questi permessi rappresentano i permessi massimi che possono essere assegnati a utenti nominati o gruppi nominati, diversi dall'utente proprietario. In quest'ultimo ruolo, la voce classe di gruppo funziona come una maschera.

3. Altri utenti e gruppi nominati possono essere associati al file

Ognuno con un campo di autorizzazione a 3 bit. I permessi elencati per un utente o un gruppo gruppo sono confrontati con il campo della maschera. Qualsiasi permesso per l'utente o il gruppo gruppo che non è presente nel campo della maschera non è permesso.

Quando un processo richiede l'accesso ad un oggetto del file system, vengono eseguiti due passi.

1. **Passo seleziona la voce ACL che più si avvicina al processo richiedente.**

Il siti ACL vengono esaminate nel seguente ordine: proprietario, utenti nominati, gruppi (proprietari o con nome) gruppi, altri. Solo una singola voce determina l'accesso.

2. **Passo controlla se la voce corrispondente contiene sufficienti permessi.**

Un processo può essere membro di più di un gruppo; quindi più di una voce di gruppo può corrispondere. Se una di queste voci di gruppo corrispondenti gruppo corrispondente contiene i permessi richiesti, ne viene scelto uno che contiene i permessi richiesti

(il risultato è lo stesso indipendentemente dalla voce scelta). Se nessuna delle delle voci di gruppo corrispondenti contiene i permessi richiesti, l'accesso sarà negato indipendentemente dalla voce scelta.

5.3 Controllo d'accesso basato sul ruolo

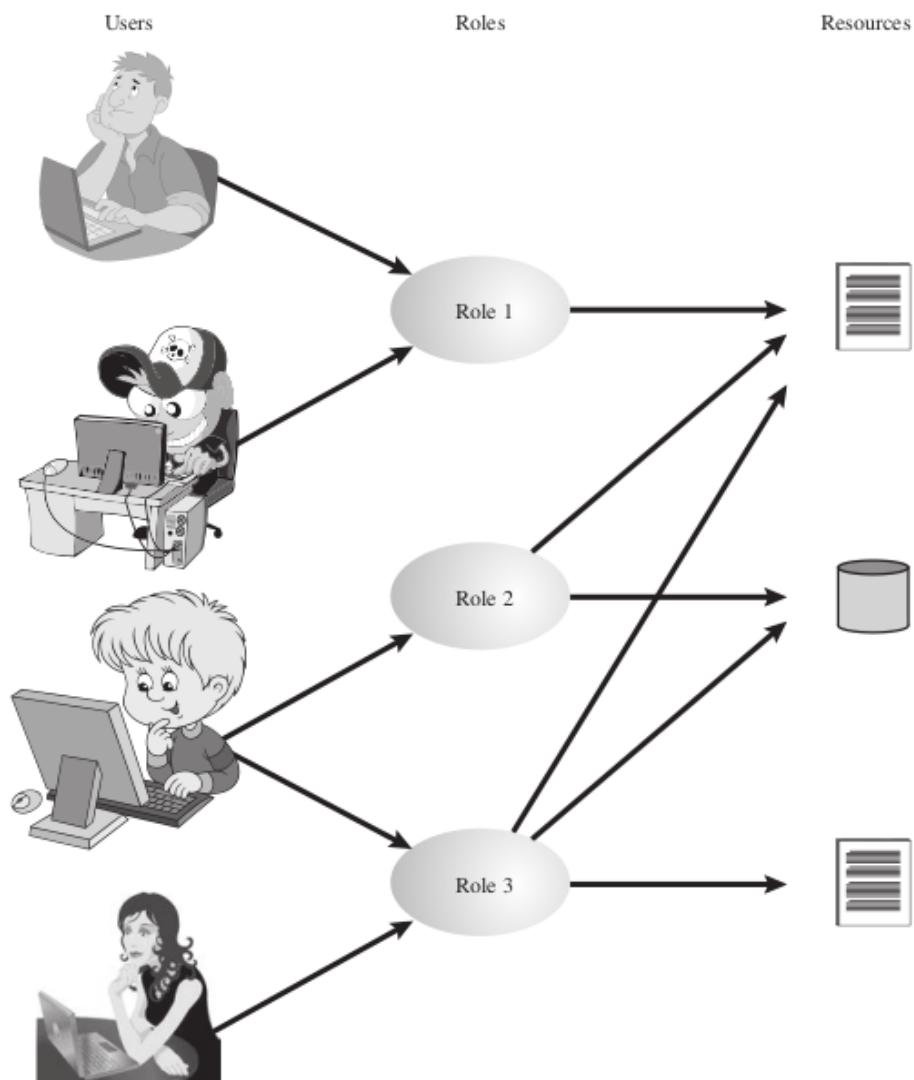


Figure 4.6 **Users, Roles, and Resources**

	R ₁	R ₂	• • •	R _n					
U ₁	X								
U ₂	X								
U ₃		X		X					
U ₄				X					
U ₅				X					
U ₆				X					
•									
•									
•									
U _m	X								
OBJECTS									
	R ₁	R ₂	R _n	F ₁	F ₂	P ₁	P ₂	D ₁	D ₂
R ₁	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
R ₂		control		write *	execute			owner	seek *
•									
•									
•									
R _n			control		write	stop			

Figure 4.7 Access Control Matrix Representation of RBAC

I sistemi DAC tradizionali definiscono i diritti di accesso dei singoli utenti e dei gruppi di utenti. Al contrario, RBAC si basa sui ruoli che gli utenti assumono in un sistema piuttosto che sull'identità dell'utente. Tipicamente, i modelli RBAC definiscono un ruolo come una funzione lavorativa all'interno un'organizzazione. I sistemi RBAC assegnano i diritti di accesso ai ruoli invece che ai singoli utenti. A loro volta, gli utenti sono assegnati a diversi ruoli, sia staticamente che dinamicamente, secondo le loro responsabilità.

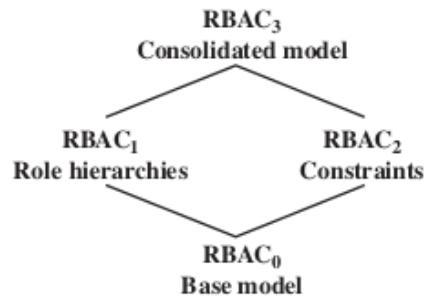
RBAC ora gode di un ampio uso commerciale e rimane un'area di ricerca attiva. Il National Institute of Standards and Technology (NIST) ha emesso uno standard, FIPS PUB 140-3 (Security Requirements for Cryptographic Modules, settembre 2009), che

richiede il supporto per il controllo degli accessi e l'amministrazione attraverso i ruoli. La relazione degli utenti con i ruoli è molti a molti, così come la relazione dei ruoli alle risorse o agli oggetti del sistema (vedi Figura 4.6). L'insieme degli utenti cambia, in alcuni ambienti frequentemente, e l'assegnazione di un utente a uno o più ruoli può anche essere dinamico. L'insieme dei ruoli nel sistema nella maggior parte degli ambienti è relativamente statico, con solo occasionali aggiunte o cancellazioni. Ogni ruolo avrà diritti di accesso specifici a una o più risorse. L'insieme delle risorse e i diritti di accesso specifici associati con un particolare ruolo è probabile che cambino raramente.

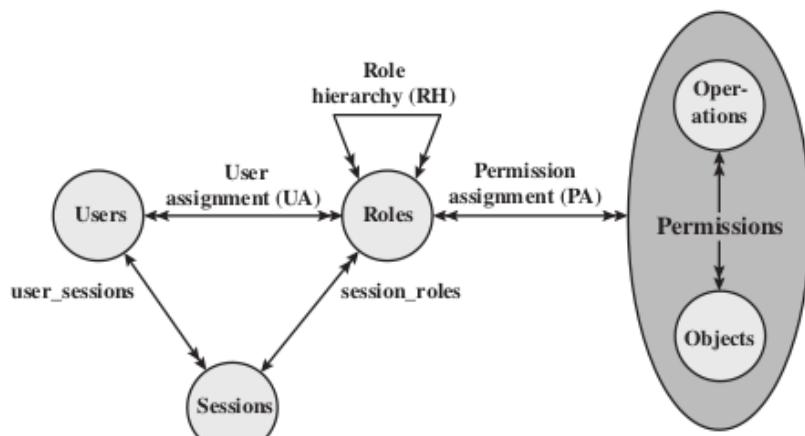
Possiamo usare la rappresentazione della matrice di accesso per rappresentare gli elementi chiave di un sistema RBAC in termini semplici, come mostrato nella figura 4.7. La matrice superiore mette in relazione i singoli utenti ai ruoli. Tipicamente ci sono molti più utenti che ruoli. Ogni matrice voce della matrice è vuota o marcata, quest'ultima indica che l'utente è assegnato a questo ruolo. Si noti che un singolo utente può essere assegnato a più ruoli (più di un segno in una riga) e più utenti possono essere assegnati a un singolo ruolo (più di un segno in una colonna). La matrice inferiore ha la stessa struttura della matrice di controllo degli accessi DAC, con i ruoli come soggetti. Tipicamente, ci sono pochi ruoli e molti oggetti, o risorse. In questa matrice, le voci sono i diritti di accesso specifici dei ruoli. Si noti che un ruolo può essere trattato come un oggetto, permettendo la definizione di gerarchie di ruoli.

RBAC si presta ad un'efficace implementazione del principio del minimo privilegio, a cui si fa riferimento nel capitolo 1. Ogni ruolo dovrebbe contenere l'insieme minimo di diritti di accesso necessari per quel ruolo. Un utente viene assegnato ad un ruolo che gli permette di eseguire solo ciò che è richiesto per quel ruolo. Più utenti assegnati allo stesso ruolo godono dello stesso insieme minimo di diritti di accesso.

5.3.1 Modelli di riferimento RBAC



(a) Relationship among RBAC models



(b) RBAC models

Figure 4.8 A Family of Role-Based Access Control Models RBAC₀ is the minimum requirement for an RBAC system. RBAC₁ adds role hierarchies and RBAC₂ adds constraints. RBAC₃ includes RBAC₁ and RBAC₂.

Table 4.4 Scope RBAC Models

Models	Hierarchies	Constraints
RBAC ₀	No	No
RBAC ₁	Yes	No
RBAC ₂	No	Yes
RBAC ₃	Yes	Yes

Una varietà di funzioni e servizi possono essere inclusi sotto l'approccio generale RBAC approccio. Per chiarire i vari aspetti di RBAC, è utile definire un insieme di modelli astratti modelli astratti di funzionalità RBAC.

Definisce una famiglia di modelli di riferimento che è servita come base per gli sforzi di standardizzazione in corso. Questa famiglia consiste di quattro modelli che sono correlati tra loro, come mostrato nella Figura 4.8a e nella Tabella 4.4. RBAC 0 contiene la funzionalità minima per un sistema RBAC.

RBAC 1 include le funzionalità di RBAC 0 e aggiunge le gerarchie dei ruoli, che permettono a un ruolo di ereditare i permessi da un altro ruolo. RBAC 2 include RBAC 0 e aggiunge vincoli, che limitano i modi in cui i componenti di un sistema RBAC possono essere configurati. RBAC 3 contiene la funzionalità di RBAC 0, RBAC1 e RBAC2.

Modello base-RBAC0 Figura 4.8b, senza la gerarchia dei ruoli e i vincoli, contiene i quattro tipi di entità in un sistema RBAC 0:

- **Utente:** un individuo che ha accesso a questo sistema informatico. Ogni individuo ha un ID utente associato.
- **Ruolo:** Una funzione di lavoro nominata all'interno dell'organizzazione che controlla questo sistema informatico. Tipicamente, associato ad ogni ruolo c'è una descrizione dell'autorità e della responsabilità conferite a questo ruolo, e a qualsiasi utente che assume questo ruolo.
- **Permesso:** Un'approvazione di una particolare modalità di accesso a uno o più oggetti. Termini equivalenti sono diritto di accesso, privilegio e autorizzazione.
- **Sessione:** Una mappatura tra un utente e un sottoinsieme attivato dell'insieme di ruoli a cui l'utente è assegnato.

Le linee con le frecce nella figura 4.8b indicano relazioni, o mappature, con una freccia singola che ne indica una e una doppia che ne indica molte. Quindi, c'è una relazione molti-a-molti tra utenti e ruoli: Un utente può avere più ruoli, e più utenti possono essere assegnati a un singolo ruolo. Allo stesso modo, c'è una relazione molti a molti tra ruoli e permessi. Una sessione è usata per definire una relazione temporanea uno-a-molti tra un utente e uno o più ruoli a cui l'utente è stato assegnato. L'utente stabilisce una sessione con solo i ruoli necessari per un particolare compito, questo è un esempio del concetto di minimo privilegio. Le relazioni molti-a-molti tra utenti e ruoli e tra ruoli e permessi forniscono una flessibilità e granularità di assegnazione che non si trova negli schemi DAC convenzionali. Senza questa flessibilità e granularità, c'è un rischio maggiore che ad un utente possa essere concesso più accesso alle risorse di quanto sia necessario a causa del controllo limitato sui tipi di accesso che possono essere permessi.

Gerarchie di ruolo-RBAC1

Le gerarchie di ruolo forniscono un mezzo per riflettere la struttura gerarchica dei ruoli in un'organizzazione. Tipicamente, le funzioni lavorative con maggiore responsabilità hanno maggiore autorità per accedere alle risorse. Una funzione lavorativa subordinata può avere un sottoinsieme dei diritti di accesso della funzione lavorativa superiore. Le gerarchie di ruolo fanno uso del concetto di ereditarietà per permettere ad un ruolo di includere implicitamente i diritti di accesso associati ad un ruolo subordinato. La figura 4.9 è un esempio di diagramma di una gerarchia di ruoli. Per convenzione, i ruoli subordinati sono più in basso nel diagramma. Una linea tra due ruoli implica che il ruolo superiore include tutti i diritti di accesso del ruolo inferiore, così come altri diritti di accesso non disponibili al ruolo inferiore. Un ruolo può ereditare diritti di accesso da più ruoli subordinati. Per esempio, nella figura 4.9, il ruolo Project Lead include tutti i diritti di accesso del ruolo Production Engineer e del ruolo Quality Engineer. Più di un ruolo può ereditare dallo stesso ruolo subordinato. Per esempio, sia il ruolo Production Engineer che il ruolo Quality Engineer includono tutti i diritti di accesso del ruolo Engineer. Ulteriori diritti di accesso sono anche assegnati al ruolo Production Engineer, e un diverso insieme di diritti di accesso aggiuntivi sono assegnati al ruolo Quality ingegnere di qualità. Quindi, questi due ruoli hanno diritti di accesso che si sovrappongono, vale a dire i diritti di accesso diritti di accesso che condividono con il ruolo Engineer.

Vincoli-RBAC2

I vincoli forniscono un mezzo per adattare RBAC alle specifiche delle politiche amministrative e di sicurezza in un'organizzazione. Un vincolo è una relazione definita tra i ruoli o una condizione relativa ai ruoli. I ruoli mutuamente esclusivi sono ruoli tali che un utente può essere assegnato ad un solo ruolo nell'insieme. Questa limitazione potrebbe essere statica, o potrebbe essere dinamica, nel senso che ad un utente potrebbe essere assegnato solo uno dei ruoli nell'insieme per una sessione. Il vincolo reciprocamente esclusivo supporta una separazione di doveri e capacità all'interno un'organizzazione. Questa separazione può essere rafforzata o migliorata dall'uso di assegnazioni di permessi reciprocamente esclusivi. Con questo vincolo aggiuntivo, un insieme di ruoli mutuamente esclusivi ha le seguenti proprietà:

Un utente può essere assegnato ad un solo ruolo nell'insieme (sia durante una sessione staticamente). Qualsiasi permesso (diritto di accesso) può essere concesso ad un solo ruolo dell'insieme.

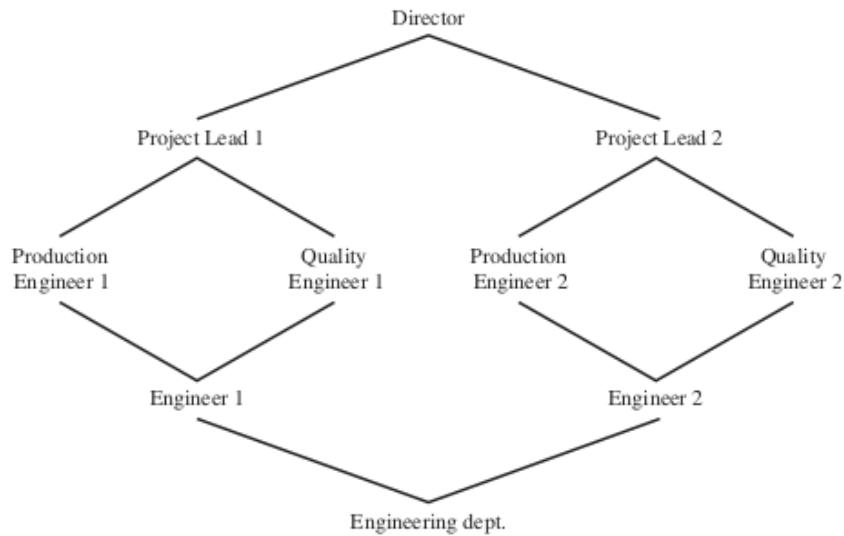


Figure 4.9 Example of Role Hierarchy

Così, l'insieme dei ruoli reciprocamente esclusivi hanno permessi non sovrapposti. Se due utenti sono assegnati a ruoli diversi nell'insieme, allora gli utenti hanno permessi non sovrapposti mentre assumono quei ruoli. Lo scopo dei ruoli mutuamente esclusivi è quello di aumentare la difficoltà di collusione tra individui con competenze diverse o funzioni lavorative divergenti per contrastare le politiche di sicurezza. La cardinalità si riferisce all'impostazione di un numero massimo rispetto ai ruoli. Uno di questi vincolo è quello di impostare un numero massimo di utenti che possono essere assegnati ad un dato ruolo.

Per esempio, un ruolo di capo progetto o un ruolo di capo reparto potrebbe essere limitato ad un singolo utente. Il sistema potrebbe anche imporre un vincolo sul numero di ruoli che un utente è assegnato, o il numero di ruoli che un utente può attivare per una singola sessione. Un'altra forma di vincolo è quella di impostare un numero massimo di ruoli che possono essere concessi un particolare permesso; questa potrebbe essere una tecnica desiderabile di mitigazione del rischio per un permesso sensitiva o potente.

Un sistema potrebbe essere in grado di specificare un ruolo prerequisito, che impone che un utente possa essere assegnato a un particolare ruolo solo se è già assegnato a qualche altro ruolo specificato.

Un prerequisito può essere usato per strutturare l'implementazione del concetto di minimo privilegio. In una gerarchia, potrebbe essere richiesto che un utente possa essere assegnato ad un ruolo (superiore) solo se gli è già stato assegnato un ruolo immediatamente inferiore.

Per esempio, nella figura 4.9 un utente assegnato a un ruolo di Project Lead deve essere assegnato anche a ai ruoli subordinati Production Engineer e Quality Engineer. Quindi, se l'utente non ha bisogno di tutti i permessi del ruolo Project Lead per un dato compito,

l'utente può invocare una sessione usando solo il ruolo subordinato richiesto. Si noti l'uso di prerequisiti legati al concetto di gerarchia richiede il modello RBAC 3.

5.4 Controllo degli accessi basato su attributi

Uno sviluppo relativamente recente nella tecnologia di controllo dell'accesso è il modello di controllo dell'accesso basato sugli attributi (ABAC). Un modello ABAC può definire autorizzazioni che esprimono condizioni su proprietà sia della risorsa che del soggetto. Per esempio, consideriamo una configurazione in cui ogni risorsa ha un attributo che identifica il soggetto che ha creato la risorsa. Quindi, una singola regola di accesso può specificare il privilegio del proprietario per tutti i creatori di ogni risorsa. La forza dell'approccio ABAC è la sua flessibilità e potenza espressiva. Sottolinea che il principale ostacolo alla sua adozione in sistemi reali è stata la preoccupazione per l'impatto della valutazione dei predicati su entrambe le proprietà della risorsa e dell'utente per ogni accesso.

Tuttavia, per applicazioni come i servizi Web cooperanti e il cloud computing questo aumento del costo delle prestazioni è meno evidente perché c'è già un costo di prestazione relativamente alto per ogni accesso. Così, i servizi Web sono stati tecnologie innovative per l'implementazione di modelli ABAC, specialmente attraverso l'introduzione dell'eXtensible Access Control Markup Language (XAMCL) e c'è un notevole interesse nell'applicare il modello ABAC ai servizi cloud.

Ci sono tre elementi chiave in un modello ABAC: gli attributi, che sono definiti per le entità in una configurazione; un modello di policy, che definisce le politiche ABAC; e il modello di architettura, che si applica alle politiche che impongono il controllo degli accessi.

5.4.1 Attributi

Gli attributi sono caratteristiche che definiscono aspetti specifici del soggetto, dell'oggetto, delle condizioni ambientali e/o delle operazioni richieste che sono predefinite e preassegnate da un'autorità. Gli attributi contengono informazioni che indicano la classe di informa un nome e un valore.

I seguenti sono i tre tipi di attributi nel modello ABAC:

- **Attributi soggetto**

Un soggetto è un'entità attiva (ad esempio, un utente, un'applicazione, un processo o un dispositivo) che causa il flusso di informazioni tra gli oggetti o cambia lo stato del sistema. Ogni soggetto ha degli attributi associati che definiscono l'identità e le caratteristiche del soggetto. Tali attributi possono includere l'identificatore del soggetto identificatore, nome, organizzazione, titolo di lavoro e così via. Anche il ruolo di un soggetto può essere visto come un attributo.

- **Attributi dell'oggetto**

Un oggetto, chiamato anche risorsa, è un oggetto passivo (nel contesto della richiesta data) un'entità legata al sistema informativo (ad esempio, dispositivi, file, record, tabelle, processi, programmi, reti, domini) che contengono o ricevere informazioni.

- **Attributi ambientali**

Questi attributi sono stati finora largamente ignorati nella maggior parte delle politiche di controllo degli accessi. Essi descrivono l'ambiente operativo, tecnico e anche ambiente o contesto situazionale in cui avviene l'accesso alle informazioni. Per esempio, attributi come la data e l'ora correnti, le attività correnti di virus/hacker e il livello di sicurezza della rete (ad esempio, Internet o intranet), non sono associati ad un particolare soggetto o ad una risorsa, ma possono comunque essere rilevanti nell'applicazione di una politica di controllo degli accessi. ABAC è un modello di controllo dell'accesso logico che si distingue perché controlla l'accesso agli oggetti valutando le regole contro gli attributi delle entità (soggetto e oggetto), le operazioni e oggetto), delle operazioni e dell'ambiente rilevanti per una richiesta. ABAC si basa sulla valutazione degli attributi del soggetto, degli attributi dell'oggetto e di una relazione forzata o una regola di controllo dell'accesso che definisce le operazioni consentite per le combinazioni di attributi di soggetto e oggetto in un dato ambiente.

Tutte le soluzioni ABAC contengono queste capacità di base per valutare gli attributi e applicare regole o relazioni tra questi attributi. I sistemi ABAC sono in grado di applicare i concetti DAC, RBAC e concetti MAC. ABAC consente un controllo dell'accesso a grana fine, che permette un numero numero di input discreti in una decisione di controllo dell'accesso, fornendo un più grande insieme di possibili combinazioni di quelle variabili per riflettere un insieme più ampio e definitivo di possibili regole, politiche o restrizioni di accesso. Così, ABAC permette un numero illimitato numero illimitato di attributi da

combinare per soddisfare qualsiasi regola di controllo dell'accesso. Inoltre, i sistemi ABAC possono essere implementati per soddisfare una vasta gamma di requisiti da dalle liste di controllo degli accessi di base a modelli di policy avanzati ed espressivi che sfruttano appieno la flessibilità dell'ABAC.

5.4.2 Architettura logica ABAC

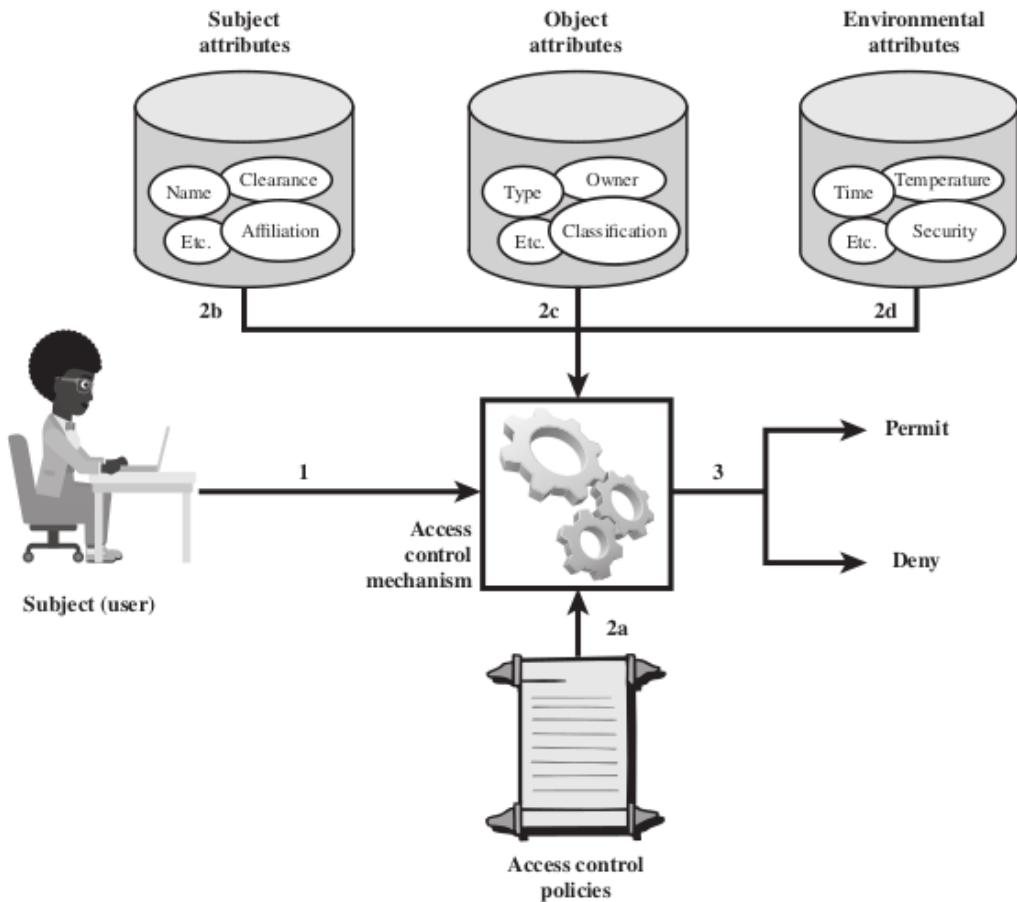


Figure 4.10 ABAC Scenario

La figura 4.10 illustra in un'architettura logica i componenti essenziali di un sistema ABAC.

Un accesso di un soggetto a un oggetto procede secondo i seguenti passi:

1. Un soggetto richiede l'accesso ad un oggetto. Questa richiesta viene inoltrata ad un meccanismo di controllo meccanismo di controllo dell'accesso.
2. Il meccanismo di controllo degli accessi è governato da un insieme di regole (2a) che sono definite da una politica di controllo dell'accesso preconfigurata. Sulla base di queste regole, il meccanismo di controllo dell'accesso valuta gli attributi del soggetto (2b), dell'oggetto (2c) e le condizioni ambientali (2d) per determinare l'autorizzazione.

3. Il meccanismo di controllo dell'accesso concede al soggetto l'accesso all'oggetto se l'accesso è autorizzato, e nega l'accesso se non è autorizzato. È chiaro dall'architettura logica che ci sono quattro fonti indipendenti di informazioni utilizzate per la decisione di controllo dell'accesso.

Il progettista del sistema può decidere quali attributi sono importanti per il controllo dell'accesso rispetto a soggetti, oggetti e condizioni ambientali. Il progettista del sistema o altra autorità può quindi definire politiche di controllo dell'accesso, sotto forma di regole, per qualsiasi combinazione desiderata di altri di soggetti, oggetti e condizioni ambientali.

Dovrebbe essere evidente che questo approccio è molto potente e flessibile. Tuttavia, il costo, sia in termini di complessità della progettazione e dell'implementazione, e in termini di impatto sulle prestazioni, è probabile che superi quello di altri approcci di controllo dell'accesso. Questo è un compromesso che autorità di sistema deve fare.

Rispetto a un modello DAC che usa liste di controllo degli accessi (ACL). Questa figura non solo illustra la complessità relativa dei due modelli, ma chiarisce anche i requisiti di fiducia dei due modelli. Un confronto delle relazioni di fiducia rappresentative (indicate dalle linee con la freccia) per l'uso di ACL e ABAC mostra che ci sono molte relazioni di fiducia più complesse richieste per ABAC per funzionare correttamente. Ignorando i punti in comune in entrambe le parti della Figura 4.11, si può osservare che con le ACL la radice della fiducia è con il proprietario dell'oggetto, il quale applica in che fa rispettare le regole di accesso all'oggetto fornendo l'accesso all'oggetto attraverso l'aggiunta di un utente ad una ACL.

L'aggiunta di un utente ad una ACL. In ABAC, la radice della fiducia deriva da molte fonti di cui il proprietario dell'oggetto non ha controllo, sviluppatori di policy e emittenti di credenziali. Di conseguenza, SP 800-162 raccomandava che un organismo di governance aziendale sia formato per gestire tutte le identità, le credenziali, di gestione delle identità, delle credenziali e degli accessi e che ogni organizzazione sub-ordinata mantenga un organismo simile per garantire la coerenza nella gestione l'implementazione e il cambiamento di paradigma associati all'implementazione di ABAC a livello aziendale.

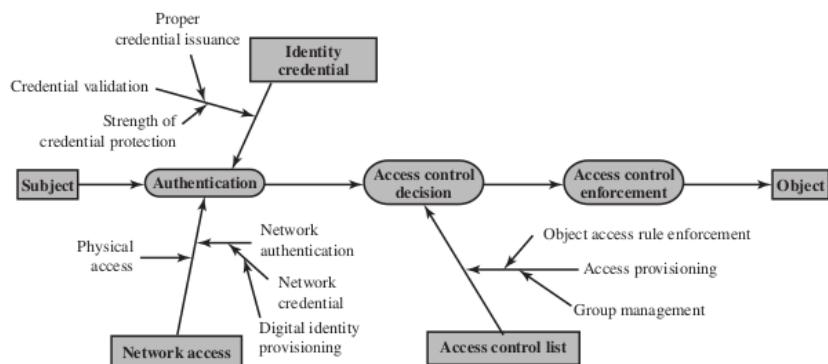
Inoltre, si raccomanda che un'impresa sviluppi un modello di fiducia che può essere usato per illustrare le relazioni di fiducia e aiutare a determinare la proprietà e la responsabilità delle informazioni e dei servizi, le esigenze di ulteriori politiche e di governance e i requisiti per soluzioni tecniche per convalidare o applicare le relazioni di fiducia. Il modello di fiducia di modello di fiducia può essere usato per influenzare le organizzazioni a condividere le loro informazioni con chiare aspettative su come queste informazioni saranno usate e protette e per essere in grado di fidarsi delle informazioni e delle asserzioni di attributi e autorizzazioni provenienti da altre organizzazioni.

5.4.3 Politiche ABAC

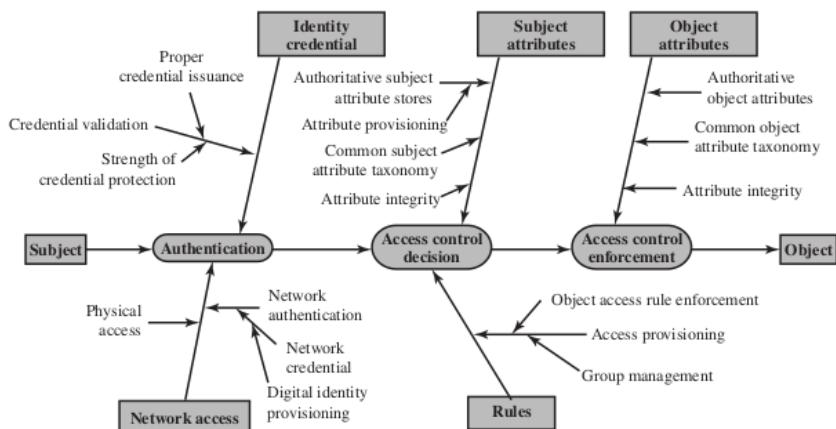
Una politica è un insieme di regole e relazioni che governano il comportamento consentito all'interno di un'organizzazione. Basato sui privilegi dei soggetti e su come le risorse o gli oggetti devono essere protetti in quali condizioni ambientali. A loro volta, i privilegi rappresentano il comportamento autorizzato di un soggetto; sono definiti da un'autorità e incorporati in una politica. Altri termini comunemente usati al posto di privilegi sono autorizzazioni e diritti. La politica è tipicamente scritta dal punto di vista dell'oggetto da proteggere e dei privilegi disponibili ai soggetti. Ora definiamo un modello di policy

ABAC. Sono utilizzate le seguenti convenzioni

S, O ed E sono rispettivamente soggetti, oggetti e ambienti
 SAk ($1 \leq k \leq K$), OAm ($1 \leq m \leq M$), e EAn ($1 \leq n \leq N$) sono gli attributi predefiniti per soggetti, oggetti e ambienti, rispettivamente



(a) ACL Trust Chain



(b) ABAC Trust Chain

Figure 4.11 ACL and ABAC Trust Relationships

$\text{ATTR}(s)$, $\text{ATTR}(o)$, e $\text{ATTR}(e)$ sono relazioni di assegnazione di attributi per il soggetto s, oggetto o, e ambiente e, rispettivamente:

$$\begin{aligned}\text{ATTR}(s) &\subseteq \text{SA}_1 \times \text{SA}_2 \times \dots \times \text{SA}_K \\ \text{ATTR}(r) &\subseteq \text{OA}_1 \times \text{OA}_2 \times \dots \times \text{OA}_M \\ \text{ATTR}(o) &\subseteq \text{EA}_1 \times \text{EA}_2 \times \dots \times \text{EA}_N\end{aligned}$$

Usiamo anche la notazione di funzione per l'assegnazione del valore dei singoli attributi.
Per esempio:

```
Role(s) = "Service Consumer"
ServiceOwner(o) = "XYZ, Inc."
CurrentDate(e) = "01-23-2005"
```

Nella forma più generale, una Policy Rule, che decide se un soggetto s può accedere ad un oggetto o in un particolare ambiente e, è una funzione booleana degli attributi di s, o ed e:

Inserire immagine

Una base di regole di policy o policy store può consistere in un certo numero di regole di policy, che coprono molti soggetti e oggetti all'interno di un dominio di sicurezza. Il controllo dell'accesso Il processo decisionale del controllo d'accesso equivale essenzialmente alla valutazione delle regole di nell'archivio delle politiche.

Ora considerate l'esempio di un negozio di intrattenimento online che trasmette film agli utenti per una tariffa mensile fissa. Useremo questo esempio per contrastare gli approcci RBAC e ABAC approcci. Il negozio deve applicare la seguente politica di controllo degli accessi basata sull'età all'età dell'utente e alla classificazione del contenuto del film:

Movie Rating	Users Allowed Access
R	Age 17 and older
PG-13	Age 13 and older
G	Everyone

In un modello RBAC, ad ogni utente verrebbe assegnato uno dei tre ruoli: Adulto, Juvenile, o Child, possibilmente durante la registrazione. Ci sarebbero tre permessi creati: Può vedere film vietati ai minori, Può vedere film vietati ai minori, e Può vedere i film vietati ai minori. Il ruolo Adulto viene assegnato con tutti e tre i permessi.

Il ruolo Juvenile ottiene le autorizzazioni Can view PG-13-rated movies e Can view G-rated movies e il ruolo Bambino ottiene solo il permesso di visualizzare i film vietati ai minori.

Entrambe le assegnazioni utente-ruolo e permesso-ruolo sono compiti amministrativi manuali. L'approccio ABAC a questa applicazione non ha bisogno di definire esplicitamente i ruoli. Invece, se un utente u può accedere o vedere un film m (in un ambiente di sicurezza e che qui viene ignorato) verrebbe risolto valutando una regola di policy come la seguente:

```
R1:can_access(u, m, e) ←
  (Age(u) ≥ 17 ∧ Rating(m) ∈ {R, PG-13, G}) ∨
  (Age(u) ≥ 13 ∧ Age(u) < 17 ∧ Rating(m) ∈ {PG-13, G}) ∨
  (Age(u) < 13 ∧ Rating(m) ∈ {G})
```

dove Age e Rating sono rispettivamente l'attributo soggetto e l'attributo oggetto. Il vantaggio del modello ABAC mostrato qui è che elimina la definizione e la gestione di ruoli statici, eliminando così la necessità di compiti amministrativi per l'assegnazione da utente a ruolo e da permesso a ruolo.

Il vantaggio di ABAC si vede più chiaramente quando imponiamo politiche a grana più fine. Per esempio, supponiamo che i film siano classificati come New Release o Old Release, in base alla data di rilascio rispetto alla data corrente, e che gli utenti siano classificati come classificati come Utente Premium e Utente Regolare, in base alla tariffa che pagano.

Vorremmo applicare una politica per cui solo gli utenti premium possono vedere i nuovi film. Per il modello RBAC, dovremmo raddoppiare il numero di ruoli, per distinguere ogni utente per età e tariffa, e dovremmo raddoppiare il numero di permessi separati pure. In generale, se ci sono K attributi soggetto e M attributi oggetto, e se per ogni attributo, $\text{Range}()$ denota l'intervallo di valori possibili che può assumere, allora il rispettivo numero di ruoli e permessi richiesti per un modello RBAC sono:

$$\prod_{k=1}^K \text{Range}(SA_k) \text{ and } \prod_{m=1}^M \text{Range}(SA_m)$$

Così, possiamo vedere che quando il numero di attributi aumenta per ospitare politiche a grana più fine, il numero di ruoli e permessi cresce esponenzialmente.

Al contrario, il modello ABAC tratta gli attributi aggiuntivi in modo efficiente. Per questo esempio, la policy R1 definita in precedenza è ancora valida. Abbiamo bisogno di due nuove regole:

```
R2:can_access(u, m, e) ←
  (MembershipType(u) = Premium) ∨
  (MembershipType(u) = Regular ∧ MovieType(m) = OldRelease)
R3:can_access(u, m, e) ← R1 ∧ R2
```

Con il modello ABAC, è anche facile aggiungere attributi ambientali. Supponiamo che vogliamo aggiungere una nuova regola di policy che è espressa in parole come segue: Gli utenti regolari sono autorizzati a vedere le nuove release nei periodi promozionali. Questo sarebbe difficile da esprimere in un modello RBAC. In un modello ABAC, abbiamo solo bisogno di aggiungere una regola congiuntiva (AND) che controlla per vedere se l'attributo ambientale la data di oggi cade in un periodo promozionale.

5.5 Gestione dell'entità, delle credenziali e dell'accesso

L'ICAM è un approccio completo alla gestione e all'implementazione delle identità digitali identità digitali (e gli attributi associati), le credenziali e il controllo degli accessi. ICAM è stato sviluppato dal governo degli Stati Uniti, ma è applicabile non solo alle agenzie governative, ma può anche essere distribuito da imprese che cercano un approccio unificato al controllo degli accessi. ICAM è progettato per:

- Creare rappresentazioni fidate di identità digitali di individui e di ciò che i documenti ICAM si riferiscono a entità non personali (NPE). Queste ultime includono processi, applicazioni e dispositivi automatici che cercano di accedere a una risorsa.
- Legare queste identità a credenziali che possono servire come proxy per l'individuo o NPE nelle transazioni di accesso. Una credenziale è un oggetto o una struttura di dati che lega autorevolmente un'identità (e optionalmente, attributi aggiuntivi) a un token posseduto e controllato da un sottoscrittore.
- Utilizza le credenziali per fornire un accesso autorizzato alle risorse di un'agenzia.

5.5.1 Gestione dell'identità

La gestione dell'identità si occupa di assegnare attributi a un'identità digitale e di collegare tale identità digitale a un individuo o a una NPE. L'obiettivo è quello di stabilire un'identità digitale affidabile che sia indipendente da una specifica applicazione o contesto.

L'approccio tradizionale, e ancora più comune, al controllo dell'accesso per applicazioni e programmi è quello di creare una rappresentazione digitale di un'identità per l'uso specifico di l'applicazione o il programma. Di conseguenza, il mantenimento e la protezione dell'identità stessa è trattata come secondaria rispetto alla missione associata all'applicazione. Inoltre, c'è una considerevole sovrapposizione di sforzi nello stabilire queste identità specifiche dell'applicazione.

A differenza degli account usati per accedere a reti, sistemi o applicazioni, i record di identità aziendali non sono legati al titolo di lavoro, alle mansioni lavorative, all'ubicazione o al fatto che sia necessario l'accesso a un sistema specifico. Questi elementi possono diventare attributi legati ad un record di identità e possono anche diventare parte di ciò che identifica in modo univoco un individuo in una specifica applicazione. Le decisioni sul controllo degli accessi saranno basate sul contesto e sugli attributi rilevanti di un utente non solo sulla sua identità. Il concetto di un'identità aziendale è che gli individui avranno una singola rappresentazione digitale di se stessi che può essere sfruttata in tutti i dipartimenti e le agenzie per molteplici scopi, compreso il controllo degli accessi.

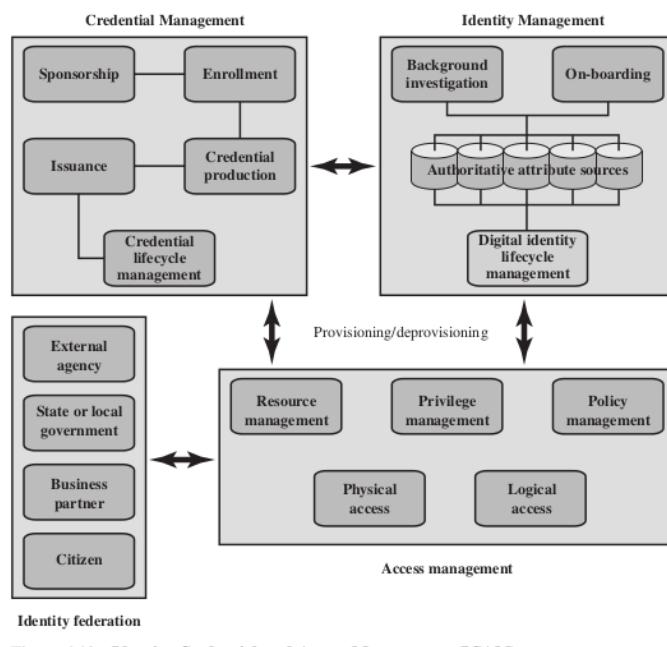


Figure 4.12 Identity, Credential, and Access Management (ICAM)

La figura 4.12 illustra le funzioni chiave coinvolte nella gestione delle identità. Un'identità digitale inizia tipicamente con la raccolta di dati di identità come parte di un processo di iscrizione. Un'identità digitale è spesso composta da un insieme di attributi che aggregati identificano in modo univoco un utente all'interno di un sistema o di un'azienda. Al fine di stabilire la fiducia nell'individuo rappresentato da un'identità digitale, un'agenzia può anche condurre un'indagine di fondo.

Gli attributi di un individuo possono essere memorizzati in varie fonti autorevoli all'interno di un'agenzia e collegati per formare una visione aziendale dell'identità digitale. Questa identità digitale può quindi essere fornita in applicazioni per supportare l'accesso fisico e logico (parte della gestione degli accessi) e de-provisionata quando l'accesso non è più richiesto.

Un elemento finale della gestione delle identità è la gestione del ciclo di vita, include quanto segue:

- Meccanismi, politiche e procedure per proteggere l'identità personale informazioni
- Controllo dell'accesso ai dati di identità
- Tecniche per condividere dati di identità autorevoli con le applicazioni che ne hanno bisogno
- Revoca di un'identità aziendale

5.5.2 Gestione delle credenziali

Come detto, una credenziale è un oggetto o una struttura di dati che lega autorevolmente un'identità (e optionalmente, attributi aggiuntivi) ad un token posseduto e controllato da un sottoscrittore. Esempi di credenziali sono le smart card, le chiavi crittografiche private/pubbliche crittografiche private/pubbliche e i certificati digitali. La gestione delle credenziali è la gestione del ciclo di vita ciclo di vita della credenziale.

La gestione delle credenziali comprende i seguenti cinque componenti logistici:

1. Un individuo autorizzato sponsorizza un individuo o un'entità per una credenziale per stabilire la necessità della credenziale. Per esempio, un supervisore di reparto sponsorizza un dipendente del dipartimento.
2. L'individuo sponsorizzato si iscrive per la credenziale, un processo che tipicamente consiste in una prova d'identità e nella cattura di dati biografici e di dati di sicurezza. un processo che tipicamente consiste nella prova dell'identità e nell'acquisizione di dati biografici e biometrici. Questo passo può anche comportare l'incorporazione di dati di attributo autorevoli, mantenuti dal componente di gestione dell'identità.
3. Viene prodotta una credenziale. A seconda del tipo di credenziale, la produzione può coinvolgere la crittografia, l'uso di una firma digitale, la produzione di una smartcard, o altre funzioni.
4. La credenziale viene rilasciata all'individuo o alla NPE.
5. Una credenziale deve essere mantenuta durante il suo ciclo di vita, che potrebbe includere la revoca, la riemissione/sostituzione, la reiscrizione, la scadenza, la reimpostazione del numero di identificazione personale (PIN), sospensione o reintegrazione.

5.5.3 Gestione degli accessi

Il componente di gestione degli accessi si occupa della gestione e del controllo delle modalità di accesso alle risorse da parte delle entità. Copre sia l'accesso logico che fisiologico e può essere interno ad un sistema o un elemento esterno. Lo scopo della gestione degli accessi è quello di garantire che venga fatta la corretta verifica dell'identità quando un individuo tenta di accedere a edifici, sistemi informatici o dati sensibili alla sicurezza. La funzione di controllo degli accessi fa uso delle credenziali presentate da chi richiede l'accesso e l'identità digitale del richiedente.

Sono necessari tre elementi di supporto per una struttura di controllo degli accessi a livello aziendale:

- **Gestione delle risorse**

Questo elemento riguarda la definizione di regole per una risorsa che richiede il controllo dell'accesso. Le regole includeranno le credenziali requisiti delle credenziali e quali attributi dell'utente, attributi della risorsa e condizioni ambientali sono richieste per l'accesso ad una data risorsa per una data funzione.

- **Gestione dei privilegi**

Questo elemento si occupa di stabilire e mantenere di diritti o attributi di privilegio che comprendono il profilo di accesso di un individuo. Questi attributi rappresentano caratteristiche di un individuo che possono essere utilizzati come base per determinare le decisioni di accesso alle risorse fisiche e logiche. I privilegi sono considerati attributi che possono essere collegati a un'identità digitale. identità digitale.

- **Gestione delle politiche**

Questo elemento governa ciò che è permesso e non permesso in una transazione di accesso. Cioè, dati l'identità e gli attributi del richiedente, gli attributi della risorsa o dell'oggetto e le condizioni ambientali, una politica specifica quali azioni questo utente può eseguire su questo oggetto.

5.5.4 Federazione delle identità

La federazione di identità affronta due questioni:

1. Come vi fidate delle identità di individui di organizzazioni esterne che hanno bisogno di accesso ai vostri sistemi?
2. Come garantisci le identità degli individui della tua organizzazione quando hanno bisogno di collaborare con organizzazioni esterne?

La federazione delle identità è un termine usato per descrivere la tecnologia, gli standard, le politiche e processi che permettono a un'organizzazione di fidarsi di identità digitali, attributi di identità e credenziali create ed emesse da un'altra organizzazione.

5.6 Strutte di fiducia

I concetti interconnessi di fiducia, identità e attributi sono diventati preoccupazioni fondamentali delle imprese Internet, dei fornitori di servizi di rete e delle grandi imprese.

Queste preoccupazioni possono essere viste chiaramente nell'ambiente del commercio elettronico. Per l'efficienza, la privacy e la semplicità legale, le parti delle transazioni generalmente applicano il principio del need-to-know: cosa si deve sapere di qualcuno per trattare con lui? La risposta varia da caso a caso, e include attributi come il numero di registrazione professionale o di licenza, l'organizzazione e il dipartimento, l'ID del personale, il nulla osta di sicurezza, il numero di riferimento del cliente, il numero di carta di credito, l'identificatore unico della salute, le allergie, il gruppo sanguigno, il numero di previdenza sociale, l'indirizzo, lo stato di cittadino, il nickname dei social network, lo pseudonimo e così via. Gli attributi di un individuo che devono essere conosciuti e verificati per permettere una transazione dipendono dal contesto.

5.6.1 Approccio tradizionale allo scambio di identità

Le transazioni online o in rete che coinvolgono parti di diverse organizzazioni, o tra un'organizzazione e un utente individuale come un cliente online, richiedono generalmente la condivisione di informazioni di identità. Queste informazioni possono includere una serie di attributi associati oltre a un semplice nome o identificatore numerico. Sia la parte che divulgla le informazioni che quella che le riceve devono avere un livello di fiducia sulle questioni di sicurezza e di privacy relative a tali informazioni. La figura 4.13a mostra la tecnica tradizionale per lo scambio di informazioni sull'identità. Questo comporta che gli utenti sviluppino accordi con un fornitore di servizi di identità per procurarsi identità e credenziali digitali, e accordi con le parti che forniscono servizi e applicazioni per gli utenti finali e che sono disposti a fare affidamento sull'identità e sulle informazioni sull'identità e le credenziali generate dal fornitore di servizi d'identità. L'accordo della Figura 4.13a deve soddisfare una serie di requisiti. La parte fidata richiede che l'utente sia stato autenticato con un certo grado di sicurezza, che gli attributi imputati all'utente dal fornitore di servizi d'identità siano accurati, e che il fornitore di servizi d'identità sia autorevole per quegli attributi. Il fornitore di servizi d'identità richiede l'assicurazione di avere informazioni accurate sull'utente e che, se condivide le informazioni, la parte che si affida le userà in accordo con i termini e le condizioni contrattuali e la legge. L'utente richiede

l'assicurazione che al fornitore di servizi d'identità e alla parte fidata possano essere affidate informazioni sensibili e che si attengano alle preferenze dell'utente e rispettino la sua privacy. Soprattutto, tutte le parti vogliono sapere se le pratiche descritte dalle altre parti sono effettivamente quelle attuate dalle parti, e quanto sono affidabili quelle parti.

5.6.2 Approccio di fiducia per l'identità aperta

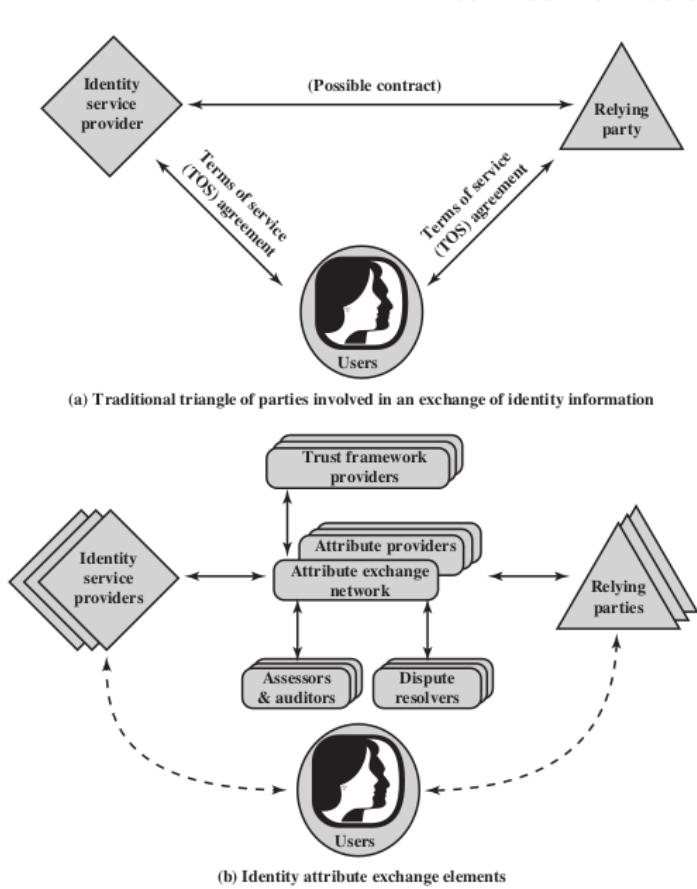


Figure 4.13 Identity Information Exchange Approaches

Senza qualche standard e quadro universale, la disposizione della Figura 4.13a deve essere replicata in molteplici contesti. Un approccio di gran lunga preferibile è quello di sviluppare un approccio aperto e standardizzato per lo scambio di identità e attributi affidabili. Nel resto di questa sezione, esamineremo un tale approccio che sta guadagnando sempre più consenso. Sfortunatamente, questo argomento è gravato da numerosi acronimi, quindi è meglio iniziare con una definizione del più importante di questi:

- **OpenID**

Questo è uno standard aperto che permette agli utenti di essere autenticati da alcuni siti cooperanti (noti come Relying Parties) utilizzando un servizio di terze parti, eliminando la necessità per i webmaster di fornire i propri sistemi ad hoc e permettendo agli utenti di consolidare le loro identità digitali. Gli utenti possono creare account con i loro fornitori di identità OpenID preferiti, quindi utilizzare tali account come base per accedere a qualsiasi sito web che accetti l'autenticazione OpenID.

- **OIDF**

La OpenID Foundation è un'organizzazione internazionale senza scopo di lucro di persone e aziende impegnate ad abilitare, promuovere e proteggere le tecnologie OpenID. OIDF assiste la comunità fornendo l'infrastruttura necessaria struttura e aiuto nel promuovere e sostenere l'adozione estesa di OpenID.

- **ICF**

La Information Card Foundation è una comunità senza scopo di lucro di aziende e individui che lavorano insieme per far evolvere l'ecosistema Information Card. Le carte d'informazione sono identità digitali personali che le persone possono usare online, e la componente chiave dei metasistemi di identità. Visivamente, ogni Information Card ha un'immagine a forma di carta e un nome di carta associato ad essa che permette alle persone di organizzare le loro identità digitali e di selezionare facilmente quella che vogliono usare per qualsiasi interazione.

- **OITF**

La Information Card Foundation è una comunità senza scopo di lucro di aziende e individui che lavorano insieme per far evolvere l'ecosistema Information Card. Le carte d'informazione sono identità digitali personali che le persone possono usare online, e la componente chiave dei metasistemi di identità. Visivamente, ogni Information Card ha un'immagine a forma di carta e un nome di carta associato ad essa che permette alle persone di organizzare le loro identità digitali e di selezionare facilmente quella che vogliono usare per qualsiasi interazione.

- **OIX**

L'Open Identity Exchange Corporation è un fornitore internazionale indipendente e neutrale internazionale di strutture di fiducia per la certificazione conformi al modello Open Identity Trust Frameworks.

- **AXN**

Un Attribute Exchange Network (AXN) è un gateway online su scala Internet per i fornitori di servizi d'identità e per le parti che si affidano a loro per accedere in modo efficiente agli attributi di identità online affermati dall'utente, autorizzati e verificati in volumi a costi accessibili.

I gestori del sistema devono potersi fidare del fatto che gli attributi associati a un soggetto o un oggetto siano autorevoli e vengano scambiati in modo sicuro. Un approccio per fornire tale fiducia all'interno di un'organizzazione è il modello ICAM, in particolare i componenti ICAM (vedi figura 4.12). Combinato con una funzionalità di federazione di identità che è condivisa con altre organizzazioni, gli attributi possono essere scambiati in modo degno di fiducia, supportando un controllo di accesso sicuro. Nei sistemi d'identità digitale, una struttura di fiducia funziona come un programma di certificazione. Permette ad una parte che accetta una credenziale di identità digitale (chiamata parte fidata) di

fidarsi delle politiche di identità, sicurezza e privacy della parte che emette la credenziale (chiamato fornitore di servizi di identità) e viceversa.

Più formalmente, OIX definisce un quadro di fiducia come un insieme di impegni verificabili da ciascuna delle varie parti di una transazione verso le loro controparti.

Questi impegni includono:

1. Controlli (compresi gli obblighi normativi e contrattuali) per aiutare a garantire che gli impegni siano
2. Rimedi per il mancato rispetto di tali impegni.

Un quadro di fiducia è sviluppato da una comunità i cui membri hanno obiettivi e prospettive simili. Esso definisce i diritti e le responsabilità dei partecipanti di quella comunità; specifica le politiche e gli standard specifici della comunità e definisce i processi e le procedure specifiche della comunità che forniscono garanzie. Possono esistere diversi quadri di fiducia, e i gruppi di partecipanti possono personalizzare le strutture di fiducia per soddisfare le loro particolari esigenze. La figura 4.13b mostra gli elementi coinvolti nell'OITF.

All'interno di una data organizzazione o agenzia, i seguenti ruoli sono parte del quadro generale:

- **Relying parties (RPs):** Chiamati anche fornitori di servizi, sono entità che forniscono servizi a specifici utenti. Le RP devono avere fiducia nelle identità e/o negli attributi dei loro utenti, e devono fare affidamento sulle varie credenziali presentate per dimostrare tali attributi e identità.
- **Soggetti:** Questi sono gli utenti dei servizi di una RP, compresi i clienti, gli impiegati, partner commerciali e abbonati.
- **Fornitori di attributi (AP):** Gli AP sono entità riconosciute dalla comunità di interesse come in grado di verificare determinati attributi come presentati dai soggetti e che sono attrezzati attraverso l'AXN per creare credenziali di attributo conformi secondo le regole e gli accordi dell'AXN. Alcuni AP saranno fonti di autorità per certe informazioni; più comunemente gli AP saranno broker di attributi derivati.
- **Fornitori di identità (IDP):** Queste sono entità in grado di autenticare le credenziali degli utenti e di garantire i nomi (o pseudonimi o handle) dei soggetti, e che sono attrezzati attraverso l'AXN o qualche altro sistema compatibile di Identità e (IDAM) compatibile per creare identità digitali che possono essere utilizzate per indicizzare gli attributi degli utenti.

Ci sono anche i seguenti importanti elementi di supporto come parte di un AXN:

- **Valutatori:** I valutatori valutano i fornitori di servizi di identità e gli RP e certificano che sono in grado di seguire il progetto del fornitore OITF.
- **Revisori:** Queste entità possono essere chiamate a controllare che le pratiche delle parti siano state in linea con quanto concordato per l'OITF.
- **Risolutori di controversie:** Queste entità forniscono arbitrato e risoluzione delle controversie secondo le linee guida dell'OIX.
- **Fornitori di strutture di fiducia:** Un fornitore di strutture di fiducia è un'organizzazione che traduce i requisiti dei politici in un proprio progetto per una struttura di fiducia quadro di fiducia che poi procede a costruire, facendolo in un modo che è coerente con i requisiti minimi stabiliti nella specifica OITF.

In quasi tutti i casi, ci sarà un'organizzazione candidata ragionevolmente ovvia ad assumere questo ruolo, per ogni settore industriale o grande organizzazione che decide che è appropriato interoperare con un AXN.

Le linee solide con le frecce nella Figura 4.13b indicano gli accordi con il fornitore del quadro fiduciario per l'implementazione dei requisiti tecnici, operativi e legali. Le linee tratteggiate indicano altri accordi potenzialmente interessati da questi requisiti. In termini generali, il modello illustrato nella Figura 4.13b funzionerebbe nel modo seguente. Le persone responsabili all'interno delle organizzazioni partecipanti determinano i requisiti tecnici, operativi e legali per gli scambi di informazioni sull'identità che ricadono sotto la loro autorità. Quindi selezionano i fornitori dell'OITF per implementare questi requisiti. Questi fornitori dell'OITF traducono i requisiti in un progetto per un quadro di fiducia che può includere ulteriori condizioni del fornitore dell'OITF. Il fornitore dell'OITF esamina i fornitori di servizi d'identità e gli RP e stipula con loro dei contratti per seguire i requisiti del suo quadro di fiducia quando conduce scambi di informazioni sull'identità. I contratti contengono disposizioni relative ai risolutori di controversie e ai revisori per l'interpretazione e l'applicazione del contratto.

5.7 Caso di studio: Sistema RBAC per una banca

La Dresdner Bank ha implementato un sistema RBAC che serve come utile esempio pratico.

Esempio pratico La banca usa una varietà di applicazioni informatiche. Molte di queste sono state inizialmente sviluppate per un ambiente mainframe; alcune di queste vecchie applicazioni sono ora supportate su una rete client-server, mentre altre rimangono su mainframe. Ci sono anche applicazioni più recenti su server. Prima del 1990, un semplice sistema DAC era usato su ogni server e mainframe. Gli amministratori mantenevano un file di controllo dell'accesso locale su ogni host e definivano i diritti di accesso per ogni dipendente su ogni applicazione su ogni host. Questo sistema era ingombrante, dispendioso in termini di tempo e soggetto a errori. Per migliorare il sistema, la banca ha introdotto uno schema RBAC, che è a livello di sistema e in cui la determinazione dei diritti di accesso è compartimentata in tre diverse unità amministrative per una maggiore sicurezza. I ruoli all'interno dell'organizzazione sono definiti da una combinazione di posizione ufficiale e funzione lavorativa. La tabella 4.5a fornisce degli esempi. Questo differisce un po' dal concetto di ruolo nello standard NIST, in cui un ruolo è definito da una funzione lavorativa. In una certa misura, la differenza è una questione di terminologia. In ogni caso, la strutturazione dei ruoli della banca porta a un mezzo naturale per sviluppare una gerarchia di eredità basata sulla posizione ufficiale. All'interno della banca, c'è un rigido ordine parziale delle posizioni ufficiali all'interno di ogni organizzazione, che riflette una gerarchia di responsabilità e potere. Per esempio, le posizioni di capo divisione, direttore di gruppo e impiegato sono in ordine decrescente. Quando la posizione ufficiale è combinata con la funzione lavorativa, c'è un conseguente un ordinamento dei diritti di accesso, come indicato nella tabella 4.5b. Così, l'analista finanziario/Group Manager (ruolo B) ha più diritti di accesso rispetto al ruolo analista finanziario/commesso (ruolo A). La tabella indica che il ruolo B ha altrettanti o più diritti di accesso del ruolo A in tre applicazioni e ha diritti di accesso a una quarta applicazione. D'altra parte, non c'è una relazione gerarchica tra office banking/Group Manager e l'analista finanziario/commesso perché lavorano in aree funzionali diverse. Possiamo definire una gerarchia di ruoli in cui un ruolo è superiore ad un altro se la sua posizione è superiore e le loro funzioni sono identiche. La gerarchia dei ruoli permette di risparmiare sulle definizioni dei diritti di accesso, come suggerito nella tabella 4.5c.

Table 4.5 Functions and Roles for Banking Example

(a) Functions and Official Positions		
Role	Function	Official Position
A	financial analyst	Clerk
B	financial analyst	Group Manager
C	financial analyst	Head of Division
D	financial analyst	Junior
E	financial analyst	Senior
F	financial analyst	Specialist
G	financial analyst	Assistant
...
X	share technician	Clerk
Y	support e-commerce	Junior
Z	office banking	Head of Division

(b) Permission Assignments		
Role	Application	Access Right
A	money market instruments	1, 2, 3, 4
	derivatives trading	1, 2, 3, 7 10, 12
	interest instruments	1, 4, 8, 12, 14, 16
B	money market instruments	1, 2, 3, 4, 7
	derivatives trading	1, 2, 3, 7, 10, 12, 14
	interest instruments	1, 4, 8, 12, 14, 16
	private consumer instruments	1, 2, 4, 7
...

(c) Permission Assignment with Inheritance		
Role	Application	Access Right
A	money market instruments	1, 2, 3, 4
	derivatives trading	1, 2, 3, 7 10, 12
	interest instruments	1, 4, 8, 12, 14, 16
B	money market instruments	7
	derivatives trading	14
	private consumer instruments	1, 2, 4, 7

Nello schema originale, l'assegnazione diretta dei diritti di accesso al singolo utente avveniva a livello di applicazione ed era associata alla singola applicazione. Nel nuovo schema, un'amministrazione dell'applicazione determina l'insieme dei diritti di accesso associati ad ogni singola applicazione. Tuttavia, un dato utente che esegue un dato compito potrebbe non avere tutti i diritti di accesso associati all'applicazione. applicazione. Quando un utente invoca un'applicazione, l'applicazione concede l'accesso sulla base di un profilo di sicurezza fornito a livello centrale. Un'amministrazione separata delle autorizzazioni associa i diritti di accesso ai ruoli e crea il profilo di sicurezza per un uso sulla base del ruolo dell'utente. Ad un utente viene assegnato staticamente un ruolo. In linea di principio (in questo esempio), ogni utente può essere assegnato staticamente fino a quattro ruoli e selezionare un dato ruolo da usare per invocare una particolare applicazione. Questo corrisponde al concetto NIST di sessione. In pratica, la maggior parte utenti sono assegnati staticamente a un singolo ruolo in base alla posizione dell'utente e alla sua funzione lavorativa. Tutti questi ingredienti sono rappresentati nella Figura 4.14. Il Dipartimento Risorse Umane assegna un ID utente

unico ad ogni dipendente che userà il sistema.

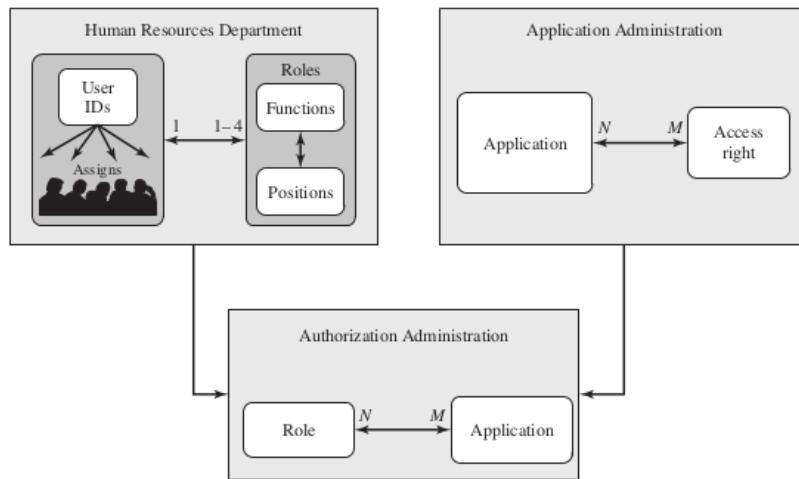


Figure 4.14 Example of Access Control Administration

In base alla posizione e alla funzione lavorativa dell’utente, il dipartimento assegna anche uno o ruoli all’utente. Le informazioni sull’utente/ruolo sono fornite all’Amministrazione delle autorizzazioni Amministrazione, che crea un profilo di sicurezza per ogni utente che associa l’ ID utente e il ruolo a un insieme di diritti di accesso. Quando un utente invoca un’applicazione, l’applicazione consulta il profilo di sicurezza per quell’utente per determinare quale sottoinsieme dei diritti di accesso dell’applicazione sono in vigore per questo utente in questo ruolo. Un ruolo può essere usato per accedere a diverse applicazioni. Così, l’insieme dei diritti di accesso associati a un ruolo possono includere diritti di accesso che non sono associati a una delle applicazioni che l’utente invoca. Questo è illustrato nella tabella 4.5b. Il ruolo A ha numerosi diritti di accesso, ma solo un sottoinsieme di questi diritti è applicabile a ciascuna delle tre applicazioni che il ruolo A può invocare.

Alcune cifre su questo sistema sono interessanti. All’interno della banca, ci sono 65 posizioni ufficiali, che vanno da un impiegato in una filiale, attraverso il direttore di filiale, a un membro del consiglio di amministrazione. Queste posizioni sono combinate con 368 diverse funzioni lavorative fornite dal database delle risorse umane. Potenzialmente, ci sono 23.920 ruoli diversi, ma il numero di ruoli attualmente in uso è di circa 1.300. Questo è in linea con l’esperienza di altre implementazioni RBAC. In media, 42.000 profili di sicurezza sono distribuiti alle applicazioni ogni giorno dal modulo di amministrazione delle autorizzazioni.

Capitolo 6

Capitolo 5

6.1 La necessità di sicurezza del database

I database delle organizzazioni tendono a concentrare le informazioni sensibili in un unico sistema logico. Gli esempi includono:

- Dati finanziari aziendali
- Registrazioni telefoniche riservate
- Informazioni sui clienti e sui dipendenti, come il nome, il numero di previdenza sociale, le informazioni sul conto bancario e le informazioni sulla carta di credito
- Informazioni proprietarie sui prodotti
- Informazioni sanitarie e cartelle cliniche

Per molte aziende e altre organizzazioni è importante poter fornire a clienti, partner e dipendenti l'accesso a queste informazioni. Tuttavia, tali informazioni possono essere oggetto di minacce interne ed esterne di uso improprio o di modifiche non autorizzate. Di conseguenza, la sicurezza specifica per i database è una componente sempre più importante di una strategia di sicurezza organizzativa complessiva.

Le seguenti sono le ragioni per le quali la sicurezza dei database non ha tenuto il passo con la crescente dipendenza dai database:

1. Esiste un drammatico squilibrio tra la complessità dei moderni sistemi di gestione dei database (DBMS) e le tecniche di sicurezza utilizzate per proteggere questi sistemi critici. Un DBMS è un software molto complesso e di grandi dimensioni, che offre molte opzioni, tutte da comprendere bene e da proteggere per evitare violazioni dei dati. Sebbene le tecniche di sicurezza siano progredite, la crescente complessità dei DBMS con molte nuove funzionalità e servizi, ha portato a una serie di nuove vulnerabilità e al potenziale uso improprio.

2. I database dispongono di un sofisticato protocollo di interazione chiamato Structured Query Language (SQL), che è di gran lunga SQL (Structured Query Language), che è molto più complesso, ad esempio, del protocollo HTTP (Hypertext Transfer Protocol) utilizzato per interagire con un servizio Web. Una sicurezza efficace dei database richiede una strategia basata sulla piena comprensione delle vulnerabilità di sicurezza dell'SQL.
3. L'organizzazione tipica non dispone di personale a tempo pieno per la sicurezza dei database. Il risultato è uno squilibrio tra requisiti e capacità. La maggior parte delle organizzazioni dispone di uno staff di amministratori di database, il cui compito è quello di gestire il database per garantire disponibilità, prestazioni, correttezza e facilità d'uso. Questi amministratori possono avere una conoscenza limitata della sicurezza e poco tempo a disposizione per padroneggiare e applicare le tecniche di sicurezza. D'altra parte, i responsabili della sicurezza all'interno di un'organizzazione possono avere una conoscenza molto limitata della tecnologia dei database e dei DBMS.
4. La maggior parte degli ambienti aziendali è costituita da un mix eterogeneo di piattaforme di database (Oracle, IBM DB2).

6.2 Sistemi di gestione dei database

Un database è una raccolta strutturata di dati memorizzati per essere utilizzati da una o più applicazioni. Oltre ai dati, un database contiene le relazioni tra i dati e i gruppi di dati. Come esempio della distinzione tra file di dati e database, si consideri quanto segue:

Un semplice file del personale potrebbe consistere in una serie di record, uno per ogni dipendente. Ogni record riporta il nome, l'indirizzo, la data di nascita, la posizione, lo stipendio e altri dettagli necessari all'ufficio del personale.

Un database del personale comprende un file del personale, come appena descritto. Può anche includere un file delle presenze, che mostra per ogni settimana le ore lavorate da ciascun dipendente. Con un'organizzazione a database, questi due file sono Questi due file sono collegati tra loro, in modo che un programma per le paghe possa estrarre le informazioni sulle ore lavorate e sulla retribuzione di ciascun dipendente e lo stipendio di ciascun dipendente per generare le buste paga. Il database è accompagnato da un sistema di gestione di database (DBMS), che è una suite di programmi per la costruzione di un sistema di gestione di database.

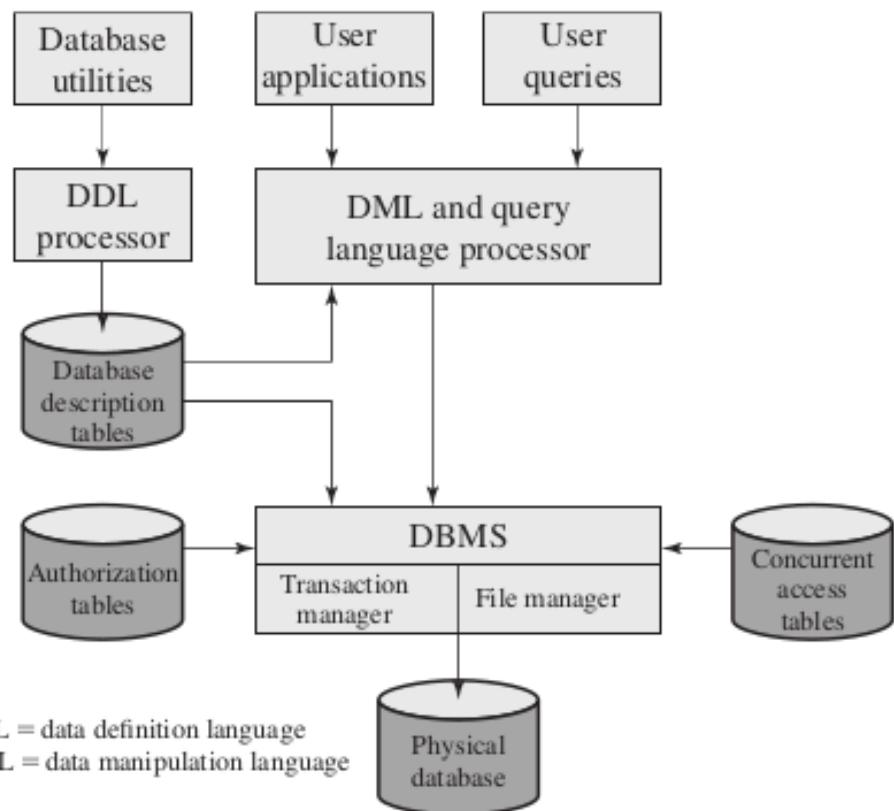


Figure 5.1 DBMS Architecture

Un linguaggio di interrogazione fornisce un'interfaccia uniforme al database per utenti e applicazioni. La Figura 5.1 mostra un diagramma a blocchi semplificato dell'architettura di un DBMS. I progettisti e gli amministratori di database utilizzano un linguaggio di definizione dei dati (DDL) per definire la struttura logica del database e le proprietà procedurali, che sono rappresentate da una serie di descrizioni del database. Un linguaggio di manipolazione dei dati (DML) fornisce un potente insieme di strumenti per gli sviluppatori di applicazioni. I linguaggi di interrogazione sono linguaggi dichiarativi progettati per supportare gli utenti finali. Il sistema di gestione del database utilizza le tabelle di descrizione del database per gestire il database fisico. L'interfaccia al database avviene attraverso un modulo di gestione dei file e un modulo di gestione delle transazioni, e un modulo di gestione delle transazioni. Oltre alla tabella di descrizione del database, altre due tabelle supportano il DBMS.

Il DBMS utilizza tabelle di autorizzazione per garantire che l'utente abbia il permesso di eseguire la query. La tabella di accesso concorrente previene i conflitti quando vengono eseguiti comandi simultanei in conflitto. I sistemi di database forniscono un accesso efficiente a grandi volumi di dati e sono vitali per il funzionamento di molte organizzazioni.

6.3 Sql Injection Attacks

L'attacco SQL injection (SQLi) è una delle minacce alla sicurezza della rete più diffuse e pericolose delle minacce alla sicurezza della rete.

Considerate i seguenti rapporti:

1. Il rapporto sugli attacchi alle applicazioni Web di Imperva del luglio 2013 ha esaminato una sezione di server di applicazioni Web nel settore e ha monitorato otto diversi tipi di attacchi comuni. Il rapporto ha rilevato che gli attacchi SQLi si sono classificati al primo o al secondo posto per numero totale di attacchi, numero di richieste di attacco per ogni attacco e numero medio di giorni al mese in cui un'applicazione ha subito almeno un attacco. Imperva ha osservato un singolo sito web che ha ricevuto 94.057 richieste di attacco SQL injection in un solo giorno.
2. Il rapporto 2013 dell'Open Web Application Security Project [OWAS13] sui 10 rischi più critici per la sicurezza delle applicazioni Web elenca gli attacchi a iniezione, in particolare gli attacchi SQLi, come il rischio principale.
3. Il rapporto Veracode 2016 State of Software Security ha rilevato che la percentuale di applicazioni colpite da attacchi SQLi si aggira intorno al 35%.
4. Il Trustwave 2016 Global Security Report [TRUS16] elenca gli attacchi SQLi come una delle due principali tecniche di intrusione. Il rapporto rileva che SQLi può rappresentare una minaccia significativa per i dati sensibili, come le informazioni di identificazione personale (PII) e i dati delle carte di credito, e può essere difficile prevenire e relativamente facile sfruttare questi attacchi.

In termini generali, un attacco SQLi è progettato per sfruttare la natura delle pagine delle applicazioni Web. pagine web. A differenza delle pagine web statiche degli anni passati, la maggior parte dei siti web attuali hanno componenti e contenuti dinamici. Molte di queste pagine richiedono informazioni, come informazioni, come la posizione, le informazioni sull'identità personale e i dati della carta di credito.

Questo contenuto dinamico viene solitamente trasferito da e verso database back-end che contengono volumi di informazioni, dai dati dei titolari di carta di credito al tipo di scarpe da corsa più acquistato. La pagina web di un server applicativo esegue query SQL ai database per inviare e ricevere informazioni fondamentali per rendere positiva l'esperienza dell'utente. In un ambiente di questo tipo, un attacco SQLi è progettato per inviare comandi SQL dannosi al server di database. L'obiettivo più comune dell'attacco è l'estrazione in blocco dei dati. Gli aggressori possono scaricare tabelle di database con centinaia di migliaia di record di clienti. A seconda dell'ambiente, l'iniezione SQL può essere sfruttata anche per modificare o eliminare dati, eseguire comandi arbitrari del sistema operativo o lanciare attacchi denial-of-service (DoS).

6.3.1 Un tipico attacco SQLi

SQLi è un attacco che sfrutta una vulnerabilità di sicurezza che si verifica nel livello di database di un'applicazione (come le query). Utilizzando l'iniezione SQL, l'aggressore può estrarre o manipolare i dati dell'applicazione web. L'attacco è attuabile quando l'input dell'utente viene filtrato in modo errato per i caratteri di escape letterali di stringa incorporati nelle istruzioni SQL oppure l'input dell'utente non è fortemente tipizzato e quindi viene eseguito inaspettatamente.

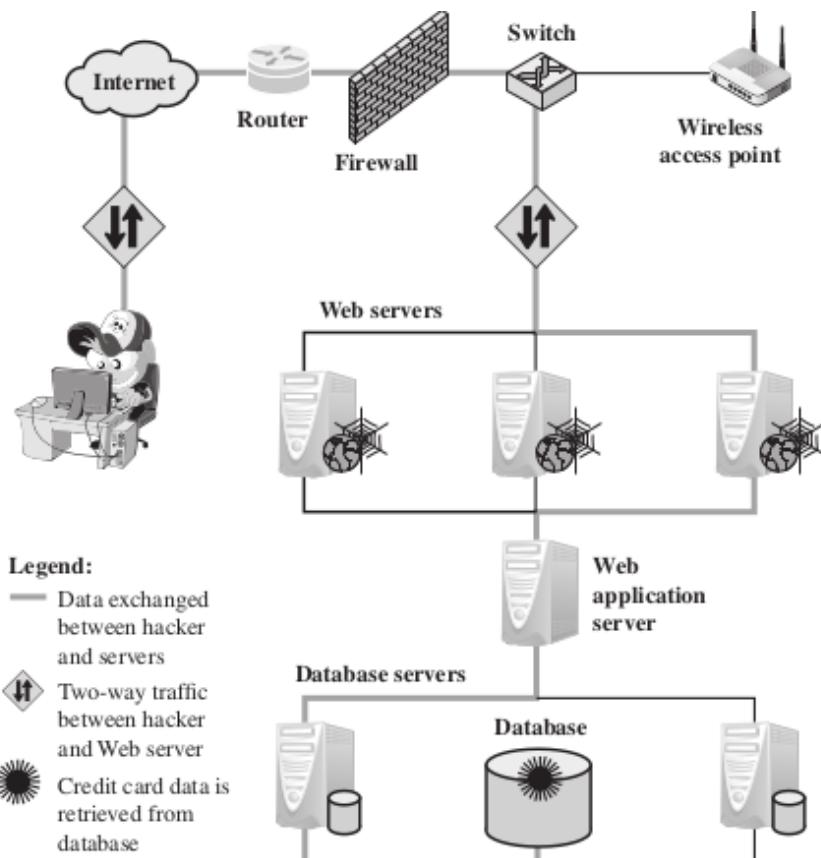


Figure 5.5 Typical SQL Injection Attack

La Figura 5.5, tratta da, è un tipico esempio di attacco SQLi. I passi sono i seguenti:

1. L'hacker trova una vulnerabilità in un'applicazione Web personalizzata e inietta un comando SQL in un database inviando il comando a un'applicazione Web. Il comando viene iniettato nel traffico che verrà accettato dal firewall.
2. Il server Web riceve il codice dannoso e lo invia al server dell'applicazione Web.
3. Il server delle applicazioni Web riceve il codice dannoso dal server Web e lo invia al server del database.

4. Il server del database esegue il codice dannoso sul database.
Il database restituisce i dati della tabella delle carte di credito.
5. Il server dell'applicazione Web genera dinamicamente una pagina con i dati della carta di credito dal database.
6. Il server Web invia i dati della carta di credito all'hacker.

6.3.2 Vie e tipi di attacco SQLi

Possiamo caratterizzare gli attacchi SQLi in termini di vie di attacco e di tipo di attacco.

Le principali vie di attacco sono le seguenti:

- **Input dell'utente**

In questo caso, gli aggressori iniettano comandi SQL fornendo input dell'utente opportunamente elaborati. Un'applicazione Web può leggere l'input dell'utente in diversi modi, a seconda dell'ambiente in cui viene distribuita. Nella maggior parte degli attacchi SQLi che hanno come obiettivo le applicazioni Web, l'input dell'utente proviene in genere da moduli inviati all'applicazione Web tramite richieste HTTP GET o POST. Le applicazioni Web sono generalmente in grado di accedere all'input dell'utente contenuto in queste richieste come accederebbero a qualsiasi altra variabile nell'ambiente.

- **Variabili del server**

Le variabili del server sono un insieme di variabili che contengono intestazioni HTTP, intestazioni del protocollo di rete e variabili ambientali. Le applicazioni Web utilizzano queste variabili del server in vari modi, come la registrazione di statistiche di utilizzo e identificare le tendenze di navigazione. Se queste variabili vengono registrate in un database senza sanitizzazione, si potrebbe creare una vulnerabilità SQL injection.

Poiché gli aggressori possono falsificare i valori inseriti nelle intestazioni HTTP e di rete, possono sfruttare questa vulnerabilità inserendo i dati direttamente nelle intestazioni. Quando la query per registrare la variabile del server viene inviata al database, l'attacco nell'intestazione falsificata viene attivato.

- **Iniezione di secondo ordine**

L'iniezione di secondo ordine si verifica quando i meccanismi di prevenzione contro gli attacchi SQL injection sono incompleti. Nell'iniezione di secondo ordine, un utente malintenzionato potrebbe basarsi su dati già presenti nel sistema o nel database per scatenare un attacco di tipo SQL injection, per cui quando si verifica l'attacco, l'input che modifica la query per causare un attacco non proviene dall'utente, ma dal sistema stesso.

- **Cookie**

Quando un client torna a un'applicazione Web, i cookie possono essere utilizzati per ripristinare le informazioni sullo stato del client. Poiché il client ha il controllo sui cookie, un aggressore potrebbe alterare i cookie in modo tale che, quando il server applicativo crea una query SQL basata sul contenuto del cookie, la struttura e la funzione della query vengano modificate.

- **Input fisico dell'utente**

L'iniezione SQL è possibile fornendo input all'utente che costruisce un attacco al di fuori dell'ambito delle richieste Web. Questo input dell'utente può assumere sotto

forma di codici a barre convenzionali, tag RFID o persino moduli cartacei che vengono scansionati con il riconoscimento ottico dei caratteri e trasmessi a un sistema di gestione di database.

I tipi di attacco possono essere raggruppati in tre categorie principali: inband, inferential e out-of-band. Un attacco inband utilizza lo stesso canale di comunicazione per iniettare codice SQL e recuperare i risultati. I dati recuperati vengono presentati direttamente nella pagina web dell'applicazione. I tipi di attacco inband includono i seguenti:

- **Tautologia:** Questa forma di attacco inietta codice in una o più dichiarazioni condizionali in modo che siano sempre valutate come vere.
- **Commento di fine riga:** Dopo aver iniettato del codice in un particolare campo, il codice legittimo che segue codice legittimo che segue viene annullato attraverso l'uso di commenti di fine riga. Un esempio esempio, aggiungere " - " dopo gli input, in modo che le query rimanenti non vengano trattate come codice non vengono trattate come codice eseguibile, ma come commenti. L'esempio di tautologia precedente è anch'esso di questa forma.
- **Query di tipo "piggybacked":** L'aggressore aggiunge ulteriori query oltre a quella prevista, aggiungendo l'attacco a una richiesta legittima. Questa tecnica si basa su configurazioni del server che consentono diverse query all'interno di un'unica stringa di codice. L'esempio riportato nella sezione precedente è di questo tipo.

Con un attacco inferenziale, non c'è un vero e proprio trasferimento di dati, ma l'aggressore è in grado di ricostruire le informazioni inviando particolari richieste e osservando il comportamento del sito web. il comportamento risultante del server del sito web/database.

I tipi di attacco inferenziale includono i seguenti:

- **Query illegali/logicamente errate**

Questo attacco consente a un aggressore di raccogliere informazioni importanti sul tipo e sulla struttura del database di backend di un'applicazione Web. L'attacco è considerato una fase preliminare di raccolta di informazioni per altri attacchi. La vulnerabilità sfruttata da questo attacco è che la pagina di errore predefinita restituita dai server applicativi è spesso eccessivamente descrittiva. Infatti, il semplice fatto che venga generato un messaggio di errore può spesso rivelare a un aggressore parametri vulnerabili/iniettabili.

- **Iniezione SQL cieca**

L'iniezione SQL cieca consente agli aggressori di dedurre i dati presenti in un sistema di database anche quando il sistema è sufficientemente sicuro da non mostrare alcuna informazione errata all'aggressore. L'attaccante pone al server domande di tipo vero/falso. Se l'affermazione iniettata è vera, il sito continua a funzionare normalmente. Se l'affermazione risulta falsa, anche se non viene visualizzato alcun messaggio di errore descrittivo, la pagina differisce in modo significativo da quella normalmente funzionante.

In un attacco **out-of-band**, i dati vengono recuperati utilizzando un canale diverso (ad esempio, un'e-mail con i risultati della query). un'e-mail con i risultati della query e inviata al tester). Questo può essere utilizzato quando ci sono limitazioni nel recupero delle informazioni, ma la connettività in uscita dal server del database è debole.

6.3.3 Contromisure per attacchi SQLi

Poiché gli attacchi SQLi sono così diffusi, dannosi e variegati sia per modalità di attacco che per tipologia, una singola contromisura è insufficiente. È piuttosto necessario un insieme integrato di tecniche. Molti attacchi SQLi hanno successo perché gli sviluppatori hanno utilizzato pratiche di codifica poco sicure. Pertanto, la codifica difensiva è un modo efficace per ridurre drasticamente la minaccia di SQLi.

Esempi di codifica difensiva sono i seguenti:

- **Pratiche di codifica difensiva manuale:** Una vulnerabilità comune sfruttata dagli attacchi SQLi è l'insufficiente convalida dell'input. La soluzione più semplice per eliminare queste per eliminare queste vulnerabilità è l'applicazione di pratiche di codifica difensiva adeguate.

Un esempio è il controllo del tipo di input, per verificare che gli input che devono essere numerici non contengano caratteri diversi dalle cifre. Questo tipo di tecnica può attaccare basati su errori di forzatura nel sistema di gestione del database.

Un altro tipo di pratica di codifica è quella che esegue la corrispondenza dei modelli per cercare di distinguere un input normale da uno anormale.

- **Inserimento di query parametrizzate:** Questo approccio cerca di prevenire l'SQLi consentendo allo sviluppatore dell'applicazione di specificare in modo più accurato la struttura di una query di una query SQL e di passarle i parametri di valore separatamente, in modo tale che qualsiasi modo che l'utente non possa modificare la struttura della query.
- **SQL DOM:** SQL DOM è un insieme di classi che consente la valutazione automatica dei tipi di dati e l'escape.

Questo approccio utilizza l'incapsulamento delle database per fornire un modo sicuro e affidabile di accedere ai database. Questo cambia il processo di creazione delle query da un processo sregolato che utilizza la concatenazione di stringhe a uno sistematico che utilizza un tipo di processo sistematico che utilizza un'API con controllo di tipo. All'interno dell'API, gli sviluppatori di codice, come il filtraggio dell'input e il controllo rigoroso del tipo di input dell'utente.

Sono stati sviluppati diversi metodi di rilevamento, tra cui i seguenti:

- **Basato sulla firma:** Questa tecnica tenta di corrispondere a specifici schemi di attacco.

Questo approccio deve essere costantemente aggiornato e potrebbe non funzionare contro gli attacchi auto-modificanti.

Basato sulle anomalie: Questo approccio cerca di definire il comportamento normale e poi rilevare i modelli di comportamento al di fuori dell'intervallo normale. Un certo numero di approcci

6.4 Controllo dell'accesso al database

I DBMS commerciali e open-source forniscono in genere una capacità di controllo degli accessi per il database. Il DBMS opera sulla base del presupposto che il sistema informatico che abbia autenticato ogni utente. Come ulteriore linea di difesa, il sistema informatico può utilizzare il sistema generale di controllo degli accessi per determinare se un utente può accedere al database nel suo complesso.

Per gli utenti che sono stati autenticati e a cui è stato concesso l'accesso al database, un sistema di controllo dell'accesso al database fornisce una funzionalità specifica che controlla l'accesso a porzioni del database.

I DBMS commerciali e open-source forniscono un controllo di accesso discrezionale o basato sui ruoli. In genere, un DBMS può supportare una serie di politiche amministrative, tra cui le seguenti:

- **Amministrazione centralizzata:** Un piccolo numero di utenti privilegiati può concedere e revocare i diritti di accesso.
- **Amministrazione basata sulla proprietà:** Il proprietario (creatore) di una tabella può concedere e revocare i diritti di accesso alla tabella.
- **Amministrazione decentralizzata:** Oltre a concedere e revocare i diritti di accesso a una tabella, il proprietario della tabella può concedere e revocare i diritti di autorizzazione ad altri utenti, consentendo loro di concedere e revocare i diritti di accesso alla tabella.

Come ogni sistema di controllo degli accessi, un sistema di controllo degli accessi ai database distingue diversi diritti di accesso, tra cui creazione, inserimento, cancellazione, aggiornamento, lettura e scrittura. I diritti di accesso possono riguardare l'intero database, singole tabelle o righe o colonne selezionate all'interno di una tabella. o colonne all'interno di una tabella.

6.4.1 Autorizzazioni a cascata

L'opzione di concessione consente di far passare un diritto di accesso a cascata attraverso un certo numero di utenti. Consideriamo un diritto di accesso specifico e illustriamo il

fenomeno della cascata nella Figura 5.6. La figura indica che Ann concede il diritto di accesso a Bob al tempo $t = 10$ e a Chris al tempo $t = 20$. Supponiamo che l'opzione di concessione sia sempre utilizzata. Pertanto, Bob è in grado di concedere il diritto di accesso a David al tempo $t = 30$. Chris concede in modo ridondante il diritto di accesso a David al tempo $t = 50$. Nel frattempo, David concede il diritto a Ellen, che a sua volta lo concede a Jim e successivamente David concede il diritto a Frank. Così come la concessione dei privilegi avviene a cascata da un utente all'altro utilizzando l'opzione grant, anche la revoca dei privilegi avviene a cascata.

Così, se Ann revoca il diritto di accesso a Bob e Chris, il diritto di accesso viene revocato anche a David, Ellen, Jim e Frank. Una complicazione sorge quando un utente riceve lo stesso diritto di accesso più volte, come accade nel caso di David. Supponiamo che Bob revochi il privilegio a David. David ha ancora il diritto di accesso perché gli è stato concesso da Chris a $t = 50$. Tuttavia, David ha concesso il diritto di accesso a un altro utente. Tuttavia, David ha concesso il diritto di accesso a Ellen dopo aver ricevuto il diritto, con opzione di concessione, da Bob ma prima di riceverlo da Chris. La maggior parte delle implementazioni prevede che in questa circostanza il diritto di accesso a Ellen e quindi a Jim venga revocato quando Bob revoca il diritto di accesso a David. Questo perché a $t = 40$, quando David ha concesso il diritto di accesso a Ellen, David aveva solo l'opzione di concessione da parte di Bob. La revoca del diritto da parte di Bob provoca la revoca di tutte le successive concessioni a cascata riconducibili esclusivamente a Bob tramite David. Poiché David ha concesso il diritto di accesso a Frank dopo che a David era stato concesso il diritto di accesso con opzione di concessione da Chris, il diritto di accesso a Frank rimane. Questi effetti sono mostrati nella parte inferiore della Figura 5.6.

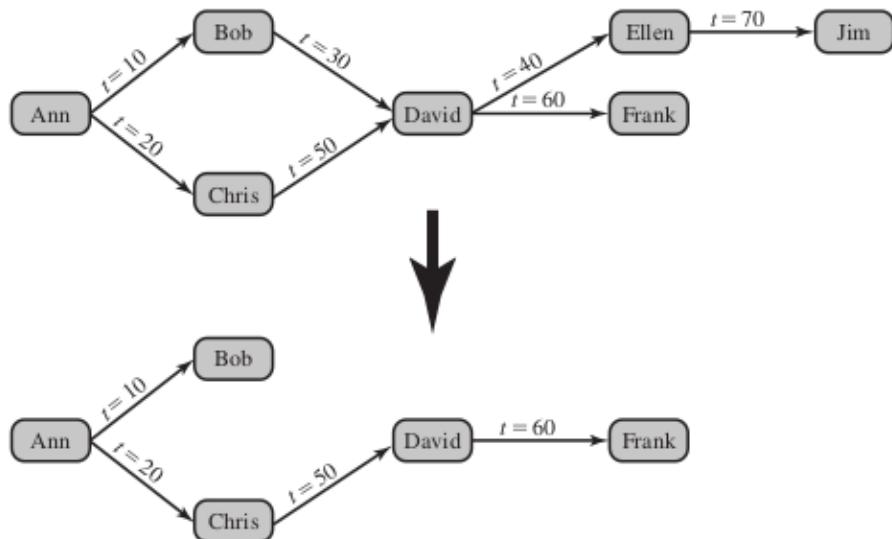


Figure 5.6 **Bob Revokes Privilege from David**

Per generalizzare la convenzione seguita dalla maggior parte delle implementazioni è la seguente. Quando l'utente A revoca un diritto d'accesso, viene revocato anche qualsiasi diritto d'accesso a cascata, a meno che a meno che quel diritto di accesso non esisterebbe anche se la concessione originale da parte di A non fosse mai avvenuta.

6.4.2 Controllo dell'accesso basato sui ruoli

Uno schema di controllo degli accessi basato sui ruoli (RBAC) si adatta naturalmente al controllo degli accessi ai database. A differenza di un file system associato a una o poche applicazioni, un sistema di database supporta spesso decine di applicazioni. In un ambiente di questo tipo, un singolo utente può utilizzare una serie di applicazioni per eseguire una varietà di compiti, ognuno dei quali richiede un proprio insieme di privilegi. Sarebbe una pratica amministrativa scorretta concedere semplicemente di concedere agli utenti tutti i diritti di accesso di cui hanno bisogno per tutte le attività che svolgono. RBAC fornisce un mezzo per alleggerire il carico amministrativo e migliorare la sicurezza.

In un ambiente di controllo degli accessi discrezionale, si possono classificare gli utenti del database in tre grandi categorie in tre grandi categorie:

1. **Proprietario dell'applicazione:** un utente finale che possiede oggetti di database (tabelle, colonne e righe) come parte di un'applicazione, e righe) come parte di un'applicazione. Cioè, gli oggetti del database sono generati dall'applicazione o sono preparati per essere utilizzati dall'applicazione.
2. **Utente finale diverso dal proprietario dell'applicazione:** un utente finale che opera sugli oggetti del database attraverso una particolare applicazione, ma non base tramite una particolare applicazione, ma non possiede alcun oggetto del database.
3. **Amministratore:** Utente che ha la responsabilità amministrativa di una parte o di tutto il database.

È possibile fare alcune affermazioni generali su RBAC riguardo a questi tre tipi di utenti.

Un'applicazione è associata a una serie di compiti, ognuno dei quali richiede diritti di accesso specifici a parti del database. ogni compito richiede diritti di accesso specifici a porzioni del database. Per ogni attività è possibile definire uno Per ogni attività è possibile definire uno o più ruoli che specificano i diritti di accesso necessari.

Il proprietario dell'applicazione dell'applicazione può assegnare ruoli agli utenti finali.

Gli amministratori sono responsabili dei ruoli più sensibili o generali, compresi quelli che hanno a che fare con la gestione dei componenti fisici e logici del database, come i file di dati, gli utenti e i meccanismi di sicurezza. Il sistema deve essere impostato in modo da dare a certi amministratori determinati privilegi. Gli amministratori, a loro volta, possono assegnare agli utenti ruoli di tipo amministrativo.

Una struttura RBAC per database deve fornire le seguenti funzionalità:

- Creare ed eliminare ruoli.
- Definire le autorizzazioni per un ruolo.
- Assegnare e annullare l'assegnazione degli utenti ai ruoli.

Un buon esempio dell'uso dei ruoli nella sicurezza dei database è la struttura RBAC di Microsoft SQL Server.

SQL Server supporta tre tipi di ruoli:

1. Ruoli del server
2. Ruoli del database
3. Ruoli definiti dall'utente.

I primi due tipi di ruoli sono definiti ruoli fissi (vedere Tabella 5.2); sono preconfigurati per un sistema con diritti di accesso specifici. L'amministratore o l'utente non possono aggiungere, eliminare o modificare i ruoli fissi; è possibile solo aggiungere e rimuovere utenti come è possibile solo aggiungere e rimuovere utenti come membri di un ruolo fisso. I ruoli fissi del server sono definiti a livello di server ed esistono indipendentemente da qualsiasi database di utenti. database degli utenti. Sono progettati per facilitare il compito amministrativo.

Table 5.2 Fixed Roles in Microsoft SQL Server

Role	Permissions
Fixed Server Roles	
sysadmin	Can perform any activity in SQL Server and have complete control over all database functions
serveradmin	Can set server-wide configuration options and shut down the server
setupadmin	Can manage linked servers and startup procedures
securityadmin	Can manage logins and CREATE DATABASE permissions, also read error logs and change passwords
processadmin	Can manage processes running in SQL Server
Dbcreator	Can create, alter, and drop databases
diskadmin	Can manage disk files
bulkadmin	Can execute BULK INSERT statements
Fixed Database Roles	
db_owner	Has all permissions in the database
db_accessadmin	Can add or remove user IDs
db_datareader	Can select all data from any user table in the database
db_datawriter	Can modify any data in any user table in the database
db_ddladmin	Can issue all data definition language statements
db_securityadmin	Can manage all permissions, object ownerships, roles and role memberships
db_backupoperator	Can issue DBCC, CHECKPOINT, and BACKUP statements
db_denydatareader	Can deny permission to select data in the database
db_denydatawriter	Can deny permission to change data in the database

Questi ruoli hanno autorizzazioni diverse e hanno lo scopo di fornire la possibilità di distribuire le responsabilità amministrative senza dover cedere il controllo completo. Gli amministratori di database possono usare questi ruoli fissi per assegnare diversi compiti amministrativi al personale e dare loro solo i diritti assolutamente necessari. I ruoli fissi di database operano a livello di singolo database. Come nel caso dei ruoli di server fissi, alcuni ruoli di database fissi, come db_accessadmin e db_securityadmin, sono progettati per aiutare il DBA a delegare le responsabilità amministrative. Altri, come db_datareader e db_datawriter, sono stati progettati per fornire di autorizzazioni generali per un utente finale.

Esistono due tipi di ruoli definiti dall'utente: Standard e Applicazione.

Per un ruolo standard un utente autorizzato può assegnare altri utenti al ruolo.

Un ruolo applicativo è associato a un'applicazione piuttosto che a un gruppo di utenti e richiede una password. Il ruolo viene attivato quando un'applicazione esegue il codice appropriato. Un utente che ha accesso all'applicazione può usare il ruolo di applicazione per accedere al database. Spesso le applicazioni di database applicano la propria sicurezza in base alla logica dell'applicazione.

Ad esempio, è possibile utilizzare un ruolo dell'applicazione con la propria password per consentire a un determinato utente di ottenere e modificare i dati solo in determinate ore, solo in determinati orari. In questo modo, è possibile realizzare una gestione della sicurezza più complessa all'interno della logica dell'applicazione.

6.5 Interferenze

L'inferenza, in relazione alla sicurezza dei database, è il processo di esecuzione di interrogazioni autorizzate e di deduzione di informazioni non autorizzate dalle risposte legittime ricevute.

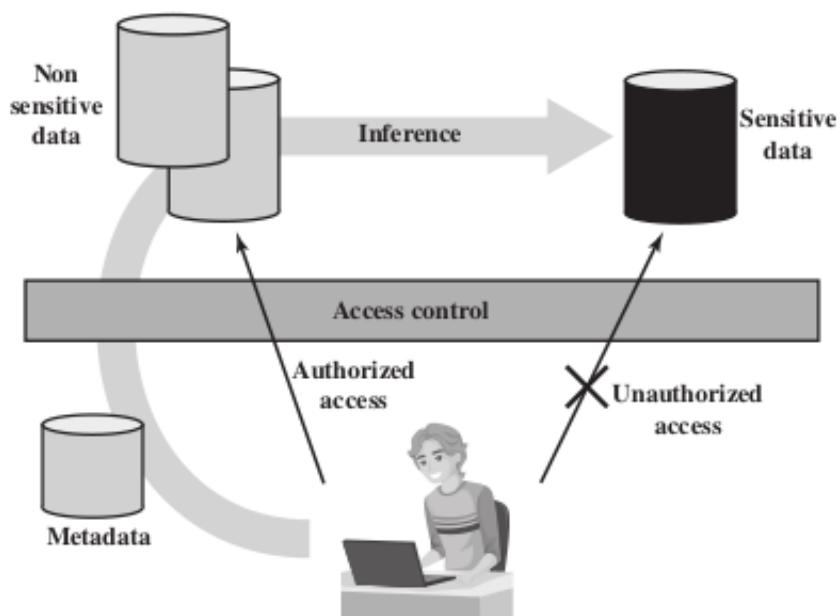


Figure 5.7 Indirect Information Access via Inference Channel

Il problema dell'inferenza si presenta quando la combinazione di un certo numero di dati è più sensibile dei singoli elementi, oppure quando una combinazione di dati può essere utilizzata per dedurre dati di maggiore sensibilità. La Figura 5.7 illustra il processo. L'attaccante può utilizzare sia i dati non sensibili sia i metadati.

I metadati si riferiscono alla conoscenza delle correlazioni o delle dipendenze tra i dati che possono essere utilizzate per dedurre informazioni non altrimenti disponibili a un particolare utente. Il percorso di trasferimento delle informazioni attraverso il quale si ottengono dati non autorizzati viene definito canale di inferenza. In termini generali, si possono utilizzare due tecniche di inferenza per ricavare ulteriori informazioni aggiuntive:

L'analisi delle dipendenze funzionali tra gli attributi all'interno di una tabella o tra le tabelle e l'unione di viste con gli stessi vincoli.

Un esempio di quest'ultima tecnica, mostrato nella Figura 5.8, illustra il problema dell'inferenza. La Figura 5.8a mostra una tabella Inventario con quattro colonne. La Figura 5.8b mostra due viste, definite in SQL come segue:

Item	Availability	Cost (\$)	Department
Shelf support	in-store/online	7.99	hardware
Lid support	online only	5.49	hardware
Decorative chain	in-store/online	104.99	hardware
Cake pan	online only	12.99	housewares
Shower/tub cleaner	in-store/online	11.99	housewares
Rolling pin	in-store/online	10.99	housewares

(a) Inventory table

Availability	Cost (\$)	Item	Department
in-store/online	7.99	Shelf support	hardware
online only	5.49	Lid support	hardware
in-store/online	104.99	Decorative chain	hardware

(b) Two views

Item	Availability	Cost (\$)	Department
Shelf support	in-store/online	7.99	hardware
Lid support	online only	5.49	hardware
Decorative chain	in-store/online	104.99	hardware

(c) Table derived from combining query answers

Figure 5.8 Inference Example

Gli utenti di queste viste non sono autorizzati ad accedere alla relazione tra Voce e Costo.

Un utente che ha accesso a una o a entrambe le viste non può dedurre la relazione attraverso le dipendenze funzionali. In altre parole, non esiste una relazione funzionale tra Articolo e Costo tale che la conoscenza dell'Articolo e forse di altre informazioni sia

sufficiente per dedurre il Costo. Tuttavia, supponiamo che le due viste siano state create con il vincolo di accesso che Voce e Costo non possono essere consultati insieme. Un utente

che conosce la struttura della tabella Inventario e che sa che le tabelle della vista mantengono lo stesso ordine delle righe della tabella Inventario è in grado di unire le due

viste per costruire la tabella mostrata nella Figura 5.8c. In questo modo si viola la politica di controllo degli accessi, secondo cui la relazione tra gli attributi Item e Cost non deve essere divulgata.

In termini generali, esistono due approcci per affrontare la minaccia della divulgazione per inferenza:

- Rilevamento dell'inferenza durante la progettazione del database: Questo approccio rimuove un canale di inferenza modificando la struttura del database o cambiando il regime di controllo degli accessi per evitare l'inferenza. Gli esempi includono la rimozione delle dipendenze di dati suddividendo una tabella in più tabelle o utilizzando ruoli di controllo dell'accesso a grana più fine in uno schema RBAC.
- Rilevamento delle inferenze al momento dell'interrogazione: Questo approccio cerca di eliminare una violazione del canale di inferenza

6.6 Crittografia del database

Il database è in genere la risorsa informativa più preziosa per qualsiasi organizzazione ed è quindi protetto da più livelli di sicurezza, tra cui firewall, meccanismi di autenticazione, sistemi di controllo dell'accesso generale e sistemi di controllo dell'accesso al database. Inoltre, per i dati particolarmente sensibili, la crittografia del database è giustificata e spesso implementata. La crittografia diventa l'ultima linea di difesa nella sicurezza dei database.

La crittografia dei database presenta due svantaggi:

1. Gestione delle chiavi

Gli utenti autorizzati devono avere accesso alla chiave di decifrazione per i dati a cui hanno accesso. per i dati ai quali hanno accesso. Poiché un database è tipicamente accessibile a un'ampia gamma di utenti e di applicazioni, è necessario fornire chiavi sicure a parti selezionate del database.

2. Inflessibilità

Quando una parte o la totalità del database è crittografata, diventa più difficile eseguire la ricerca dei record. La crittografia può essere applicata all'intero database, a livello di record (crittografia di record record selezionati), a livello di attributi (crittografia di colonne selezionate) o a livello di singoli campi.

Sono stati adottati diversi approcci alla crittografia dei database. In questa sezione, esaminiamo un approccio rappresentativo per un database multiutente. Un DBMS è un complesso insieme di hardware e software. Richiede una grande capacità di memorizzazione e richiede personale qualificato per la manutenzione, la protezione dai disastri, l'aggiornamento e la sicurezza.

Per molte organizzazioni di piccole e medie dimensioni, una soluzione interessante è quella di esternalizzare il DBMS e il database a un fornitore di servizi. Il fornitore di servizi gestisce il database fuori sede e può garantire un'elevata disponibilità, la prevenzione dei disastri e un accesso e un aggiornamento efficienti. Il problema principale di questa soluzione è la riservatezza dei dati.

Una soluzione semplice al problema della sicurezza in questo contesto è quella di crittografare l'intero database e non fornire la crittografia dell'intero database e non fornire le chiavi di crittografia/decrittografia al fornitore di servizi. Questa soluzione è di per sé poco flessibile. L'utente non ha la possibilità di accedere a singoli dati in base a ricerche o indicizzazioni su parametri chiave, ma deve scaricare intere tabelle dal database, decifrarle e lavorare con i risultati.

Per garantire una maggiore flessibilità, deve essere possibile lavorare con il database nella sua forma criptata. Un esempio di questo tipo di approccio, illustrato nella Figura 5.9. Un approccio simile è descritto in [HACI02]. Le entità coinvolte sono quattro coinvolte:

1. **Proprietario dei dati:** un'organizzazione che produce dati da rendere disponibili per il rilascio controllato, sia all'interno dell'organizzazione che all'esterno.
2. **Utente:** entità umana che presenta richieste (query) al sistema. L'utente può essere un dipendente dell'organizzazione a cui viene concesso l'accesso al database tramite il server, oppure un utente esterno.
3. **Client:** Front-end che trasforma le interrogazioni dell'utente in interrogazioni sui dati crittografati memorizzati sul server.
4. **Server:** Un'organizzazione che riceve i dati crittografati da un proprietario di dati e li rende disponibili per la distribuzione ai clienti.

Il server può essere di fatto di proprietà del proprietario dei dati ma, più tipicamente, è una struttura posseduta e gestita da un fornitore esterno.

Inserire figura 5.9

Esaminiamo innanzitutto la soluzione più semplice possibile basata su questo scenario.

Supponiamo che ogni singolo elemento del database sia crittografato separatamente, utilizzando la stessa chiave di crittografia. Il database crittografato è memorizzato sul server, ma il server non possiede la chiave, quindi i dati sono al sicuro sul server. Anche se qualcuno fosse in grado di penetrare nel sistema del server, avrebbe accesso solo ai dati crittografati. Il sistema client dispone di una copia della chiave di crittografia. Un utente del client può recuperare un record dal database con la seguente sequenza:

1. L'utente esegue una query SQL per i campi di uno o più record con un valore specifico della chiave primaria.
2. Il processore di query del client critta la chiave primaria, modifica la query SQL di conseguenza e la trasmette al server.
3. Il server elabora la query utilizzando il valore criptato della chiave primaria e restituisce il record o i record appropriati.
4. L'elaboratore della query decifra i dati e restituisce i risultati.

6.7 Sicurezza dei data center

Un data center è una struttura aziendale che ospita un gran numero di server, dispositivi di archiviazione, switch e apparecchiature di rete. Il numero di server e dispositivi di archiviazione può raggiungere le decine di migliaia in una singola struttura.

Esempi di utilizzo di questi grandi data center sono i fornitori di servizi cloud, i motori di ricerca, le grandi strutture di ricerca scientifica e le strutture IT per le grandi aziende. Un data center generalmente include alimentatori ridondanti o di backup, connessioni di rete ridondanti, controlli ambientali (ad esempio, aria condizionata e soppressione degli incendi) e vari dispositivi di sicurezza. I data center di grandi dimensioni sono operazioni su scala industriale che utilizzano l'energia elettrica come una piccola città. Un data center può occupare una stanza di un edificio, uno o più piani, o un intero edificio.

6.7.1 Elementi dei data center

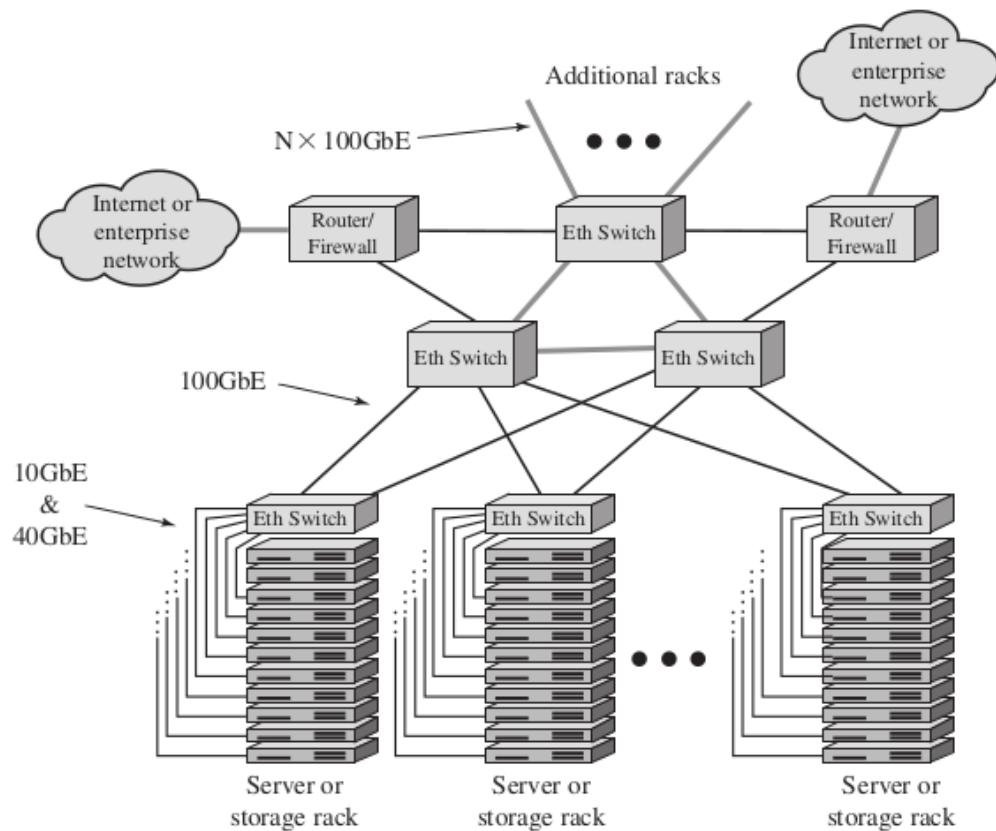


Figure 5.11 Key Data Center Elements

La Figura 5.11 illustra gli elementi chiave della configurazione di un grande data center. La maggior parte delle apparecchiature di un grande data center è costituita da pile di server e moduli di archiviazione montati in rack aperti o armadietti chiusi, che di solito sono disposti in file singole con corridoi tra di loro. Ciò consente l'accesso alla parte anteriore e posteriore di ciascun rack o armadio. In genere, i singoli moduli sono dotati di porte Ethernet da 10 o 40 Gbps per gestire il traffico massiccio da e verso i server. Inoltre, ogni rack è dotato di uno o due switch Ethernet da 10, 40 o 100 Gbps per interconnettere tutti i server e fornire connettività al resto della struttura. Gli switch sono spesso montati nel rack e vengono definiti switch top-of-rack (ToR). Il termine ToR è diventato sinonimo di switch di accesso al server, anche se non si trova "top of rack". I data center di grandi dimensioni, come i provider di cloud, richiedono switch che operano a 100 Gbps per supportare l'interconnessione dei rack di server e fornire una capacità adeguata per la connessione fuori sede tramite controller di interfaccia di rete (NIC) su router o firewall. o firewall.

Gli elementi chiave non mostrati nella Figura 5.11 sono il cablaggio e le connessioni incrociate, che possiamo elencare come segue.

- **Cross connect:** Una struttura che consente la terminazione dei cavi, nonché la loro interconnessione con altri cavi o apparecchiature.
- **Cablaggio orizzontale:** Qualsiasi cablaggio utilizzato per collegare l'armadio di cablaggio di un piano alle piastrelle a muro nelle aree di lavoro per fornire le linee della rete locale (LAN) per collegare server e altre apparecchiature digitali alla rete. Il termine orizzontale è usato perché questo tipo di cablaggio è in genere eseguito lungo il soffitto o il pavimento.
- **Cablaggio backbone:** Eseguito tra le stanze o gli armadi del data center e il punto di collegamento principale di un edificio.

6.7.2 Considerazioni sulla sicurezza dei data center

Tutte le minacce alla sicurezza e le contromisure discusse in questo testo sono rilevanti nel contesto dei grandi centri dati. Nel contesto dei grandi centri di elaborazione dati, ed è infatti in questo contesto che i rischi sono più acuti.

Si consideri che il data center ospita enormi quantità di dati che sono:

- Situati in uno spazio fisico limitato.
- Interconnessi con cablaggi a connessione diretta.
- Accessibili attraverso connessioni di rete esterne, per cui una volta superato il confine, una minaccia per l'intero complesso.
- Tipicamente rappresentano la più grande risorsa dell'azienda.

Pertanto, la sicurezza dei data center è una priorità assoluta per qualsiasi azienda con un data center di grandi dimensioni.

Alcune delle minacce più importanti da considerare sono le seguenti:

- Negazione del servizio
- Minacce persistenti avanzate da attacchi mirati
- Violazioni della privacy
- Sfruttamenti di applicazioni come l'iniezione di SQL
- Malware
- Minacce alla sicurezza fisica



Figure 5.12 Data Center Security Model

La Figura 5.12 mette in evidenza gli aspetti importanti della sicurezza dei data center, rappresentati come un modello a quattro livelli. La sicurezza del sito si riferisce principalmente alla sicurezza fisica dell'intero sito, compreso l'edificio che ospita il data center. La sicurezza fisica del data center stesso comprende barriere all'ingresso, come ad esempio una all'ingresso, come ad esempio un mantra (uno spazio di controllo dell'accesso a due porte per una sola persona) e tecniche di autenticazione per ottenere l'accesso fisico. La sicurezza della rete è estremamente importante in una struttura in cui un insieme così ampio di risorse è concentrato in un unico luogo e accessibile da connessioni di rete esterne.

6.7.3 TIA-492

Lo standard TIA (Telecommunications Industry Association) TIA-492 (Telecommunications Infrastructure Standard for Data Centers) specifica i requisiti minimi per le infrastrutture di telecomunicazione dei data center.

Questa architettura anticipa la crescita e contribuisce a creare un ambiente in cui le applicazioni e i server possono essere aggiunti e aggiornati con tempi di inattività minimi. Questo approccio standardizzato supporta l'alta disponibilità e un ambiente uniforme per l'implementazione di misure di sicurezza.

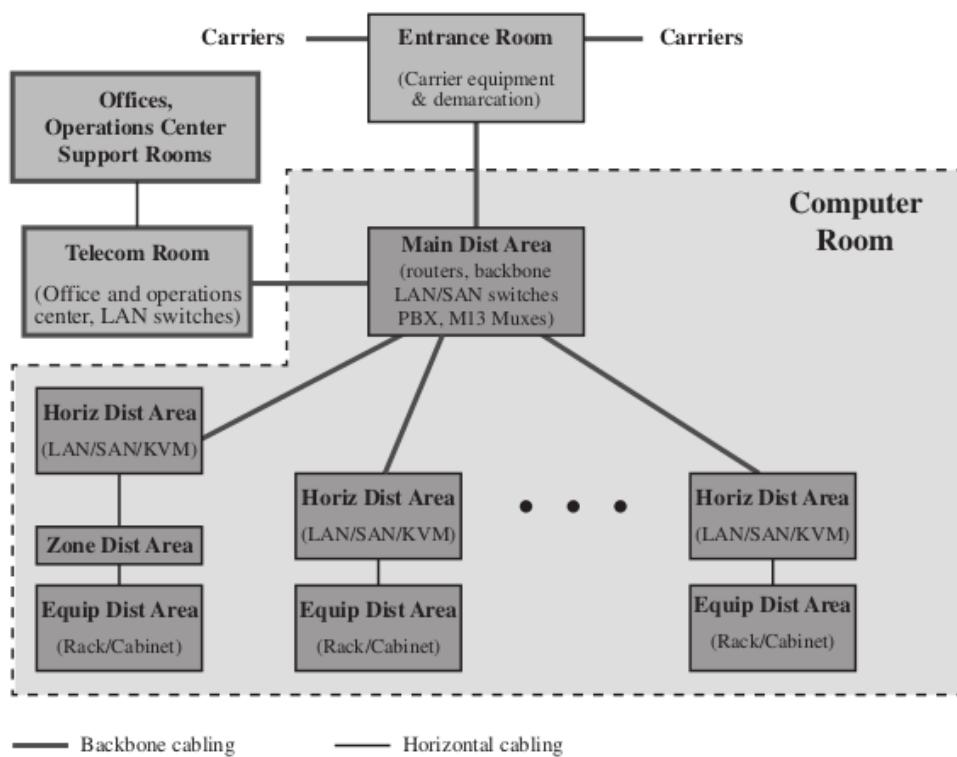


Figure 5.13 TIA-942 Compliant Data Center Showing Key Functional Areas

La norma TIA-942 specifica che un data center dovrebbe includere le seguenti aree funzionali (vedi Figura 5.13).

- **Sala computer:** Porzione del data center che ospita le apparecchiature di elaborazione dati.
- **Sala d'ingresso:** Una o più sale d'ingresso ospitano le apparecchiature esterne di accesso alla rete esterna e forniscono l'interfaccia tra le apparecchiature della sala

computer e i sistemi di cablaggio dell'azienda. e i sistemi di cablaggio aziendali. La separazione fisica della sala d'ingresso dalla sala computer sala d'ingresso dalla sala computer garantisce una maggiore sicurezza.

- **Area di distribuzione principale:** Un'area centrale che ospita il collegamento trasversale principale e i router e gli switch principali per le infrastrutture LAN e SAN (storage area network).
- **Area di distribuzione orizzontale (HDA):** Serve come punto di distribuzione per il cablaggio orizzontale e ospita le connessioni incrociate e le apparecchiature attive per la distribuzione dei cavi all'area di distribuzione delle apparecchiature.
- **Area di distribuzione delle apparecchiature (EDA):** L'ubicazione degli armadietti delle apparecchiature e dei rack, con cavi orizzontali che terminano con pannelli patch.
- **Area di distribuzione di zona (ZDA):** Un punto di interconnessione opzionale nel cablaggio orizzontale tra l'HDA e l'EDA. La ZDA può fungere da punto di consolidamento come punto di consolidamento per la flessibilità di riconfigurazione o per l'alloggiamento di apparecchiature indipendenti come i mainframe.

Capitolo 7

Capitolo 22

7.1 Posta elettronica sicura e S/MIME

S/MIME (Secure/Multipurpose Internet Mail Extension) è un miglioramento della sicurezza allo standard di formato di posta elettronica Internet MIME.

7.1.1 MIME

MIME è un'estensione del vecchio RFC 822 (Standard For The Format Of ARPA Internet Text Messages, 1982): la specifica di un formato di posta Internet. L'RFC 822 definisce una semplice intestazione con i campi "A", "Da", "Oggetto" e altri, che può essere utilizzata per instradare un messaggio di posta elettronica attraverso Internet e che fornisce informazioni di base sul contenuto dell'e-mail. RFC 822 presuppone un semplice formato di testo ASCII per il contenuto.

MIME fornisce una serie di nuovi campi di intestazione che definiscono le informazioni sul corpo del messaggio, tra cui la il corpo del messaggio, compreso il formato del corpo stesso.

Soprattutto, MIME definisce una serie di formati di contenuto, che standardizzano le rappresentazioni formati di contenuto, che standardizzano le rappresentazioni per il supporto della posta elettronica multimediale. Gli esempi includono testo, immagini, audio e video.

7.1.2 S/MIME

S/MIME è una funzionalità complessa, definita in una serie di documenti.

I documenti più importanti relativi a S/MIME sono i seguenti:

- **RFC 5750** (S/MIME Version 3.2 Certificate Handling, 2010)
Specifica le convenzioni convenzioni per l'utilizzo dei certificati X.509 da parte di (S/MIME) v3.2.

- **RFC 5751** (S/MIME Version 3.2 Message Specification, 2010)

Il principale documento di definizione per la creazione e l'elaborazione dei messaggi S/MIME.

- **RFC 4134** (Esempi di messaggi S/MIME, 2005)

Fornisce esempi di messaggi formattati utilizzando S/MIME.

- **RFC 2634** (Enhanced Security Services for S/MIME, 1999)

Describe quattro estensioni opzionali dei servizi di sicurezza per S/MIME.

- **RFC 5652** (Sintassi crittografica dei messaggi (CMS), 2009)

La sintassi crittografica è utilizzata per firmare digitalmente, digerire, autenticare o crittografare il contenuto di un messaggio contenuto di un messaggio.

- **RFC 3370** (Algoritmi CMS, 2002)

Describe le convenzioni per l'utilizzo di diversi algoritmi crittografici con il CMS.

- **RFC 5752** (Multiple Signatures in CMS, 2010)

Describe l'uso di firme multiple firme parallele per un messaggio.

- **RFC 1847** (Security Multiparts for MIME-Multipart/Signed and Multipart/ Encrypted, 1995)

Definisce un quadro di riferimento all'interno del quale i servizi di sicurezza possono essere di sicurezza alle parti del corpo MIME. L'uso di una firma digitale è rilevante per S/MIME, come spiegato in seguito.

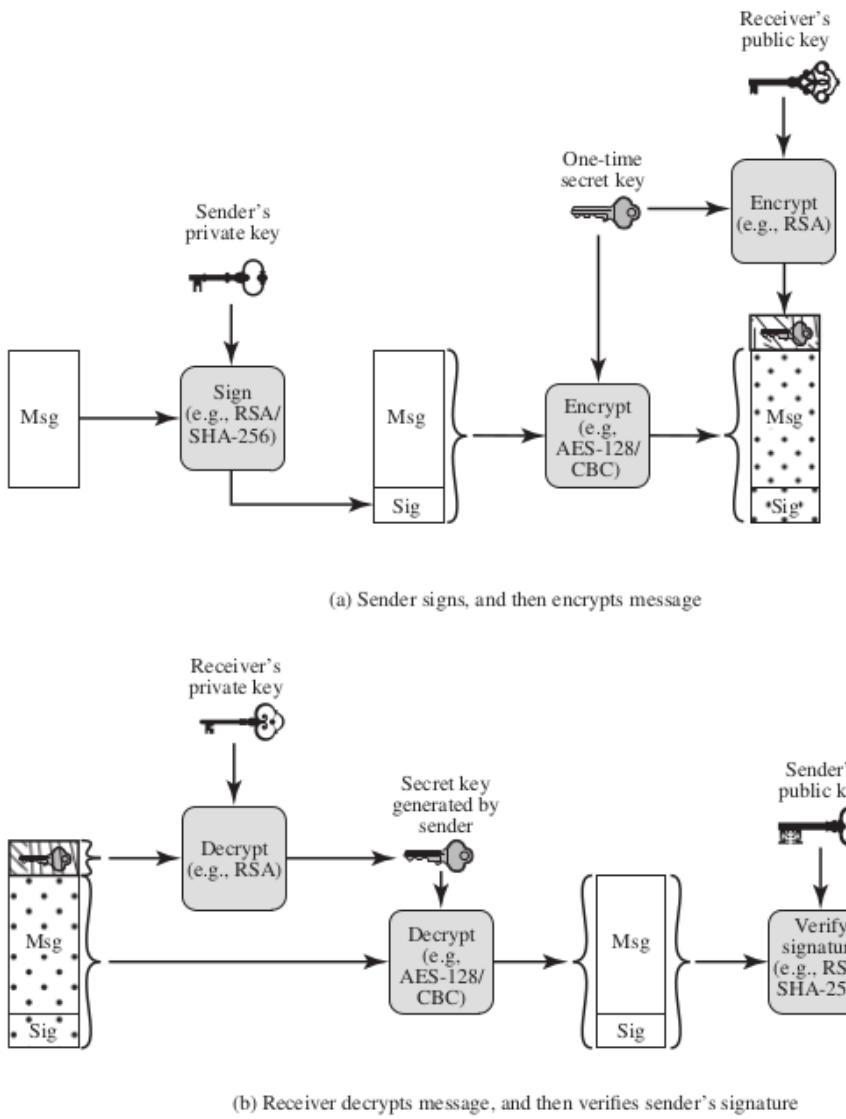


Figure 22.1 Simplified S/MIME Functional Flow

La funzionalità S/MIME è integrata nella maggior parte dei moderni software di posta elettronica e interoperano tra loro. S/MIME è definito come un insieme di tipi di contenuto MIME aggiuntivi (vedi Tabella 22.1). (vedi Tabella 22.1) e fornisce la possibilità di firmare e/o crittografare i messaggi di posta elettronica. In sostanza, questi tipi di contenuto supportano quattro nuove funzioni:

- **Dati criptati:** Consiste in contenuti criptati di qualsiasi tipo e chiavi di criptazione del contenuto criptato per uno o più tipi di messaggi di uno o più destinatari.
- **Dati firmati:** Una firma digitale si forma prendendo il message digest del contenuto da firmare, quindi crittografandolo con la chiave privata del firmatario. Il contenuto

e la firma sono poi codificati con la codifica base64. Un messaggio di dati firmato può essere visualizzato solo da un destinatario con capacità S/MIME.

- **Dati con firma in chiaro:** Come per i dati firmati, viene creata una firma digitale del contenuto. Tuttavia, in questo caso, solo la firma digitale è codificata utilizzando base64. Di conseguenza, i destinatari senza capacità S/MIME possono visualizzare il messaggio, ma non possono verificare la firma.
- **Dati firmati e imbustati:** Le entità solo firmate e solo crittografate possono essere nidificate, quindi i dati crittografati possono essere firmati e i dati firmati o con firma in chiaro possono essere crittografati.

Dati firmati e firmati in chiaro Gli algoritmi preferiti per la firma dei messaggi S/MIME utilizzano una firma RSA o un algoritmo di firma digitale (DSA) di un hash del messaggio SHA-256. Il processo funziona come segue. Si prende il messaggio che si vuole inviare e lo si mappa in un codice a lunghezza fissa di 256 bit, utilizzando SHA-256. Il digest del messaggio a 256 bit è, a tutti gli effetti, unico per questo messaggio. Quindi, S/MIME critta il digest utilizzando RSA e la chiave privata RSA del mittente. Il risultato è la firma digitale, che viene allegata al messaggio. Ora, chiunque riceva il messaggio può ricalcolare il digest del messaggio e decifrare la firma usando RSA e la chiave pubblica RSA del mittente. Se il digest del messaggio nella firma corrisponde al digest del messaggio calcolato, la firma è valida. Poiché questa operazione comporta solo la crittografia e la decrittografia di un blocco di 256 bit, richiede poco tempo. Il DSA può essere utilizzato come algoritmo di firma al posto dell'RSA. La firma è una stringa binaria, e l'invio in questa forma attraverso il sistema di posta elettronica Internet potrebbe comportare un'alterazione involontaria del contenuto, perché alcuni software di posta elettronica tenteranno di interpretare il messaggio. Per proteggere i dati, la firma da sola o la firma più il messaggio vengono mappati in caratteri ASCII stampabili utilizzando uno schema noto come mappatura radix-64 o base64. Radix-64 mappa ogni gruppo di ingresso di tre ottetti di dati binari in quattro caratteri ASCII. di dati binari in quattro caratteri ASCII (vedi Appendice G).

Dati criptati Gli algoritmi predefiniti utilizzati per la crittografia dei messaggi S/MIME sono AES e RSA. Per iniziare, S/MIME genera una chiave segreta pseudorandom. In qualsiasi applicazione di crittografia convenzionale, è necessario affrontare il problema della distribuzione della chiave. In S/MIME, ogni chiave convenzionale viene utilizzata una sola volta. Cioè, viene generata una nuova chiave pseudorandom per ogni nuova crittografia del messaggio. Questa chiave di sessione è legata al messaggio e trasmessa con esso. La chiave segreta viene utilizzata come input per l'algoritmo di crittografia a chiave pubblica RSA, che critta la chiave con la chiave pubblica RSA del destinatario. Sul lato ricevente, S/MIME utilizza la chiave privata RSA del destinatario per recuperare la chiave segreta. e poi utilizza la chiave segreta e AES per recuperare il messaggio in chiaro. Se si utilizza solo la crittografia, si usa il radix-64 per convertire il testo cifrato in formato ASCII.

Certificati a chiave pubblica Come si può notare da quanto discusso finora, S/MIME contiene un insieme intelligente, efficiente e interconnesso di funzioni e formati per fornire un efficace servizio di crittografia e firma. Per completare il sistema, è necessario affrontare un'ultima area, quella della gestione delle chiavi pubbliche. Lo strumento di base che consente un uso diffuso di S/MIME è il certificato a chiave pubblica. S/MIME utilizza certificati conformi allo standard internazionale X.509v3.

7.2 Domainkeys che identifica la posta

DomainKeys Identified Mail (DKIM) è una specifica per la firma crittografica dei messaggi di posta elettronica, che consente a un dominio firmatario di rivendicare la responsabilità di un messaggio nel flusso di posta. I destinatari dei messaggi (o gli agenti che agiscono per loro conto) possono verificare la firma interrogando direttamente il dominio del firmatario per recuperare la chiave pubblica appropriata, confermando così che il messaggio è stato attestato da una parte in possesso della chiave privata del dominio firmatario.

7.2.1 Architettura della posta elettronica

Per comprendere il funzionamento di DKIM, è utile avere una conoscenza di base dell'architettura di posta Internet, attualmente definita nella RFC 5598 (Internet Mail Architecture). Al suo livello più elementare, l'architettura della posta Internet consiste in un mondo di utenti sotto forma di **Message User Agents (MUA)** e di un mondo di trasferimento, sotto forma di **Message Handling Service (MHS)**. **Message Handling Service (MHS)**, che è composto da **Message Transfer Agents (MTA)**. L'MHS accetta un messaggio da un utente e lo consegna a uno o più altri utenti, creando un ambiente di scambio virtuale MUA-to-MUA. Questa architettura prevede tre tipi di interoperabilità. Uno è quello diretto tra utenti: i messaggi devono essere formattati dal MUA per conto dell'autore del messaggio, in modo che il messaggio possa essere visualizzato dal MUA di destinazione al destinatario del messaggio. Esistono anche requisiti di interoperabilità tra il MUA e il MHS, in primo luogo quando un messaggio viene inviato da un MUA al MHS e successivamente quando viene consegnato dal MHS al MUA di destinazione. L'interoperabilità è necessaria tra i componenti dell'MTA lungo il percorso di trasferimento attraverso l'MHS.

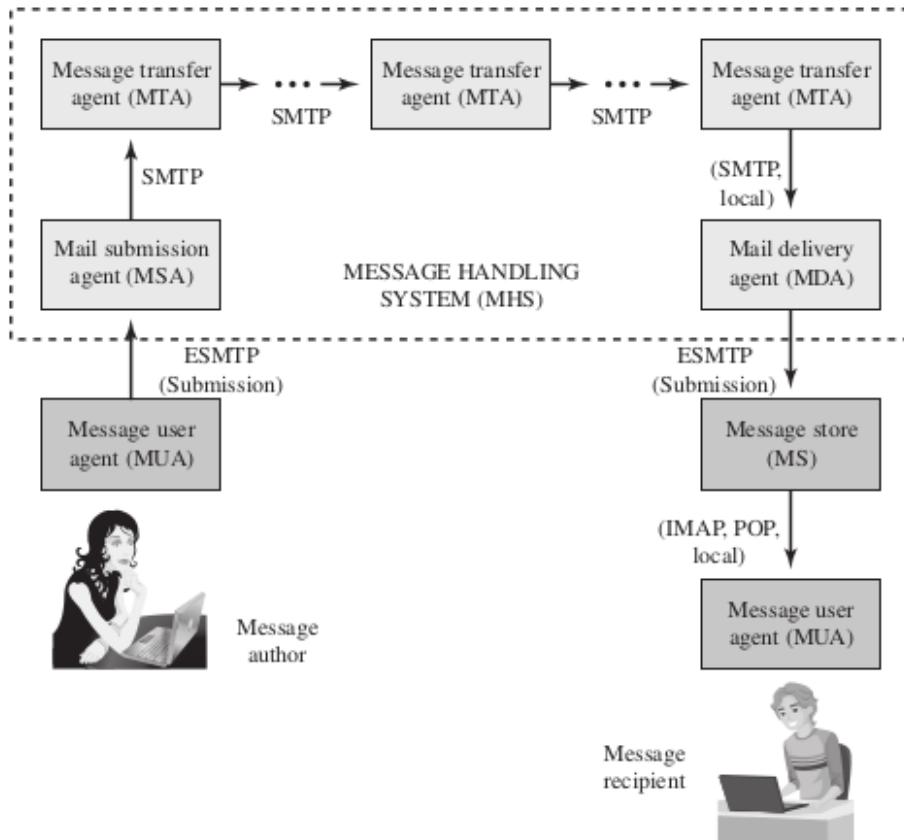


Figure 22.2 Function Modules and Standardized Protocols Used Between Them in the Internet Mail Architecture

La Figura 22.2 illustra i componenti chiave dell'architettura della posta Internet, che comprendono i seguenti elementi:

- **Agente utente del messaggio (MUA):** Lavora per conto degli attori utente e delle applicazioni utente. Utenti e delle applicazioni degli utenti. È il loro rappresentante all'interno del servizio di posta elettronica. In genere, questa funzione è ospitata nel computer dell'utente e viene indicata come programma di posta elettronica client o server di posta elettronica della rete locale.

L'autore MUA formatta un messaggio e lo invia all'MHS tramite un programma di l'invio iniziale nel sistema MHS tramite un MSA. Il MUA destinatario elabora la posta la posta ricevuta per l'archiviazione e/o la visualizzazione all'utente destinatario.

- **Agente di invio della posta (MSA):** Accetta il messaggio inviato da un MUA e applica le politiche del dominio ospitante e i requisiti degli standard Internet.

Questa funzione può essere collocata insieme al MUA o come modello funzionale separato. In quest'ultimo caso, tra il MUA e il dominio di hosting viene utilizzato il Simple Mail Transfer Protocol (SMTP).

- **Agente di trasferimento dei messaggi (MTA):** Trasmette la posta per un salto a livello di applicazione.

È simile a un commutatore di pacchetti o a un router IP, in quanto il suo compito è quello di fare valutazioni di instradamento e spostare il messaggio più vicino ai destinatari. Il relay viene eseguito da una sequenza di MTA finché il messaggio non raggiunge un MDA di destinazione. Un MTA aggiunge anche aggiunge informazioni di tracciamento all'intestazione del messaggio. SMTP viene utilizzato tra gli MTA e tra un MTA e un MSA o MDA.

- **Agente di consegna della posta (MDA):** Responsabile del trasferimento del messaggio dal MHS al MS.
- **Message store (MS):** un MUA può utilizzare un MS a lungo termine. Un MS può essere un MS può trovarsi su un server remoto o sullo stesso computer del MUA. In genere, un MUA recupera i messaggi da un server remoto usando POP (Post Office Protocol) o IMAP (Internet Message Protocol).

7.2.2 Strategia DKIM

DKIM è stato progettato per fornire una tecnica di autenticazione della posta elettronica trasparente all'utente finale. In sostanza, il messaggio di posta elettronica di un utente è firmato da una chiave privata del dominio amministrativo da cui proviene l'e-mail. La firma copre tutto il contenuto del messaggio e alcune delle intestazioni del messaggio RFC 5322 (Internet Message Format, 2008). All'estremità ricevente, l'MDA può accedere alla chiave pubblica corrispondente tramite un DNS e verificare la firma, autenticando così che il messaggio proviene dal dominio amministrativo dichiarato. Pertanto, la posta che proviene da un altro luogo ma che afferma di provenire da un determinato dominio non supererà il test di autenticazione e potrà essere rifiutata. Test di autenticazione e può essere rifiutata. Questo approccio differisce da quello di S/MIME, che utilizza la chiave privata dell'originatore per firmare i messaggi.

La motivazione del DKIM si basa sul seguente ragionamento:

1. S/MIME dipende dall'utilizzo di S/MIME da parte degli utenti mittenti e riceventi.
Per quasi tutti gli utenti, la maggior parte della posta in arrivo non utilizza S/MIME, e la maggior parte della posta che l'utente vuole ricevere non viene firmata.
2. S/MIME firma solo il contenuto del messaggio.
Pertanto, le informazioni dell'intestazione RFC 5322 sull'origine possono essere compromesse.
3. DKIM non è implementato nei programmi client (MUA) ed è quindi trasparente per l'utente.
4. DKIM si applica a tutta la posta dei domini che collaborano.

- DKIM consente ai mittenti corretti di dimostrare di aver inviato un determinato messaggio e di impedire ai falsari di mascherarsi.

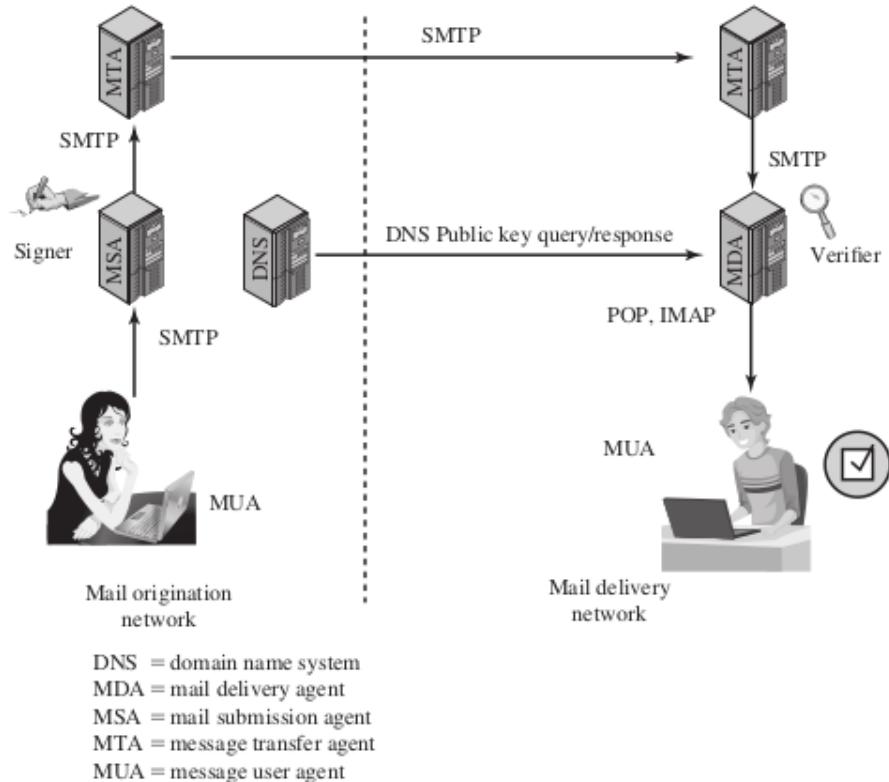


Figure 22.3 Simple Example of DKIM Deployment

La Figura 22.3 mostra un semplice esempio del funzionamento di DKIM. Si parte da un messaggio generato da un utente e trasmesso nell'MHS a un MSA che si trova nel dominio amministrativo dell'utente. Un messaggio di posta elettronica viene generato da un programma client di posta elettronica. Il contenuto del messaggio, più le intestazioni RFC 5322 selezionate, è firmato dal provider di posta elettronica dal provider di posta elettronica utilizzando la chiave privata del provider. Il firmatario è associato a un dominio, che può essere una rete locale aziendale, un ISP o una struttura di posta elettronica pubblica come Gmail. Il messaggio firmato passa poi in Internet attraverso una sequenza di MTA. A destinazione, l'MDA recupera la chiave pubblica per la firma in arrivo

e verifica la firma prima di in entrata e verifica la firma prima di passare il messaggio al client di posta elettronica di destinazione. e-mail di destinazione. L'algoritmo di firma predefinito è RSA con SHA-256. È possibile utilizzare anche RSA con SHA-1 può anche essere utilizzato.

7.3 Secure Sockets Layer (SSL) e Transport Layer Security (TLS)

Uno dei servizi di sicurezza più utilizzati è il Secure Sockets Layer (SSL) e il successivo standard Internet RFC 4346 (The Transport Layer Security (TLS) Protocol Version 1.1, 2006). TLS ha ampiamente soppiantato le precedenti implementazioni di SSL. TLS è un servizio di uso generale implementato come una serie di protocolli che si basano su TCP. A questo livello, ci sono due scelte di implementazione. Per ottenere la massima generalità, TLS potrebbe essere fornito come parte della suite di protocolli sottostante e quindi essere trasparente alle applicazioni. In alternativa, TLS può essere incorporato in pacchetti specifici. Ad esempio, la maggior parte dei browser è dotata di SSL e la maggior parte dei server Web ha implementato il protocollo.

7.3.1 Architettura TLS

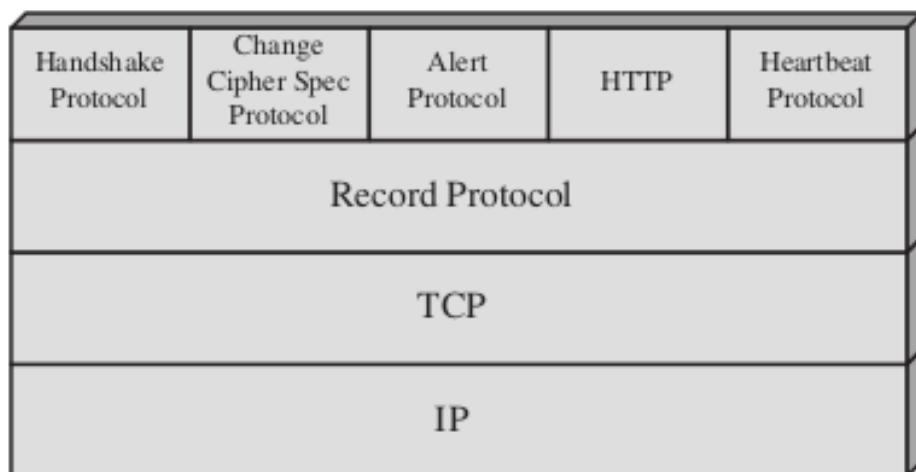


Figure 22.4 SSL/TLS Protocol Stack

TLS è stato progettato per utilizzare il protocollo TCP e fornire un servizio sicuro end-to-end affidabile. TLS non è un singolo protocollo, ma piuttosto due livelli di protocolli, come illustrato nella Figura 22.4. Il protocollo Record fornisce servizi di sicurezza di base a vari protocolli di livello superiore. Il protocollo Record fornisce servizi di sicurezza di base a vari protocolli di livello superiore. In particolare, l'Hypertext Transfer Protocol (HTTP), che fornisce il servizio di trasferimento per l'interazione tra client e server Web, può operare su TLS. Tre protocolli di livello superiore sono definiti come parte di TLS: **il protocollo Handshake, il protocollo Change Cipher Spec e il protocollo Alert**.

Questi protocolli specifici di TLS sono utilizzati nella gestione degli scambi TLS e sono esaminati più avanti in questa sezione. Due importanti concetti di TLS sono la sessione TLS e la connessione TLS, definiti nella specifica come segue:

- **Connessione:** Una connessione è un trasporto (secondo la definizione del modello di stratificazione OSI) che fornisce un tipo di servizio adeguato. Per TLS, tali connessioni sono relazioni peer-to-peer.

Le connessioni sono transitorie. Ogni connessione è associata a una sessione.

- **Sessione:** Una sessione TLS è un'associazione tra un client e un server. Le sessioni sono create dal protocollo Handshake.

Le sessioni definiscono un insieme di parametri di sicurezza parametri di sicurezza, che possono essere condivisi tra più connessioni. Le sessioni sono utilizzate per evitare la costosa negoziazione di nuovi parametri di sicurezza per ogni connessione.

Tra qualsiasi coppia di parti (applicazioni come HTTP su client e server), possono esistere più connessioni sicure. In teoria, possono esistere anche più sessioni simultanee tra le parti, ma questa caratteristica non viene utilizzata nella pratica.

7.3.2 Protocollo TLS

Protocollo di registrazione Il protocollo di registrazione SSL fornisce due servizi per le connessioni SSL. per le connessioni SSL:

- **Riservatezza:** Il protocollo Handshake definisce una chiave segreta condivisa che viene utilizzata per la crittografia simmetrica dei payload SSL.
- **Integrità del messaggio:** Il protocollo di handshake definisce anche una chiave segreta condivisa che viene utilizzata per formare un codice di autenticazione dei messaggi (MAC).

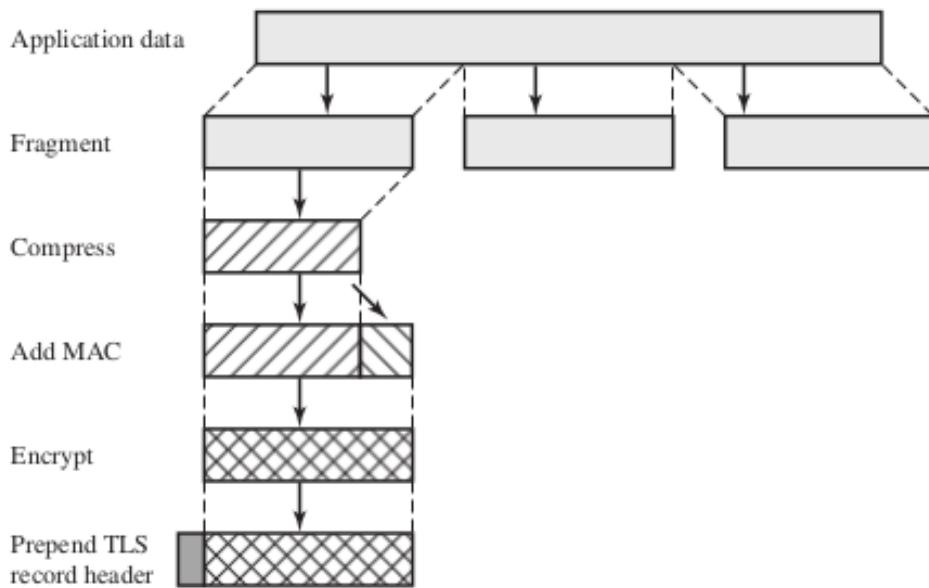


Figure 22.5 TLS Record Protocol Operation

La Figura 22.5 indica il funzionamento complessivo del protocollo SSL Record. Il primo passo è la frammentazione. Ogni messaggio di livello superiore viene frammentato in blocchi di 214 byte (16.384 byte) o meno. Successivamente, viene applicata facoltativamente la compressione. La fase successiva dell'elaborazione consiste nel calcolare un codice di autenticazione del messaggio sui dati compressi. Successivamente, il messaggio compresso e il MAC vengono crittografati utilizzando la crittografia simmetrica. L'ultima fase dell'elaborazione del protocollo SSL Record è l'aggiunta di un'intestazione, che comprende i campi di versione e lunghezza. I tipi di contenuto definiti sono `change_cipher_spec`, `alert`, `handshake` e `application_data`.

I primi tre sono i protocolli specifici di TLS, di cui si parlerà in seguito. Si noti che non viene fatta alcuna distinzione tra le varie applicazioni (ad es, HTTP) che potrebbero utilizzare TLS; il contenuto dei dati creati da tali applicazioni non è visibile a TLS. Il protocollo Record trasmette quindi l'unità risultante in un segmento TCP. I dati ricevuti vengono decifrati, verificati, decompressi e riassemblati, quindi consegnati agli utenti di livello superiore. agli utenti di livello superiore.

Change Cipher Spec Protocol Il Change Cipher Spec Protocol è uno dei quattro protocolli specifici di TLS. quattro protocolli specifici di TLS che utilizzano il TLS Record Protocol ed è il più semplice. Questo protocollo è costituito da un singolo messaggio, che consiste in un singolo byte con valore valore 1. L'unico scopo di questo messaggio è far sì che lo stato in sospeso venga copiato nello stato corrente. nello stato corrente, che aggiorna la suite di cifratura da utilizzare su questa connessione.

Protocollo di avviso Il protocollo di avviso viene utilizzato per trasmettere all'entità peer avvisi relativi a TLS. Come per altre applicazioni che utilizzano TLS, i messaggi di avviso sono compressi e crittografati, come specificato e criptati, come specificato dallo stato corrente. Ogni messaggio di questo protocollo è composto da due byte. Il primo byte assume il valore **warning** o **fatal** per indicare la gravità del messaggio. Se il livello è fatale, TLS termina immediatamente la connessione. Altre connessioni sulla stessa sessione possono continuare, ma non possono essere stabilite nuove connessioni su questa sessione. Il secondo byte contiene un codice che indica l'avviso specifico. Un esempio di avviso fatale è un MAC errato. Un esempio di avviso non fatale è un messaggio `close_notify`, che informa il destinatario che il mittente non invierà più messaggi su questa connessione.

Protocollo Handshake La parte più complessa di TLS è il protocollo Handshake. Questo protocollo consente al server e al client di autenticarsi reciprocamente e di negoziare un algoritmo di crittografia e MAC e le chiavi crittografiche da utilizzare per proteggere i dati inviati in un record TLS. Il protocollo Handshake viene utilizzato prima di trasmettere i dati dell'applicazione.

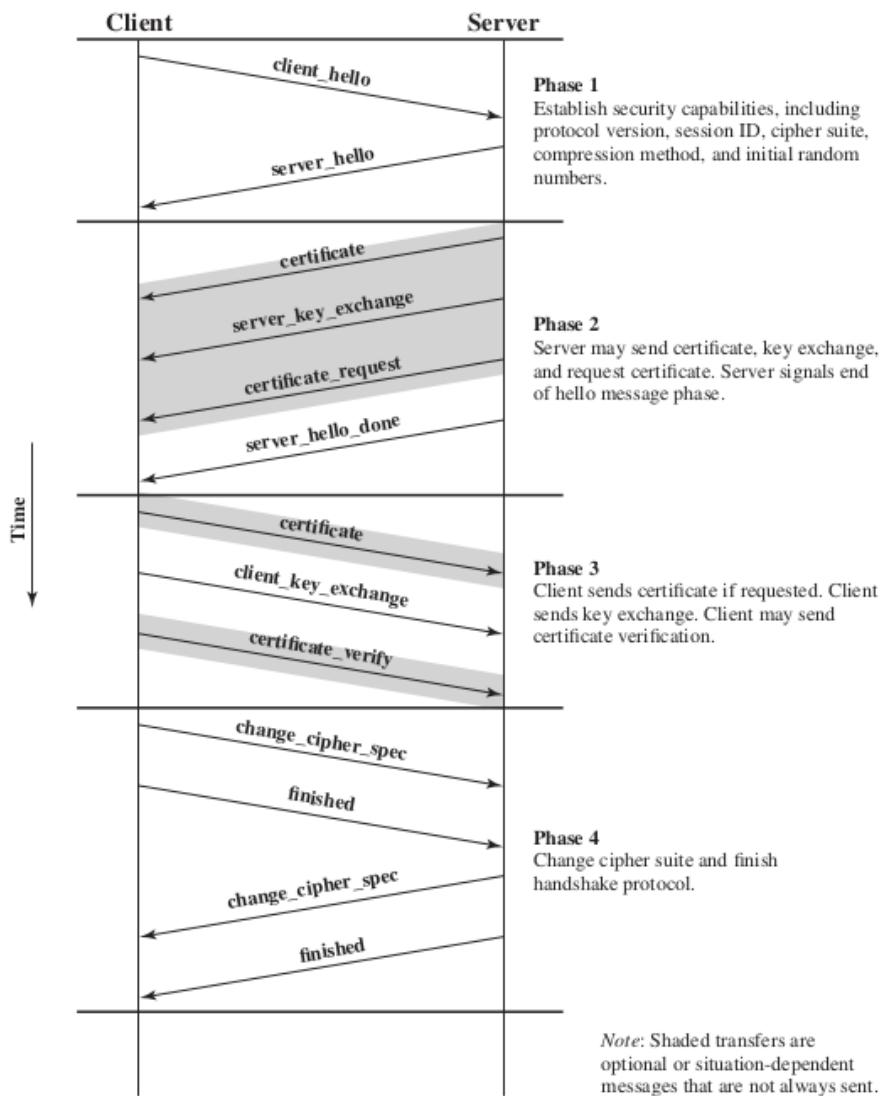


Figure 22.6 Handshake Protocol Action

Il protocollo Handshake consiste in una serie di messaggi scambiati da client e server.

La Figura 22.6 mostra lo scambio iniziale necessario per stabilire una connessione logica tra client e server. Lo scambio può essere visto come composto da quattro fasi.

La fase 1 serve ad avviare una connessione logica e a stabilire le capacità di sicurezza che vi saranno associate. Lo scambio è avviato dal client, che invia un messaggio `client_hello` con i seguenti parametri:

- Versione: La versione TLS più alta compresa dal client.

- Casuale: Una struttura casuale generata dal client, composta da un timestamp a 32 bit e da 28 byte generati da un generatore di numeri casuali sicuri.
- ID sessione: Un identificatore di sessione di lunghezza variabile. Un valore non nullo indica che un valore non nullo indica che il client desidera aggiornare i parametri di una connessione esistente o creare una nuova connessione su questa sessione. Un valore nullo indica che il client desidera stabilire una nuova connessione in una nuova sessione.
- CipherSuite: È un elenco che contiene le combinazioni di algoritmi crittografici supportati dal client, crittografiche supportate dal client, in ordine decrescente di preferenza. Ogni elemento dell'elenco (ogni suite di cifratura) definisce sia un algoritmo di scambio di chiavi che un CipherSpec.
- Metodo di compressione: È un elenco dei metodi di compressione supportati dal client.

Dopo aver inviato il messaggio `client_hello`, il client attende il messaggio `server_hello`, che contiene gli stessi parametri del messaggio `client_hello`.

Fase 2 i dettagli dipendono dallo schema di crittografia a chiave pubblica utilizzato. In alcuni casi, il server passa un certificato al client, eventualmente informazioni aggiuntive sulla chiave e una richiesta di certificato da parte del client. Il messaggio finale della fase 2, che è sempre richiesto, è il messaggio `server_done`, che viene inviato dal server per indicare la fine del server hello e dei messaggi associati. Dopo l'invio di questo messaggio, il server attende la risposta del client.

Fase 3 dopo aver ricevuto il messaggio `server_done`, il client deve verificare che il server abbia fornito un certificato valido se il server abbia fornito un certificato valido, se richiesto, e verificare che i parametri di `server_hello` siano accettabili. Se tutto è soddisfacente, il client invia uno o più messaggi al server, a seconda dello schema a chiave pubblica sottostante.

Fase 4 completa l'impostazione di una connessione sicura. Il client invia un messaggio `change_cipher_spec` e copia il `CipherSpec` in sospeso nel `CipherSpec` corrente. Si noti che questo messaggio non è considerato parte del protocollo di handshake, ma viene ma viene inviato utilizzando il protocollo Change Cipher Spec. Il client invia quindi immediatamente il messaggio finito con i nuovi algoritmi, chiavi e segreti. Il messaggio finito verifica che i processi di scambio di chiavi e di autenticazione siano andati a buon fine. In risposta a questi due messaggi, il server invia il proprio messaggio `change_cipher_spec`, trasferisce il pending al `CipherSpec` corrente e invia il suo messaggio finito. A questo punto, l'handshake è completo e il client e il server possono iniziare a scambiare dati di livello applicativo.

Protocollo Heartbeat Nell'ambito delle reti di computer, un heartbeat è un segnale di pericolo generato dall'hardware o dal software per indicare il normale funzionamento o per indicare la presenza di un'interfaccia. segnale periodico generato dall'hardware o dal software per indicare il normale funzionamento o per sincronizzare altre parti del sistema.

Il protocollo Heartbeat si basa sul protocollo TLS Record e si compone di due tipi di messaggi: heartbeat (cuore), heartbeat (cuore) e heartbeat (cuore), due tipi di messaggi: heartbeat_request e heartbeat_response. L'uso del protocollo Heartbeat viene stabilito durante la fase 1 del protocollo Handshake (vedi Figura 22.6).

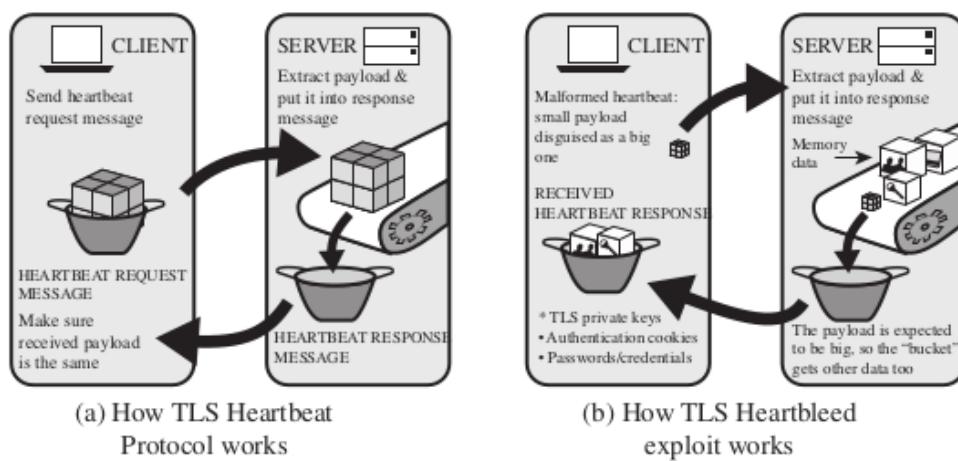


Figure 22.7 The Heartbleed Exploit

Source: "Heartbleed-The Open SSL Heartbeat Exploit" Copyright © 2014 BAE Systems Applied Intelligence. Reprinted with permission.

Ogni peer indica se supporta gli heartbeat. Se i battiti cardiaci sono supportati, il peer indica se è disposto a ricevere messaggi heartbeat_request e a rispondere con messaggi heartbeat_request. e rispondere con messaggi heartbeat_response o se è disposto solo a inviare messaggi heartbeat_request. Un messaggio heartbeat_request può essere inviato in qualsiasi momento. Ogni volta che viene ricevuto un messaggio di richiesta, si deve rispondere prontamente con un corrispondente messaggio di risposta heartbeat di risposta.

Il messaggio heartbeat_request include la lunghezza del payload, il payload, e campi di riempimento. Il payload è un contenuto casuale di lunghezza compresa tra 16 byte e 64 Kbyte. di lunghezza. Il messaggio heartbeat_response corrispondente deve includere una copia esatta del payload ricevuto. del payload ricevuto. Anche il padding è un contenuto casuale. Il padding consente al mittente di eseguire un'operazione di scoperta dell'unità massima di trasferimento (MTU) del percorso, inviando richieste con padding crescente fino a quando non si ottiene più risposta.

Il battito cardiaco ha due scopi. In primo luogo, assicura al mittente che il destinatario è ancora vivo, anche se per un po' di tempo non c'è stata alcuna attività sulla connessione TCP sottostante. In secondo luogo, l'heartbeat genera attività sulla connessione durante i periodi di inattività, evitando così la chiusura da parte di un firewall che non tollera le

connessioni inattive. Il requisito dello scambio di un carico utile è stato progettato nel sistema Heartbeat per supportarne l'uso in una versione senza connessione di TLS, nota come DTLS. Poiché un servizio senza connessione è soggetto alla perdita di pacchetti, il carico utile consente al richiedente di abbinare i messaggi di risposta alle richieste. Per semplicità, la stessa versione del protocollo Heartbeat viene utilizzata sia con TLS che con DTLS. Pertanto, il carico utile è necessario sia per TLS che per DTLS.

7.3.3 Attacchi SSL/TLS

Dalla prima introduzione di SSL nel 1994 e dalla successiva standardizzazione di TLS, sono stati ideati numerosi attacchi contro questi protocolli. La comparsa di ogni attacco ha reso necessarie modifiche al protocollo, agli strumenti di crittografia utilizzati o ad alcuni aspetti dell'implementazione di SSL e TLS per contrastare queste minacce.

Categorie di attacchi Possiamo raggruppare gli attacchi in quattro categorie generali:

- Attacchi al protocollo Handshake: Già nel 1998, un approccio al protocollo Handshake basato sul promettere il protocollo Handshake basato sullo sfruttamento della formattazione e dell'implementazione dello schema di crittografia RSA. Con l'implementazione di contromisure sono state implementate, l'attacco è stato perfezionato e adattato per non solo per vanificare le contromisure, ma anche per velocizzare l'attacco.
- Attacchi ai protocolli di registrazione e di dati applicativi: Sono state scoperte numerose vulnerabilità in questi protocolli.
- Attacchi alla PKI: la verifica della validità dei certificati X.509 è un'attività soggetta a una serie di attacchi, sia in una serie di attacchi, sia nel contesto di SSL/TLS che altrove.

7.4 HTTPS

HTTPS (HTTP over SSL) si riferisce alla combinazione di HTTP e SSL per implementare una comunicazione sicura tra un browser Web e un server Web. La funzionalità HTTPS è integrata in tutti i browser Web moderni. Il suo utilizzo dipende dal server Web che supporta la comunicazione HTTPS. La differenza principale per l'utente di un browser Web è che gli indirizzi URL (uniform resource locator) iniziano con `https://` invece che con `http://`. Una normale connessione HTTP utilizza la porta 80. Se viene specificato HTTPS, viene utilizzata la porta 443, che richiama SSL.

Quando si utilizza HTTPS, i seguenti elementi della comunicazione sono crittografati:

- URL del documento richiesto
- Contenuto del documento
- Contenuto dei moduli del browser (compilati dall'utente del browser)
- Cookie inviati dal browser al server e dal server al browser
- Contenuto dell'intestazione HTTP

7.4.1 Avvio della connessione

Per HTTPS, l'agente che agisce come client HTTP agisce anche come client TLS. Il client avvia una connessione al server sulla porta appropriata e invia il ClientHello TLS per iniziare l'handshake TLS. Quando l'handshake TLS è terminato, il client può avviare la prima richiesta HTTP. Tutti i dati HTTP devono essere inviati come dati dell'applicazione TLS. Quando l'handshake TLS è terminato, il client può avviare la prima richiesta HTTP. È necessario chiarire che ci sono tre livelli di consapevolezza di una connessione in HTTPS. A livello HTTP, un client HTTP richiede una connessione a un server HTTP inviando una richiesta di connessione al livello immediatamente inferiore. In genere, il livello più basso è il TCP, ma può anche essere il TLS/SSL. A livello di TLS, viene stabilita una sessione tra un client TLS e un server TLS. Questa sessione può supportare una o più connessioni in qualsiasi momento. Come abbiamo visto, una richiesta TLS per stabilire una connessione inizia con la creazione di una connessione TCP tra l'entità TCP sul lato client e l'entità TCP sul lato server.

7.4.2 Chiusura della connessione

Un client o un server HTTP può indicare la chiusura di una connessione includendo la linea seguente in un record HTTP: **Connection: close**. Questo indica che la connessione sarà chiusa dopo la consegna di questo record.

La chiusura di una connessione HTTPS richiede che TLS chiuda la connessione con l'entità peer TLS sul server, con l'entità peer TLS sul lato remoto, il che comporta la chiusura della connessione TCP sottostante. A livello di TLS, il modo corretto per chiudere una connessione è che ciascuna parte utilizzi il protocollo di avviso TLS per inviare un avviso `close_notify`. Le implementazioni TLS devono avviare uno scambio di avvisi di chiusura prima di chiudere una connessione. Un'implementazione TLS può, dopo l'invio di un avviso di chiusura, chiudere la connessione senza attendere che il peer invii il suo avviso di chiusura, generando una "chiusura incompleta". Si noti che un'implementazione che fa questo può scegliere di riutilizzare la sessione. Questo dovrebbe essere fatto solo quando l'applicazione sa (tipicamente attraverso il rilevamento dei limiti dei messaggi HTTP) di aver ricevuto tutti i dati del messaggio che le interessano.

I client HTTP devono inoltre essere in grado di gestire una situazione in cui la connessione TCP viene terminata senza un avviso precedente di `close_notify` e senza un indicatore di **Connection: close** indicator. Tale situazione potrebbe essere dovuta a un errore di programmazione del server o a un errore di comunicazione che causa l'interruzione della connessione TCP. Tuttavia, la chiusura TCP non annunciata potrebbe essere la prova di un qualche tipo di attacco. Pertanto, il client HTTPS dovrebbe emettere una sorta di avviso di sicurezza quando ciò si verifica.

7.5 Sicurezza IPv4 e IPv6

La comunità di Internet ha sviluppato meccanismi di sicurezza specifici per le applicazioni di sicurezza specifici per le applicazioni in diverse aree, tra cui la posta elettronica (S/MIME), il client/server (Kerberos), l'accesso al Web (SSL) e altri ancora. Tuttavia, gli utenti hanno alcune preoccupazioni sulla sicurezza che riguardano livelli di protocollo.

Per esempio, un'azienda può gestire una rete TCP/IP privata e sicura disabilitando i collegamenti a siti non attendibili, criptando i pacchetti che lasciano la sede, e autenticando i pacchetti che vi entrano. Implementando la sicurezza a livello IP, un'organizzazione può garantire la sicurezza della rete non solo per le applicazioni che utilizzano meccanismi di sicurezza, ma anche per le molte applicazioni ignare della sicurezza.

In risposta a questi problemi, l'Internet Architecture Board (IAB) ha incluso l'autenticazione e la crittografia come elementi necessari di autenticazione e crittografia come caratteristiche di sicurezza necessarie nell'IP di nuova generazione, che è stato rilasciato con il nome di IP. IP di prossima generazione, che è stato rilasciato con il nome di IPv6. Fortunatamente, queste funzionalità di sicurezza sono state fortunatamente, queste funzionalità di sicurezza sono state progettate per essere utilizzabili sia con l'attuale IPv4 che con il futuro IPv6.

Questo significa che i fornitori possono iniziare a offrire queste funzionalità fin da subito e molti di essi hanno già alcuni prodotti con funzionalità IPsec. La sicurezza a livello IP comprende tre aree funzionali:

1. Autenticazione
2. Confidenza
3. Gestione delle chiavi

Il meccanismo di autenticazione assicura che un messaggio ricevuto sia stato trasmesso dalla parte identificata come sorgente nell'intestazione del pacchetto. Inoltre, questo meccanismo assicura che il pacchetto non sia stato alterato durante il transito. La funzione di confidenzialità consente ai nodi comunicanti di crittografare i messaggi per impedire l'intercettazione da parte di terzi. La funzione di gestione delle chiavi si occupa lo scambio sicuro di chiavi. La versione attuale di IPsec, nota come IPsecv3, comprende l'autenticazione e la riservatezza. La gestione delle chiavi è fornita dallo standard Internet Key Exchange, IKEv2.

Applicazioni di Ipisec IPsec offre la possibilità di proteggere le comunicazioni attraverso una LAN, una WAN privata e pubblica e Internet.

Esempi di utilizzo sono i seguenti:

- **Connettività sicura delle filiali su Internet:** Un'azienda può creare una rete privata virtuale sicura su Internet o su una WAN pubblica. In questo modo Internet è ridurre la necessità di reti private, risparmiando sui costi e sulla gestione delle reti private, risparmiando sui costi e sulle spese di gestione della rete.
- **Accesso remoto sicuro via Internet:** Un utente finale il cui sistema è dotato di sicurezza IP può effettuare una chiamata locale a un provider di servizi Internet e accedere in modo sicuro alla rete aziendale. In questo modo si riducono i costi di pedaggio di pedaggio per i dipendenti in viaggio e i telelavoratori.
- **Stabilire una connettività extranet e intranet con i partner:** IPsec può essere utilizzato per proteggere le comunicazioni con altre organizzazioni, garantendo l'autenticazione e la riservatezza e fornendo uno scambio di chiavi.
- **Migliorare la sicurezza del commercio elettronico:** Anche se alcune applicazioni per il Web e il commercio elettronico dispongono di protocolli di sicurezza integrati, l'uso di IPsec migliora tale sicurezza.

La caratteristica principale di IPsec, che gli consente di supportare queste diverse applicazioni, è la possibilità di criptare e che può crittografare e/o autenticare tutto il traffico a livello IP. In questo modo, tutte le applicazioni distribuite, tra cui accesso remoto, client/server, e-mail, trasferimento di file, accesso al Web e così via, accesso al Web e così via, possono essere protette. La Figura 9.3 mostra uno scenario tipico di utilizzo di IPsec.

Benefici di IPsec I vantaggi di IPsec sono i seguenti:

- Quando IPsec è implementato in un firewall o in un router, fornisce una forte sicurezza che può essere applicata a tutto il traffico che attraversa il perimetro.
- IPsec in un firewall è resistente all'aggiramento se tutto il traffico proveniente dall'esterno deve utilizzare il protocollo IP e il firewall è l'unico mezzo di accesso al perimetro.
- IPsec si trova al di sotto del livello di trasporto (TCP, UDP) ed è quindi trasparente alle applicazioni. Non è necessario modificare il software di un sistema utente o server quando IPsec è implementato nel firewall o nel router.
- IPsec può essere trasparente per gli utenti finali. Non è necessario formare gli utenti sui meccanismi di sicurezza, di rilasciare materiale di chiave per ogni utente o di revocare il materiale di chiave quando gli utenti lasciano l'organizzazione.
- Se necessario, IPsec può garantire la sicurezza per i singoli utenti. Questo è utile per i lavoratori fuori sede e per la creazione di una sottorete virtuale sicura all'interno di un'organizzazione per le applicazioni sensibili.

Applicazioni di routing oltre a supportare gli utenti finali e a proteggere i sistemi e le reti di una rete e dei sistemi delle sedi, IPsec può svolgere un ruolo fondamentale nell'architettura di routing necessaria per l'internetworking.

I seguenti esempi di utilizzo di IPsec che può garantire che:

- Un annuncio di router (un nuovo router pubblica la sua presenza) provenga da un router autorizzato.
- Un annuncio di vicinato (un router cerca di stabilire o mantenere una relazione di vicinato con un router di un altro paese).
- Un messaggio di reindirizzamento proviene dal router a cui è stato inviato il pacchetto iniziale.
- Un aggiornamento di routing non viene falsificato.

Senza queste misure di sicurezza, un avversario può interrompere le comunicazioni o deviare il traffico. I protocolli di routing, come l'Open Shortest Path First (OSPF), devono essere eseguiti in base a essere eseguiti sulla base di associazioni di sicurezza tra router definite da IPsec.

7.5.1 Il campo di applicazione di IPsec

IPsec fornisce due funzioni principali: una funzione combinata di autenticazione/crittografia chiamata Encapsulating Security Payload (ESP) e una funzione di scambio di chiavi. Per le reti private virtuali, in genere si desiderano sia l'autenticazione che la crittografia, perché è importante

1. Assicurare che gli utenti non autorizzati non penetrino nella rete privata virtuale
2. Assicurare che gli intercettatori su Internet non possano leggere i messaggi inviati attraverso la rete privata virtuale

Esiste anche una funzione di sola autenticazione, implementata mediante un'intestazione di autenticazione (AH). Poiché l'autenticazione dei messaggi è fornita da ESP, l'uso di AH è deprecato. È incluso in IPsecv3 per compatibilità con il passato, ma non dovrebbe essere utilizzato in nuove applicazioni. In questo capitolo non si parlerà di AH. La funzione di scambio di chiavi consente lo scambio manuale delle chiavi e uno schema automatizzato.

7.5.2 Security Associations

Un concetto chiave che compare in entrambi i meccanismi di autenticazione e riservatezza per IP è l'associazione di sicurezza (SA). Un'associazione è una relazione unidirezionale tra un mittente e un destinatario che offre servizi di sicurezza al traffico trasportato su di essa. Se è necessaria una relazione tra pari, per uno scambio sicuro bidirezionale, sono necessarie

due associazioni di sicurezza. I servizi di sicurezza sono garantiti da una SA per l'uso di ESP.

Una SA è identificata in modo univoco da tre parametri:

- **Indice del parametro di sicurezza (SPI):** Una stringa di bit assegnata a questa SA e avente significato solo a livello locale. L'SPI è contenuto in un'intestazione ESP per consentire al sistema ricevente di selezionare il SA in base al quale il SA è stato inviato.
- **Indirizzo di destinazione IP:** È l'indirizzo dell'endpoint di destinazione del SA, che può essere un sistema dell'utente finale o un sistema di rete come un firewall o un router.
- **Identificatore di protocollo:** Questo campo dell'intestazione IP esterna indica se l'associazione è un protocollo AH o ESP.

Un'implementazione di IPsec include un database di associazioni di sicurezza che definisce i parametri associati a ogni SA. Un SA è caratterizzato dai seguenti parametri:

- Contatore del numero di sequenza: Un valore a 32 bit usato per generare il campo Sequence Number nelle intestazioni AH o ESP.
- Overflow del contatore di sequenza: Un flag che indica se l'overflow del contatore di numeri di sequenza deve generare un evento verificabile e impedire l'ulteriore transazione di pacchetti su questa SA.
- Finestra antireplay: Utilizzata per determinare se un pacchetto AH o ESP in entrata è un replay, definendo una finestra scorrevole un replay, definendo una finestra scorrevole entro la quale il numero di sequenza deve rientrare.
- Informazioni AH: Algoritmo di autenticazione, chiavi, durata delle chiavi e parametri correlati utilizzati con AH.
- Informazioni su ESP: Algoritmo di crittografia e di autenticazione, chiavi, valori di inizializzazione, durate delle chiavi e parametri correlati.
- Durata di questa associazione di sicurezza: Un intervallo di tempo o un conteggio di byte dopo il quale una SA deve essere sostituita da una nuova SA (e da un nuovo SPI) o terminata, oltre a un'indicazione di quale di queste azioni deve avvenire.
- Modalità del protocollo IPsec: Tunnel, trasporto o wildcard (richiesto per tutte le implementazioni).
- MTU del percorso: Qualsiasi unità di trasmissione massima del percorso osservata (dimensione massima di un pacchetto che può essere trasmesso senza frammentazione) e variabili di invecchiamento.

Il meccanismo di gestione delle chiavi utilizzato per la distribuzione delle chiavi è collegato ai meccanismi di meccanismi di autenticazione e privacy solo attraverso i parametri di sicurezza. Pertanto, l'autenticazione e la privacy sono state specificate indipendentemente da qualsiasi meccanismo di gestione delle chiavi.

7.5.3 L'Encapsulating Security Payload

Fornisce servizi di riservatezza, tra cui la riservatezza del contenuto dei messaggi e la riservatezza limitata del flusso di traffico. Come caratteristica opzionale, ESP può anche fornire un servizio di autenticazione.

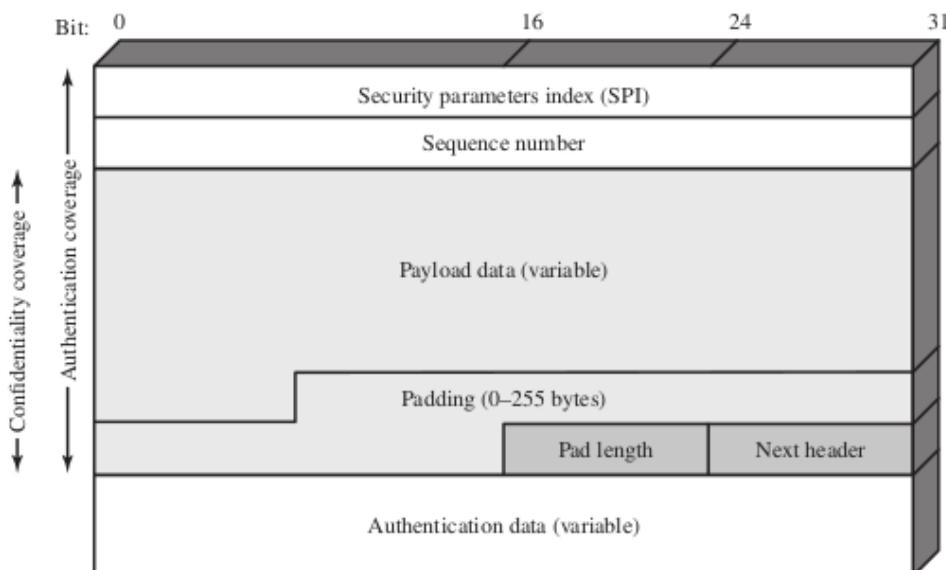


Figure 22.8 IPsec ESP Format

La Figura 22.8 mostra il formato di un pacchetto ESP. Esso contiene i seguenti campi:

- **Indice dei parametri di sicurezza (32 bit)**: Identifica un'associazione di sicurezza.
- **Numero di sequenza (32 bit)**: Un valore di contatore monotonicamente crescente.
- **Dati del carico utile (variabile)**: Si tratta di un segmento a livello di trasporto (modalità di trasporto) o di un pacchetto IP (modalità tunnel) protetto da crittografia.
- **Padding (0-255 byte)**: Può essere necessario se l'algoritmo di crittografia richiede che il testo in chiaro sia il testo in chiaro sia un multiplo di un certo numero di ottetti.
- **Lunghezza pad (8 bit)**: Indica il numero di byte di pad immediatamente precedenti a questo campo.

- **Intestazione successiva (8 bit):** Identifica il tipo di dati contenuti nel campo Dati payload identificando la prima intestazione del carico utile (ad esempio, un'intestazione di estensione in IPv6 o un'intestazione superiore).
- **Valore di controllo dell'integrità (variabile):** Un campo di lunghezza variabile (deve essere un numero integrale di parole a 32 bit) che contiene il valore di controllo dell'integrità calcolato su il pacchetto ESP meno il campo Dati di autenticazione.

7.5.4 Transport and Tunnel Modes

L'ESP supporta due modalità di utilizzo: quella di trasporto e quella di tunnel. Iniziamo questa sezione con una breve panoramica.

Modalità di trasporto La modalità di trasporto fornisce protezione principalmente ai protocolli di livello superiore. In altre parole, la protezione in modalità trasporto si estende al carico utile di un pacchetto IP. Tra gli esempi vi sono i segmenti TCP o UDP, che operano entrambi direttamente sopra l'IP nello stack di protocolli dell'host. In genere, la modalità di trasporto viene utilizzata per la comunicazione end-to-end tra due host (ad esempio, un client e un server o due workstation). Quando un host esegue ESP su IPv4, il payload è costituito dai dati che normalmente seguono l'intestazione IP. Per IPv6, il payload è costituito dai dati che normalmente seguono l'intestazione IP e le eventuali intestazioni di estensione IPv6 presenti, con l'eventuale eccezione dell'intestazione delle opzioni di destinazione di destinazione, che può essere inclusa nella protezione. ESP in modalità di trasporto critta e, facoltativamente, autentica il payload IP, ma non l'intestazione IP.

Modalità tunnel La modalità tunnel fornisce protezione all'intero pacchetto IP. Per ottenere questo risultato, dopo che i campi ESP sono stati aggiunti al pacchetto IP, l'intero pacchetto più i campi di sicurezza viene trattato come il carico utile di un nuovo pacchetto IP esterno con una nuova intestazione IP esterna. L'intero pacchetto originale, interno, viaggia attraverso un tunnel da un punto all'altro di una rete IP. da un punto all'altro di una rete IP; nessun router lungo il percorso è in grado di esaminare l'intestazione IP interna. Poiché il pacchetto originale è incapsulato, il nuovo pacchetto più grande può avere indirizzi di origine e di destinazione completamente diversi, il che aumenta la sicurezza. La modalità tunnel viene utilizzata quando una o entrambe le estremità di un'associazione di sicurezza sono un gateway di sicurezza, come un firewall o un router che implementa IPsec. Con la modalità tunnel, un certo numero di host su reti dietro firewall può effettuare comunicazioni sicure senza implementare IPsec. I pacchetti non protetti generati da tali host vengono inoltrati attraverso le reti esterne tramite SA in modalità tunnel impostati dal software IPsec nel firewall o nel router sicuro al confine del o router sicuro al confine della rete locale.

Ecco un esempio di come funziona IPsec in modalità tunnel.

L'host A su una rete genera un pacchetto IP con l'indirizzo di destinazione dell'host B su un'altra rete, simile a quello mostrato nella Figura 9.3. Questo pacchetto viene instradato dall'host di origine verso un firewall o un router sicuro a un firewall o a un router sicuro al confine della rete di A. Il firewall filtra tutti i pacchetti IP con l'indirizzo di destinazione dell'host B. Il firewall filtra tutti i in uscita per determinare la necessità di un'elaborazione IPsec. Se questo pacchetto da A a B richiede IPsec, il firewall esegue l'elaborazione IPsec e incapsula il pacchetto con un'intestazione IP esterna. pacchetto con un'intestazione IP esterna. L'indirizzo IP di origine di questo pacchetto IP esterno è questo firewall, mentre l'indirizzo di destinazione può essere un firewall che delimita la rete locale di B. rete locale di B. Il pacchetto viene ora instradato verso il firewall di B, con router intermedi che esaminano solo l'intestazione IP esterna. esaminano solo l'intestazione IP esterna. Al firewall di B, l'intestazione IP esterna viene eliminata e il pacchetto interno viene e il pacchetto interno viene consegnato a B. L'ESP in modalità tunnel cripta e, facoltativamente, autentica l'intero pacchetto IP interno, compresa l'intestazione IP interna.

Capitolo 8

Secure Interoperation

Si tratta della Composizione di sistemi sicuri, sistemi con attributi o politiche di sicurezza identici o compatibili che condividono dati dove c'è comunicazione quando oggetti e soggetti hanno un livello di sicurezza assegnato (sistemi multilivello).

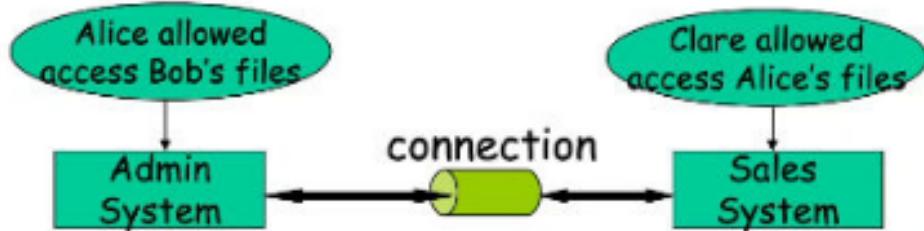


Figura 121: Esempio sistema non sicuro

I sistemi sono individualmente sicuri.

- È sicuro permettere la condivisione dei file tra i sistemi del personale e delle vendite?
 - Clare non è autorizzata ad accedere ai file di Bob, ma Clare però può accedere ai file di Bob tramite il sistema di amministratore clare - alice - bob.
 - Necessità di riconfigurare le connessioni/sistemi per chiudere questa via d'accesso tortuosa
 - Nel caso sopra bisognerebbe staccare la connessione visto che Clare accede a Bob tramite Alice, perché Alice può accedere ai file di Bob.

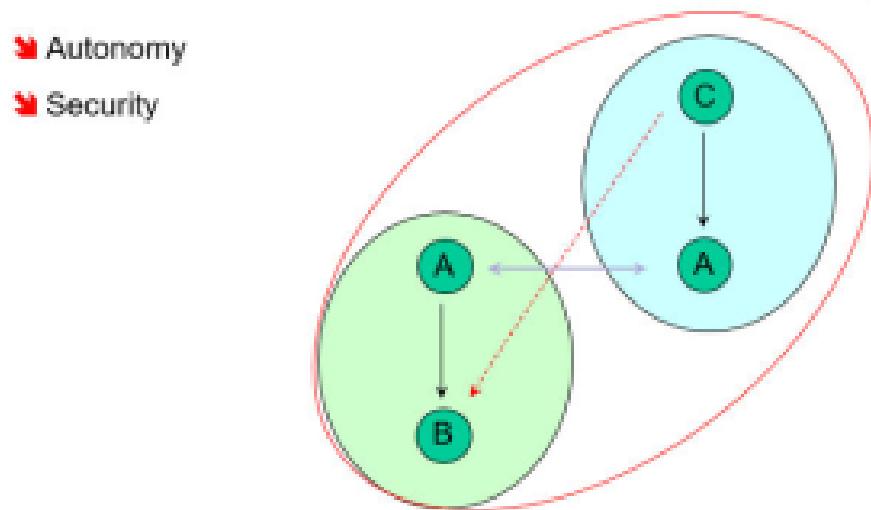


Figura 122: Interoperation Principles

Per risolvere il problema possiamo operare su:

- Interoperabilità
- Autonomia

Nel caso riportato sopra:

1. Riduzione Interoperabilità

Tolgo le relazioni I/O di 2 sistemi

2. Riduzione di Autonomia

Tolgo l'accesso di alice ai dati di bob

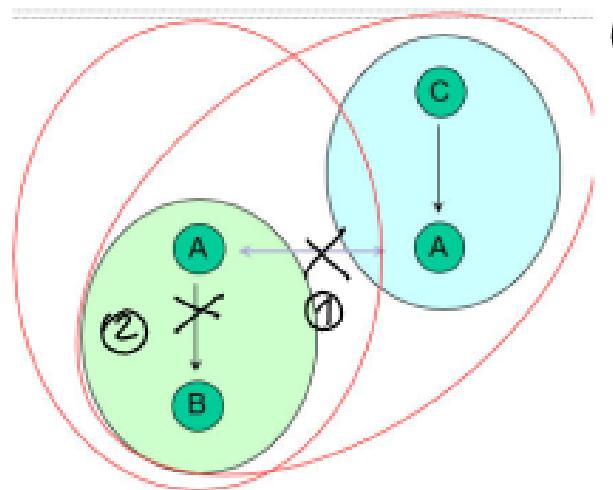


Figura 8.1: Sicurezza riducendo l'autonomia

Access configuration

Un insieme di vincoli tra entità (soggetti, oggetti) che specificano i permessi di accesso

- Variables $V = \{S, O\}$
- Domain $D = \{a, b, c\}$
- $CS1(a, b) = T$
- $CS1(a, c) = F$

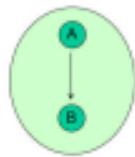


Figura 124: Esempio base

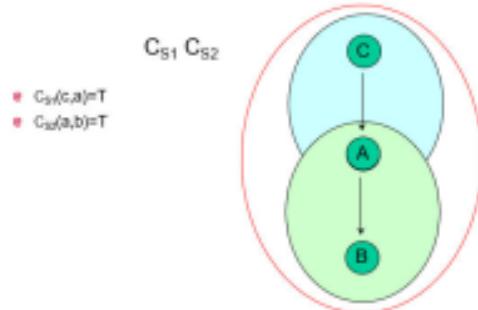


Figura 125: Combinazione di configurazioni

④ Represented as a semiring

◊ $S = \langle \text{PERM}, +, \cdot, ?, > \rangle$

$S_{\text{bool}} = \langle \{\text{F}, \text{T}\}, \cdot, \&, \text{F}, \text{T} \rangle$

$$a \xrightarrow{\text{F}} b$$

$$C_{S,0}(a,b) = \text{F}$$

$$a \xrightarrow{\text{T}} b$$

$$C_{S,0}(a,b) = \text{T}$$

$S_{rw} = \langle 2^{\langle r, w \rangle}, [A], \cdot, \{r, w\} \rangle$

$$a \xrightarrow{(w)} b$$

$$C_{S,0}(a,b) = \{w\}$$

Figura 126: Regole permessi: Verde permesso, rosso negato.

④ $S_{\text{bool}} = \langle \{\text{F}, \text{T}\}, \cdot, \&, \text{F}, \text{T} \rangle$

$$\text{④ } C_{S,0}(b,a) = \text{F}$$

$$\text{④ } C_{S,0}(c,b) = \text{F}$$

$$\text{④ } C_{S,0}(x,y) = \text{T}$$

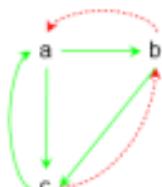


Figura 127: Modello/matrice d'accesso applicato

Per esempio posso non rappresentare tutti i diritti se al modello sono applicate regole di default deny o default permitted.

Se la lettera b che rappresenta l'utente fosse contenuta dentro un quadrato sarebbe transitiva, quindi permetterebbe la transitività. Transitività quadrata - Freccia verde.
Transitività tonda - Freccia rossa.

8.0.1 Acces Reconfiguration

E' necessaria per fondere 2 sistemi.

Esempio fusione di due aziende con policy diverse:

- se decidono di fondersi devono riconfigurare le policy in modo sicuro
- è sicuro se vengono ridotti i diritti che c'erano prima
- quindi una configurazione Cs' è sicura se gli accessi che do sono un sottoinsieme di Cs: Gli utenti non devono avere piu' accessi di quelli che avevano prima.

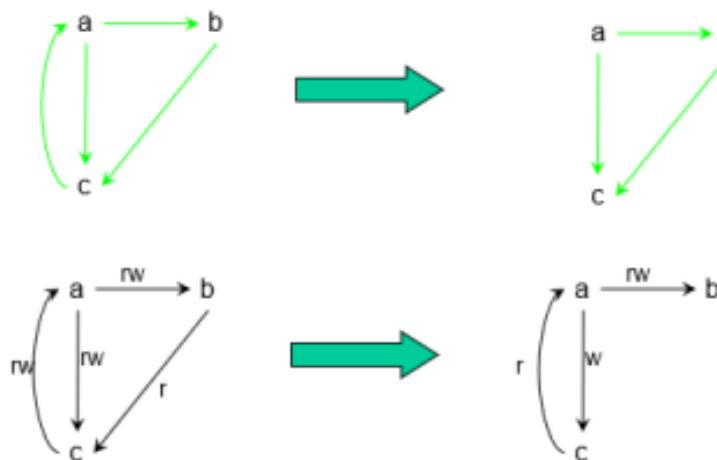


Figura 129: Esempio riconfigurazione accessi sicura

Supponiamo di avere un sistema Cs1 ed un sistema Cs3 che hanno policy e utenti diversi.

- Cs1 ha gli utenti a,b,c.
- Cs3 ha gli utenti a,c,d.

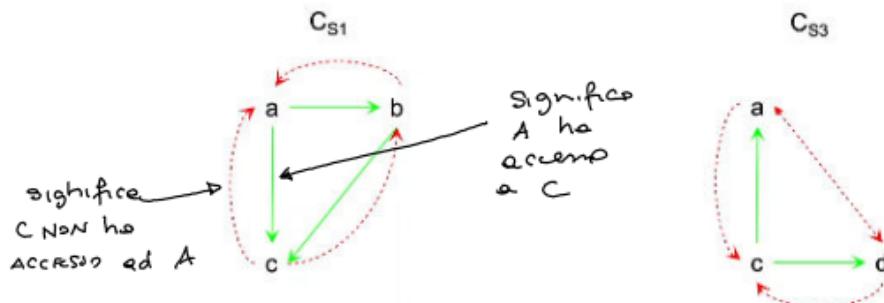


Figura 130: Rappresentazione ESPLICITA matrice degli accessi C_{S1} , C_{S3} .

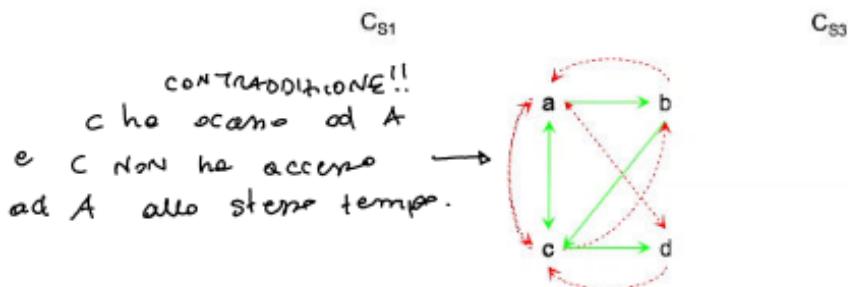


Figura 131: Esempio fusione di base C_{S1} , C_{S3} .

L'esempio di fusione sopra riportato non andrebbe bene, perché pieno di contraddizioni.

Quindi come faccio a sapere quali sono le cose da ridurre ?

Parto dalla fusione base rappresentato però solo quello che è vietato, così non ho neanche contraddizioni.

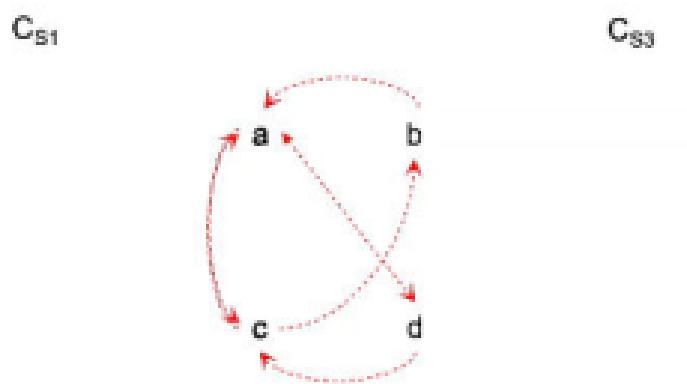


Figura 132: Fusione con sole negazioni

Quindi, proseguo aggiungendo i verdi (Indicano che ho l'accesso) per Cs1 e Cs3 dove non ci sono contraddizioni:

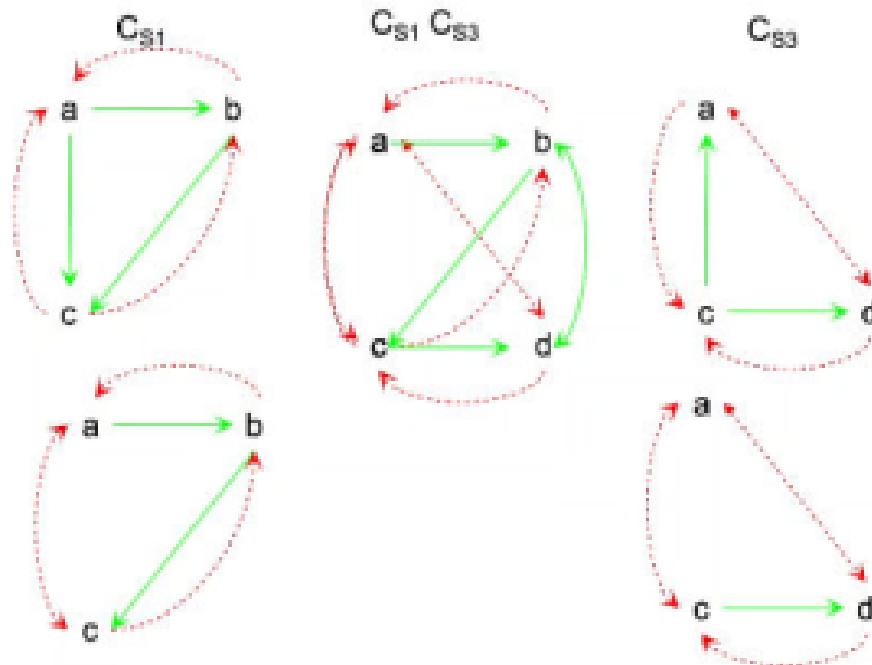


Figura 133: Fusione finale

Essendo C transitiva, A non può accedere a B perché A non accedere a C e C non può accedere a B. Stessa cosa per D e B. Aggiungo quindi più frecce rosse rispetto a sistemi non transitivi e non aggiungerò frecce verdi dove è presente la proprietà transitiva.



Figura 134: Esempio di partenza con transitività

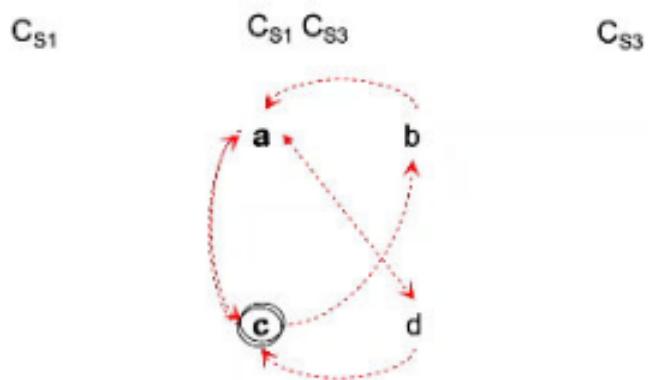


Figura 135: Passo 1 fusione con transitività

Proietto C_{S1} e C_{S3} su C_{S1} e C_{S3} , solo gli archi su cui non abbiamo contraddizioni

8.0.2 Acces Transitivity

Access Transitivity vs non-transitivity

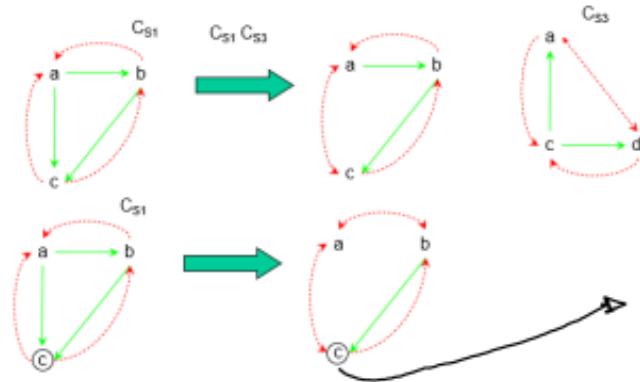


Figura 137: Modello transitivo vs Non transitivo

C è nodo
TRANSITIVO!
(rispetto alle
frecce rosse).
Cioè non

Le strategie principali per una massimale rinconfigurazione di sistema sono 2:

1. Rimuovere il numero minimo degli archi
2. Ridurre il numero degli archi mantenendo il massimo delle connessioni

Esempio riconfigurazione:

- Sistema 1: Arco verde A - B
- Sistema 2: Arco verde B - C
- B transitivo, quindi A - C
- Se però nella fusione mi ritrovo una policy con arco rosso tra A e C devo riconfigurare in modo diverso i sistemi. Avrei tre opzioni:
 - Togliere la freccia da A a B sul sistema 1
 - Togliere la freccia da B a C sul sistema 2
 - Elimino la transitività di B

Maintain maximum connections

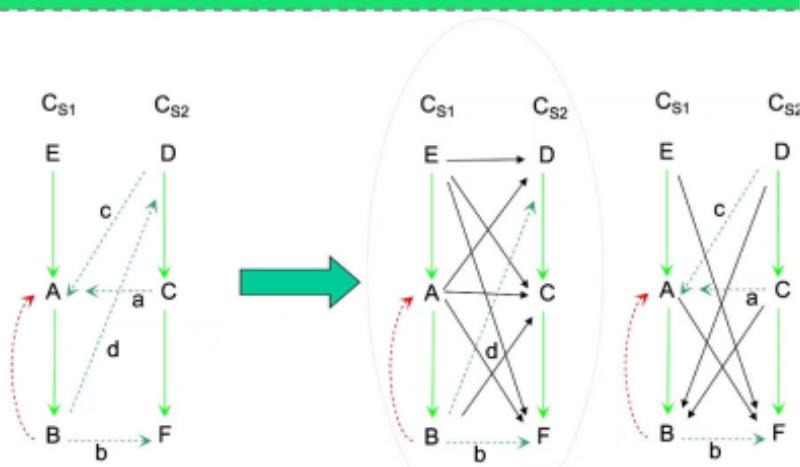


Figura 138: Esempio avanzato strategia maximum connections

Topologie speciali:

- Forma ad albero (sistema master e interoperabilità locale)
 - Tempo polinomiale
- Vita reale:
 - Client-server vs Peer-to-peer

8.0.3 Sistemi speciali: multilevel security

Considera il rischio di assurance nella composizione di sistemi sicuri multilivello valutati in base a criteri di valutazione della sicurezza.

- L'analisi della sicurezza di sistemi interoperanti e individualmente sicuri può essere fatta in tempo polinomiale.
- Data una configurazione di rete non sicura, allora riconfigurare le connessioni in modo ottimale (per minimizzare l'impatto sull'interoperabilità) è NP.

Esempio sistema MLS:

- A è in grado di gestire informazioni di 2 categorie: Secret e top secret
- N è in grado di gestire informazioni di 3 categorie: Secret, top secret e unclassified
- C è in grado di gestire informazioni di 2 categorie: Secret e unclassified

- Quale dovrebbe essere configurata in modo più stringente? La B perché ha più livelli di informazione (3) rispetto ad A e C e perchè un attaccante ha più livelli da attaccare. Avendo più livelli un sistema in genere più affidabile.

Configuring MLS Networks Channel Cascade Attacks

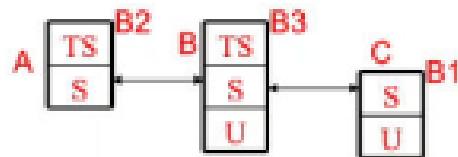


Figura 139: Configurazione rete multilivello

Nella figura TS non deve essere collegato a S visto che hanno un livello diverso di sicurezza. Le frecce in genere sono bidirezionali.

Gli evaluation criteria sono livelli di assurance che indicano quanto ci si può fidare di una macchina (testo di riferimento citato a lezione: orange book, sicurezza multilivelli. | red book per le connessioni tra questi livelli di sicurezza).

Ogni macchina ha un livello di assurance a seconda dei livelli confidenzialità delle informazioni che gestiscono.

Multilevel security modello Bell LaPadula:

- Livelli di sicurezza L definiscono la classificazione dei soggetti (processi) e degli oggetti.
 - Per esempio, Non classificato, Segreto, Top-Secret.
- Politica: reticolo di livelli di sicurezza (L, \leq)
 - $x \leq y$: le informazioni del livello x possono passare al livello y.
 - Non classificato $<$ Segreto $<$ Top-Secret

Per l'esempio riportato nella figura sopra si utilizza la logica di Bell LaPadula per la gestione e la comunicazione tra livelli.

Evaluation Criteria ["Orange" e "Red" Books]

- Sistemi MLS assicurati a diversi livelli di garanzia basati su criteri di valutazione.
 - (peggiore) $D < C1 < C2 < C3 < B1 < B2 < B3 < A1$ (migliore).
 - I sistemi valutati devono soddisfare i requisiti minimi di rischio.
 - I sistemi che memorizzano combinazioni di dati ad alto rischio hanno bisogno di alti livelli di garanzia

System Stores	Minimum Assurance
topsecret+unclassified	B3
topsecret+secret	B2
secret+unclassified	B1

Figura 140: Livelli di sicurezza esempio

Configuring MLS Networks Channel Cascade Attacks

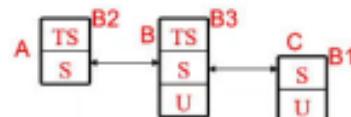


Figura 141: Struttura d'esempio per un attacco Cascade

Esempi attacchi Cascade

- Supponiamo che i singoli sistemi A, B, C sono sicuri.
- Ogni sistema valutato soddisfa i criteri.
- Tuttavia, la rete ha un rischio a cascata:
 - L'attaccante rompe il sistema A, copia i dati TS a S
 - Copia questi dati dal sistema A a B e C
 - Rompe il sistema C, copia i dati di S(TS) in U.

- L'assurance B3 è richiesta quando si proteggono TS e U, ma un attacco a cascata rompe i sistemi B2 e inferiori.

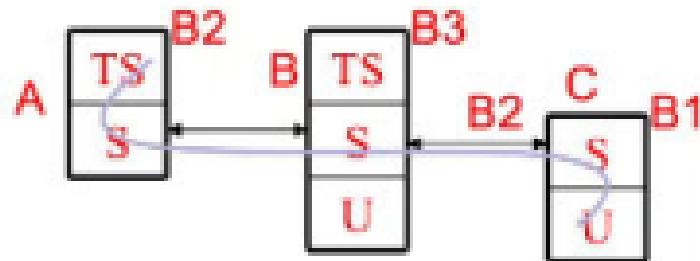
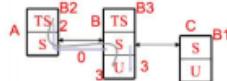


Figura 142: Flusso attacco Cascade

Usando questa strategia si evita lo sforzo di livello 3 per ‘rompere’ B3.

In questo caso ho uno sforzo di livello 2 andando inizialmente su B2, poi successivamente porto l'informazione da TS ad U sul sistema C che è anch'esso uno sforzo di livello 2. Con questo Cascade Path l'effort/sforzo è minore del costo dell'attacco perché costa 2 invece di 3, quindi si dice che uno sforzo 2 per un guadagno 3. Andando direttamente a rompere B3 invece mi sarebbe costato 3 e non ci sarebbe stato un attacco Cascade. In conclusione per passare da TS ad U uso B2 e B1 (passando per B3) e non B2 per poi andare a rompere B3, perché costerebbe di più.

Modeling MLS networks Strategy (using Constraints)



- Systems as *flow-constraints* between the levels of data that they store.
- Networks as *flow-constraints* that represent the channels that connect systems
- Soft constraint semi-ring as assurance levels
- Cascade Detection: finding cascades.

Figura 143: Ultimo esempio Cascade

Ex1: Cascade Free Path

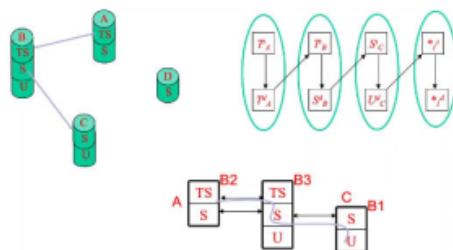


Figura 144: Esempio NON Cascade.

L'esempio non è cascade perché facendo l'operazione TS -> S su B3 la devo 'rompere' e di conseguenza fare uno sforzo 3.

Capitolo 9

Capitolo 27

9.1 Il modello Bell-LaPadula per la sicurezza informatica

9.1.1 Modelli di sicurezza informatica

In primo luogo, tutti i sistemi software complessi alla fine hanno rivelato difetti o bug che successivamente hanno dovuto essere corretti. Una buona discussione di questo può essere trovata nel classico *The Mythical Man-Month*. Secondo cui è straordinariamente difficile, se non impossibile, costruire un sistema hardware/software che non sia vulnerabile ad una varietà di attacchi alla sicurezza.

I problemi per fornire una forte sicurezza del computer hanno coinvolto sia il design che l'implementazione. È difficile, progettare qualsiasi hardware o software, ed essere sicuri che il progetto fornisca di fatto il livello di sicurezza che è stato previsto. Questa difficoltà si traduce in molte vulnerabilità di sicurezza non previste. Anche se il progetto è in un certo senso corretto, è difficile, se non impossibile, implementare il progetto senza errori o bug, fornendo un'altra serie di vulnerabilità. Questi problemi hanno portato al desiderio di sviluppare un metodo per dimostrare, logicamente o matematicamente, che un particolare progetto soddisfa un insieme dichiarato di requisiti di sicurezza e che l'implementazione di quel progetto è fedelmente conforme alle specifiche del progetto.

A tal fine, i ricercatori di sicurezza hanno cercato di sviluppare modelli formali di sicurezza del computer che possono essere utilizzati per verificare i progetti e le implementazioni di sicurezza.

Inizialmente, la ricerca in questo settore è stata finanziata dal Dipartimento della Difesa degli Stati Uniti e sono stati fatti notevoli progressi nello sviluppo di modelli e nella loro applicazione a sistemi prototipo. Quel finanziamento è notevolmente diminuito così come i

tentativi di costruire modelli formali di sistemi complessi. Il modello di sicurezza informatica più influente il modello Bell-LaPadula (BLP).

9.1.2 Descrizione Generale

Il modello BLP è stato sviluppato negli anni '70 come modello formale per il controllo degli accessi. Il modello si basava sul concetto di controllo dell'accesso. Nel modello, ad ogni soggetto e ad ogni oggetto viene assegnata una classe di sicurezza. Nella formulazione più semplice formulazione, le classi di sicurezza formano una rigida gerarchia e sono chiamate livelli di sicurezza.

Un esempio è lo schema di classificazione militare degli Stati Uniti:

top secret > secret > confidential > restricted > unclassified

È possibile anche aggiungere un insieme di compartimenti, o categorie, ad ogni livello di sicurezza, così che un soggetto deve essere assegnato sia al livello appropriato che al compartimento per accedere ad un oggetto. Questo concetto è ugualmente applicabile in altre aree, dove le informazioni possono essere organizzate in livelli vuoti e scompartimenti pieni, e agli utenti possono essere concesse autorizzazioni per accedere a certi compartimenti di dati. Per esempio, il livello più alto di sicurezza potrebbe essere per i documenti ed i dati strategici di pianificazione aziendale, accessibili solo ai funzionari aziendali e al loro staff.

Questo suggerisce uno schema di classificazione:

strategic > sensitive > confidential > public

- Un soggetto ha un nulla osta di sicurezza di un determinato livello
- Un oggetto deve avere una classificazione di sicurezza di un determinato livello.

Le classi di sicurezza controllano il modo con cui un soggetto può accedere ad un oggetto. Il modello ha definito quattro modalità di accesso.

I modi sono i seguenti:

1. **Lettura:** Al soggetto è permesso solo l'accesso in lettura all'oggetto.
2. **Append:** Al soggetto è permesso solo l'accesso in scrittura all'oggetto.
3. **Scrittura:** Il soggetto è autorizzato ad accedere sia in lettura che in scrittura all'oggetto.
4. **Esecuzione:** Il soggetto non è autorizzato né a leggere né a scrivere sull'oggetto ma può invocare l'oggetto per l'esecuzione.

Quando vengono definite più categorie o livelli di dati, il requisito viene definito sicurezza multilivello (MLS). La dichiarazione generale del requisito per la sicurezza multilivello incentrata sulla riservatezza è che un soggetto ad un livello alto non può trasmettere informazioni ad un soggetto ad un livello inferiore a meno che quel flusso rifletta accuratamente la volontà di un utente autorizzato come rivelato da una declassificazione autorizzata. Ai fini dell'implementazione, questo requisito è in due parti ed è dichiarato semplicemente.

Un sistema sicuro multilivello per la riservatezza deve far rispettare quanto segue:

- **Nessuna lettura:** Un soggetto può leggere solo un oggetto di livello di sicurezza inferiore o uguale. Questo è indicato in letteratura come la proprietà di sicurezza semplice (ss-property).
- **Nessuna scrittura:** Un soggetto può solo scrivere in un oggetto di livello di sicurezza maggiore o uguale. Questo è indicato in letteratura come la proprietà1 (pronunciato star property).

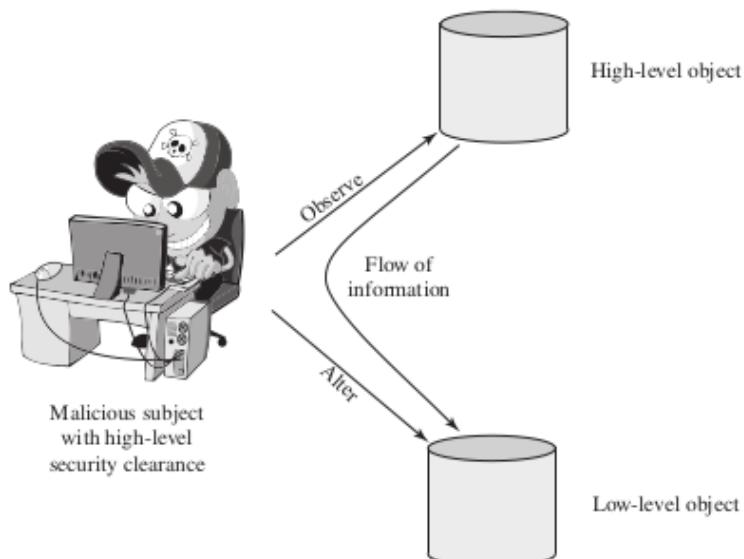


Figure 27.1 Information Flow Showing the Need for the *-Property

Qui, un soggetto malintenzionato passa informazioni classificate mettendole in un contenitore di informazioni etichettato con una classificazione di sicurezza inferiore a quella delle informazioni stesse. Questo permetterà un successivo accesso in lettura a queste informazioni da parte di un soggetto con un livello di autorizzazione inferiore. Queste due proprietà forniscono la forma di riservatezza di ciò che è noto come controllo obbligatorio dell'accesso (MAC). Sotto il MAC, non è permesso alcun accesso che non soddisfi queste due proprietà. Inoltre, il modello BLP prevede un controllo di accesso discrezionale (DAC).

- **DS-proprietà**

Un individuo (o ruolo) può concedere a un altro individuo (o ruolo) l'accesso a un documento in base alla discrezione del proprietario, limitato dalle regole MAC regole.

Così, un soggetto può esercitare solo gli accessi per i quali ha la necessaria autorizzazione e che soddisfano le regole del MAC.

9.1.3 Descrizione formale del modello

Il modello si basa sul concetto di uno stato attuale del sistema. Lo stato è descritto dalla 4-tupla (b, M, f, H) , definita come segue:

- **Insieme di accesso corrente b:**

Questo è un insieme di triple della forma (soggetto, oggetto, accesso-modo). Una tripla (s, o, a) significa che il soggetto s ha accesso corrente a o nel modo di accesso a . Questo non significa semplicemente che s ha il diritto di accesso ad a o ad o . La tripla significa che s sta attualmente esercitando tripla significa che s sta attualmente esercitando quel diritto di accesso; cioè, s sta attualmente accedendo ad a o in modalità a .

- **Matrice di accesso M:**

L'elemento della matrice M_{ij} registra le modalità di accesso in cui il soggetto S_i è autorizzato di accedere all'oggetto O_j .

- **Funzione di livello F:**

Questa funzione assegna un livello di sicurezza ad ogni soggetto e oggetto. Consiste di tre mappature: $f_o(O_j)$ è il livello di classificazione dell'oggetto O_j e $f_s(S_i)$ che è il nulla osta di sicurezza del soggetto S_i . $f_c(S_i)$ è il livello di sicurezza attuale del soggetto S_i . Il nulla osta di sicurezza di un soggetto è il massimo livello di sicurezza del soggetto. Il soggetto può operare a questo livello o a un livello inferiore. Così, un utente può accedere al sistema ad un livello inferiore al nulla osta di sicurezza dell'utente. Questo è particolarmente utile in un sistema di controllo dell'accesso basato sui ruoli.

- **Gerarchia H:**

Si tratta di un albero con radici dirette i cui nodi corrispondono agli oggetti del sistema. Il modello richiede che il livello di sicurezza di un oggetto domini il livello di sicurezza del suo genitore. Per la nostra discussione, possiamo equiparare questo, alla condizione che il livello di sicurezza di un oggetto deve essere maggiore o uguale al suo genitore.

Possiamo ora definire le tre proprietà di BLP in modo più formale. Per ogni soggetto S_i e ogni oggetto O_j , i requisiti possono essere dichiarati come segue:

- **ss-property:** Tutte le triple della forma (S_i, O_j, read) nell'insieme corrente di accesso b hanno la proprietà $fc(S_i) \geq fo(O_j)$.
- **property:** Tutte le triple della forma $(S_i, O_j, \text{append})$ nell'insieme di accesso corrente b ha la proprietà $fc(S_i) \leq fo(O_j)$. Tutte le triple della forma (S_i, O_j, write) nell'attuale set di accesso b hanno la proprietà $fc(S_i) = fo(O_j)$.
- **ds-property:** Se (S_i, O_j, Ax) è un'accesso corrente (in b), la modalità di accesso Ax è registrata in (S_i, O_j) elementi di M . Questo da (S_i, O_j, Ax) ed implica che $Ax \in M$

S_i, O_j

Queste tre proprietà possono essere utilizzate per definire un sistema sicuro per la riservatezza.

In sostanza, un sistema sicuro è caratterizzato da quanto segue:

1. Lo stato di sicurezza attuale del sistema (b, M, f, H) è sicuro se e solo se ogni elemento di b soddisfa le tre proprietà.
2. Lo stato di sicurezza del sistema viene cambiato da qualsiasi operazione che causa un cambiamento uno qualsiasi dei quattro componenti del sistema, (b, M, f, H) .
3. Un sistema sicuro rimane sicuro finché qualsiasi cambiamento di stato non viola le tre proprietà.

Questi tre punti possono essere espressi come teoremi usando il modello formale. Inoltre, dato un progetto o un'implementazione attuale, è teoricamente possibile dimostrare che il sistema è sicuro provando che ogni azione che influenza sullo stato del sistema soddisfa le tre proprietà. In pratica, per un sistema complesso, tale prova non è mai stata completamente sviluppata. Tuttavia, come menzionato prima, la dichiarazione formale dichiarazione formale dei requisiti può portare ad una progettazione e implementazione più sicura.

9.1.4 Operazioni Astratte

Il modello BLP include un insieme di regole basate su operazioni astratte che cambiano lo stato del sistema. Le regole sono le seguenti:

1. Ottenere l'accesso

Aggiungere una tripla (soggetto, oggetto, modalità di accesso) al set di accesso corrente
b. Utilizzato da un soggetto per avviare l'accesso a un oggetto nel modo richiesto

2. Rilasciare l'accesso

Rimuove una tripla (soggetto, oggetto, modo di accesso) dal set di accesso corrente b.
Usato per rilasciare un accesso precedentemente iniziato.

3. Cambiare livello dell'oggetto

Cambia il valore di $fo(Oj)$ per qualche oggetto Oj . Usato da un soggetto per modificare il livello di sicurezza di un oggetto

4. Cambiare livello attuale

Cambia il valore di $fc(Si)$ per qualche soggetto Si . Usato da un soggetto per alterare il livello di sicurezza di un oggetto.

5. Dare il permesso di accesso

Aggiungere una modalità di accesso a qualche voce della permissione M . Usato da un soggetto per concedere un modo di accesso su un oggetto specificato a un altro soggetto.

6. Rescindere il permesso di accesso

Cancella un modo di accesso da qualche voce di M . Usato da un soggetto per revocare un accesso precedentemente concesso.

7. Creare oggetto

Attacca un oggetto alla struttura ad albero corrente H come foglia. Usato per creare un nuovo oggetto o attivare un oggetto che è stato precedentemente definito ma è inattivo perché non è stato inserito in H .

8. Cancellare un gruppo di oggetti

Stacca da H un oggetto e tutti gli altri oggetti sotto di esso nella gerarchia. Questo rende il gruppo di oggetti inattivo. Questa operazione può anche modificare l'attuale set di accesso b perché tutti gli accessi all'oggetto vengono rilasciati.

Le regole 1 e 2 alterano l'accesso corrente. Le regole 3 e 4 alterano le funzioni di livello. Le regole 5 e 6 alterano il permesso di accesso e le regole 7 e 8 alterano la gerarchia. Ogni regola è governata dall'applicazione delle tre proprietà. Per esempio, per ottenere l'accesso per una lettura, dobbiamo avere $fc(Si) \geq fo(Oj)$ e $Ax \in M[Si, Oj]$.

9.1.5 Esempio di utilizzo Bella-Pabula

Questo esempio illustra il funzionamento del modello BLP ed evidenzia anche un problema pratico problema pratico che deve essere affrontato. Assumiamo un sistema di controllo degli accessi basato sui ruoli.

Carla e Dirk sono utenti del sistema. Carla è una studentessa (s) nel corso c1. Dirk è un insegnante (t) nel corso c1, ma può anche accedere al sistema come studente; così, due ruoli sono assegnati a Dirk:

$$\begin{aligned} \text{Carla : } & (c1 - s) \\ \text{Dirk : } & (c1 - t), (c1 - s) \end{aligned}$$

Al ruolo di studente è assegnato un nulla osta di sicurezza inferiore e al ruolo di insegnante un autorizzazione di sicurezza più alta. Vediamo alcune possibili azioni:

1. **Dirk crea un nuovo file f1 come c1-t; Carla crea il file f2 come c1-s (vedi Figura 27.2a).**

Carla può leggere e scrivere su f2, ma non può leggere f1, perché è a un livello di classificazione più alto (livello insegnante).

2. **Nel ruolo c1-t, Dirk può leggere e scrivere f1 e può leggere f2 se Carla concede l'accesso a f2.**

Tuttavia, in questo ruolo, Dirk non può scrivere f2 a causa della proprietà. Né Dirk né un cavallo di Troia per suo conto possono declassare i dati dal livello dell'insegnante al livello dello studente.

Solo se Dirk si collega come studente può creare un file c1-s o scrivere su un file c1-s esistente, come f2. Nel ruolo di studente, Dirk può anche leggere f2.

3. **Dirk legge f2 e vuole creare un nuovo file con commenti a Carla come feedback.**

Dirk deve firmare nel ruolo studente c1-s per creare f3 in modo che sia accessibile da Carla (vedi Figura 27.2b). In un ruolo di insegnante, Dirk non può creare un file a livello di classificazione studente.

4. **Dirk crea un esame basato su un file modello esistente memorizzato a livello c1-t.**

Dirk deve accedere come c1-t per leggere il modello, e anche il file che crea (f4) deve essere a livello dell'insegnante (vedi Figura 27.2c).

5. **Dirk vuole che Carla faccia l'esame, e quindi deve fornirle un accesso in lettura.**

Tuttavia, tale accesso violerebbe la proprietà ss. Dirk deve declassare la classificazione di f4 da c1-t a c1-s.

Dirk non può farlo nel ruolo c1-t perché questo violerebbe la proprietà. Pertanto, un amministratore di sicurezza (possibilmente Dirk in questo ruolo) deve avere l'autorità di downgrade e deve essere in grado di eseguire il downgrade al di fuori del modello BLP. La linea tratteggiata nella Figura 27.2d che collega f4 con c1-s-read indica che questa connessione non è stata generata dalle regole predefinite di BLP ma da un'operazione di sistema.

6. Carla scrive le risposte all'esame in un file f5.

Crea il file a livello c1-t in modo che solo Dirk possa leggere il file.

Questo è un esempio di scrittura, che non è vietato dalle regole BLP. Carla può ancora vedere le sue risposte alla sua stazione di lavoro, ma non può accedere a f5 per leggere.

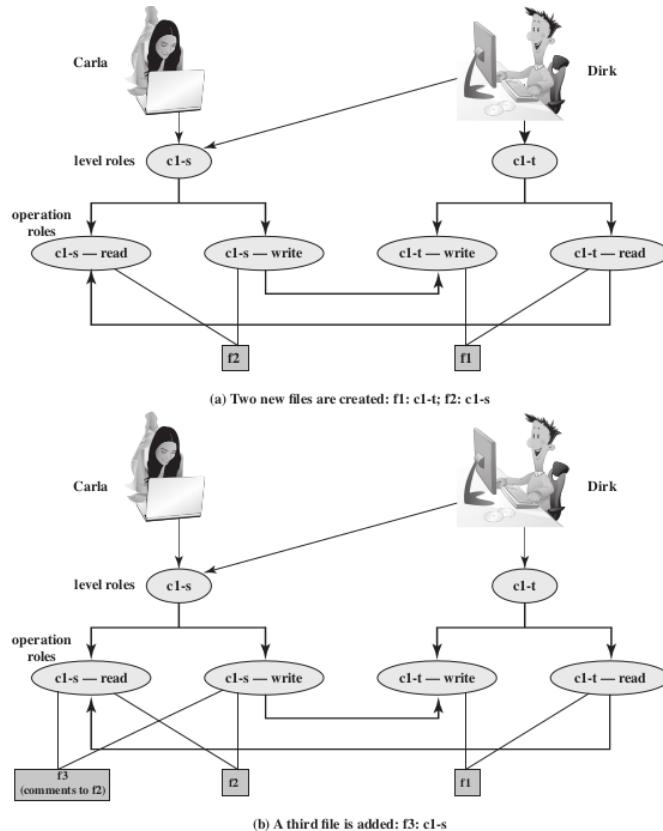
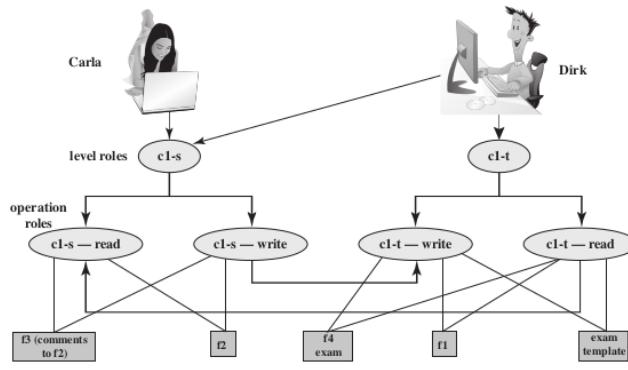
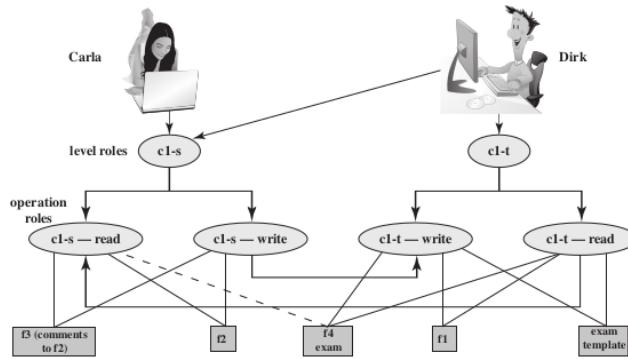


Figure 27.2 Example of Use of BLP Concepts

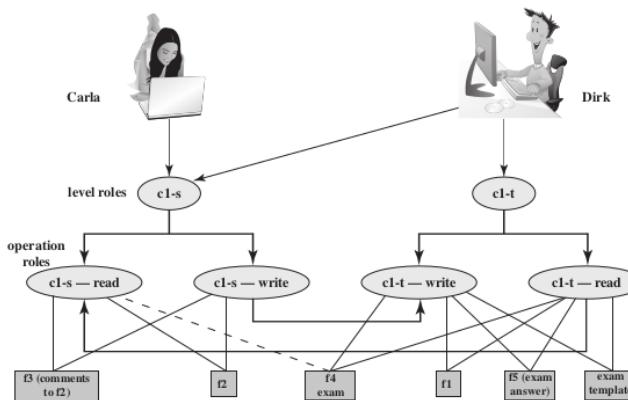


(c) An exam is created based on an existing template: f4: c1-t



(d) Carla, as student, is permitted access to the exam: f4: c1-s

Figure 27.2 Example of Use of BLP Concepts



(e) The answers given by Carla are only accessible for the teacher: f5: c1-t

In primo luogo, come notato al punto 4, il modello BLP non ha alcuna disposizione per gestire il "downgrade" di oggetti, anche se i requisiti per la sicurezza multilivello riconoscono che tale un flusso di informazioni da un livello superiore a uno inferiore può essere richiesto, purché rifletta la volontà di un utente autorizzato. Quindi, qualsiasi implementazione pratica di un sistema multilivello deve supportare tale processo in modo controllato e monitorato. Collegato a questo c'è un'altra preoccupazione. Un soggetto

vincolato dal modello BLP può solo "editare" (leggere e scrivere) un file ad un livello di sicurezza mentre visualizza anche file allo stesso livello o a livelli inferiori. Se il nuovo documento consolida informazioni da una gamma di fonti e livelli, alcune di quelle informazioni sono ora classificate ad un livello rispetto a quello originale. Questo è noto come classification creep ed è un problema ben noto preoccupazione quando si gestiscono informazioni multilivello. Anche in questo caso, è necessario un processo di declassamento delle informazioni è necessario per ripristinare livelli di classificazione ragionevoli.

9.1.6 Esempio di implementazione Multics

Un'implementazione di MLS sul sistema operativo Multics.

Iniziamo con una breve descrizione degli aspetti rilevanti di Multics. Multics è un sistema operativo a tempo condiviso che fu sviluppato da un gruppo del MIT noto come Progetto MAC (computer ad accesso multiplo) negli anni '60. Multics era non solo anni, ma decenni in anticipo sui tempi. Anche a metà degli anni '80, quasi 20 anni dopo essere diventato operativo, Multics aveva caratteristiche di sicurezza superiori e una maggiore sofisticazione nell'interfaccia utente e in altre aree rispetto ad altri sistemi operativi per mainframe contemporanei. Sia la gestione della memoria che il file system in Multics sono basati sul concetto di segmenti. La memoria virtuale è segmentata. Ogni file nel file system è definito come un segmento. Così, il sistema operativo utilizza lo stesso meccanismo per caricare un segmento di dati dalla memoria virtuale nella memoria principale, e per caricare un file dalla memoria virtuale nella memoria principale. I segmenti sono organizzati gerarchicamente, da una directory principale fino ai singoli segmenti.

Multics gestisce lo spazio di indirizzamento virtuale per mezzo di un segmento descrittore, che è associato ad un processo e che ha una voce per ogni segmento nella memoria virtuale accessibile da questo processo. Il registro base del segmento del descrittore punta all'inizio del segmento descrittore per il processo attualmente in esecuzione. Per MLS, sono necessarie due caratteristiche aggiuntive. Una tabella a livello di processo include ed una voce per ogni processo attivo, e la voce indica l'autorizzazione di sicurezza del processo. Associato ad ogni segmento c'è un livello di sicurezza, che è memorizzato nel segmento del segmento in questione.

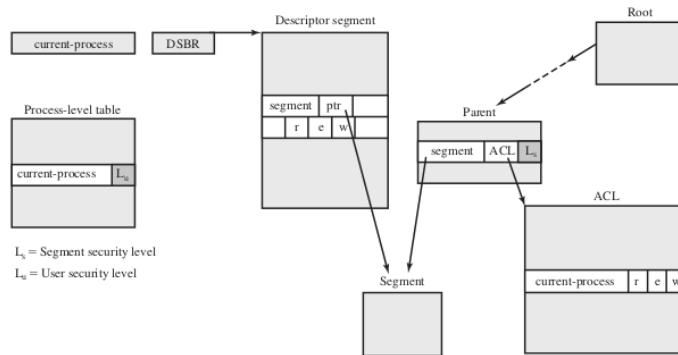


Figure 27.3 Multics Data Structures for MLS

Corrispondente allo stato di sicurezza del modello BLP (b, M, f, H) è un insieme di strutture dati Multics (vedi Figura 27.3).

La corrispondenza è la seguente:

- **b**: Segmento descrittore parola.

Il segmento descrittore identifica il soggetto (processo). Il puntatore di segmento nella parola descrittore di segmento identifica l'oggetto (segmento dati). I tre bit di controllo dell'accesso nel segmento descrittore di segmento identificano il modo di accesso.

- **M:** elenco di controllo dell'accesso
- **f:** Informazioni nel segmento directory e nella tabella a livello di processo.
- **H:** Struttura gerarchica del segmento.

Con queste strutture di dati, Multics può imporre il controllo di accesso discrezionale e obbligatorio. Quando un processo tenta un accesso ad un segmento, deve avere il permesso di accesso desiderato come specificato dalla lista di controllo degli accessi. Inoltre, la sua autorizzazione di sicurezza viene confrontata con la classificazione di sicurezza del segmento a cui accedere per determinare se la regola di sicurezza semplice e la regola di sicurezza sono soddisfatte.

9.1.7 Limitazioni modello BLP

Mentre il modello BLP potrebbe, in teoria, porre le basi per un calcolo sicuro all'interno di un ambiente di amministrazione singola, ci sono alcune importanti limitazioni alla sua usabilità e difficoltà di implementazione.

1. L'incompatibilità di riservatezza e integrità all'interno di un singolo sistema MLS.

In termini generali, MLS può funzionare sia per i poteri che per i segreti, ma non prontamente per entrambi. Questa esclusione reciproca esclude alcune interessanti tecnologie centrate sulla potenza e sull'integrità di essere usate efficacemente in ambienti MLS in stile BLP.

2. Limitazione all'usabilità è il cosiddetto problema del cospiratore cooperante.

In presenza di canali nascosti. In presenza di risorse condivise, la proprietà * può diventare inapplicabile. Questo è un problema specialmente nella presenza contenuto attivo che è prevalente nell'attuale elaborazione di testi e in altri formati di documenti. Un documento maligno potrebbe contenere un soggetto che quando eseguito trasmette documenti classificati usando canali segreti a risorse condivise. In essenza, il modello BLP si rompe efficacemente quando i dati eseguibili (non fidati) a bassa classificazione dati eseguibili a bassa classificazione possono essere eseguiti da un soggetto ad alta autorizzazione (fidato).

9.2 Altri modelli per la sicurezza informatica

È importante notare che i modelli descritti in questo capitolo si concentrano sulla riservatezza o sull'integrità, con l'eccezione del Chinese Wall Model. L'incompatibilità delle preoccupazioni di riservatezza e integrità è riconosciuta come una grande limitazione all'usabilità di MLS in generale, e a MLS focalizzati sulla riservatezza in particolare.

9.2.1 Modello di integrità Biba

Il modello BLP si occupa della riservatezza ed è preoccupato della divulgazione non autorizzata delle informazioni. Il modello Biba si occupa dell'integrità e si occupa della modifica non autorizzata dei dati. Il modello Biba è destinato a trattare il caso in cui ci sono dati che devono essere visibili agli utenti a più o a tutti i livelli di sicurezza, ma devono essere modificati solo in modi controllati da agenti autorizzati.

Gli elementi di base del modello Biba hanno la stessa struttura del modello BLP. Come con BLP, il modello Biba si occupa di soggetti e oggetti. Ogni soggetto e oggetto è assegnato un livello di integrità, indicato come $I(S)$ e $I(O)$ per il soggetto S e l'oggetto O , rispettivamente. Si può usare una semplice classificazione gerarchica, in cui c'è un ordine rigoroso dei livelli dal più basso al più alto.

Come nel modello BLP, è anche possibile aggiungere un insieme di compartimenti allo schema di classificazione.

Il modello considera le seguenti modalità di accesso:

1. **Modificare:** Scrivere o aggiornare le informazioni in un oggetto
2. **Osservare:** Leggere le informazioni di un oggetto
3. **Eseguire:** Eseguire un oggetto
4. **Invocare:** Comunicazione da un oggetto all'altro

I primi tre modi sono analoghi ai modi di accesso BLP. Il modo invoke è nuovo. Biba propone poi una serie di politiche alternative che possono essere imposte a questo modello.

La più rilevante è la politica di integrità rigorosa, basata sulle seguenti regole:

- **Integrità semplice:** Un soggetto può modificare un oggetto solo se il livello di integrità del soggetto domina il livello di integrità dell'oggetto: $I(S) >= I(O)$.
- **Confinamento dell'integrità:** Un soggetto può leggere un oggetto solo se il livello di integrità del soggetto è dominato dal livello di integrità dell'oggetto: $I(S) <= I(O)$.
- **Proprietà di invocazione:** Un soggetto può invocare un altro soggetto solo se il livello di integrità del primo soggetto domina il livello di integrità del secondo soggetto: $I(S1) >= I(S2)$.

Le prime due regole sono analoghe a quelle del modello BLP ma riguardano dell'integrità e invertono il significato di lettura e scrittura. La semplice regola di integrità è la restrizione logica di scrittura che impedisce la contaminazione dei dati ad alta integrità. Un processo a bassa integrità può leggere dati a bassa integrità ma gli viene impedito di contaminare un file ad alta integrità con quei dati grazie alla semplice regola di integrità. Se solo questa regola è in vigore, un processo ad alta integrità potrebbe plausibilmente copiare dati a bassa integrità in un file ad alta integrità in un file ad alta integrità. Normalmente, ci si aspetterebbe che un processo ad alta integrità non contamini un file ad alta integrità, ma un errore nel codice del processo o un cavallo di Troia potrebbe risultare in tale contaminazione; da qui la necessità della regola di confinamento dell'integrità.

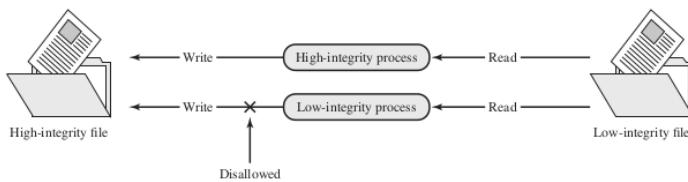


Figure 27.4 Contamination with Simple Integrity Controls

9.2.2 Modello di integrità Clark-Wilson

Un modello di integrità più elaborato e forse più pratico è stato proposto da Clark e Wilson. Il modello di integrità Clark-Wilson (CWM) è rivolto ad applicazioni commerciali piuttosto che militari e modella da vicino le reali operazioni commerciali. Il modello si basa su due concetti che sono tradizionalmente usati per applicare politiche di sicurezza commerciali:

- **Transazioni ben formate:** Un utente non dovrebbe manipolare i dati arbitrariamente, ma solo in modi limitati che preservano o assicurano l'integrità dei dati.
- **Separazione dei compiti tra gli utenti:** Ogni persona autorizzata a creare o certificare una transazione ben formata non può essere autorizzata ad eseguirla (almeno contro dati di produzione). Il modello impone controlli di integrità sui dati e sulle transazioni che manipolano i dati.

I componenti principali del modello sono i seguenti:

- **Elementi di dati vincolati (CDI):** Soggetti a severi controlli di integrità
- **Elementi di dati non vincolati (UDI):** Elementi di dati non controllati. Un esempio è un semplice file di testo
- **Procedure di verifica dell'integrità (IVP):** Destinate ad assicurare che tutte le CDI sono conformi a qualche modello di integrità e coerenza specifico dell'applicazione
- **Procedure di trasformazione (TP):** Transazioni di sistema che cambiano l'insieme delle CDI da uno stato coerente ad un altro. Il CWM fa rispettare l'integrità per

mezzo di regole di certificazione e applicazione sui TP. Le regole di certificazione sono restrizioni di politica di sicurezza sul comportamento di IVP e dei TP.

Le regole di applicazione sono meccanismi di sicurezza integrati nel sistema che raggiungono gli obiettivi delle regole di certificazione.

Le regole sono le seguenti:

C1: Tutti gli IVP devono garantire adeguatamente che tutti i CDI siano in uno stato valido nel momento in cui l'IVP viene eseguito.

C2: Tutti i TP devono essere certificati per essere validi. Cioè, devono portare un CDI ad uno stato finale valido, dato che è in uno stato valido per cominciare. Per ogni TP, ogni insieme di CDI che può manipolare, il responsabile della sicurezza deve specificare una relazione che definisce tale esecuzione. Una relazione è quindi della forma $(TP_i, (CDI_a, CDI_b, CDI_c \dots))$, dove la lista dei CDI definisce un particolare insieme di argomenti per i quali il TP è stato certificato.

E1: Il sistema deve mantenere la lista di relazioni specificata nella regola C2 e deve assicurare che l'unica manipolazione di qualsiasi CDI sia da parte di un TP, dove il TP sta operando sul CDI come specificato in qualche relazione.

E2: Il sistema deve mantenere una lista di relazioni della forma $(UserID, TP_i, (CDI_a, CDI_b, CDI_c, \dots))$, che mette in relazione un utente, un TP e gli oggetti dati che il TP può referenziare per conto di quell'utente. Deve assicurare che solo esecuzioni descritte in una delle relazioni.

C3: L'elenco delle relazioni in E2 deve essere certificato per soddisfare il requisito di separazione dei compiti.

E3: Il sistema deve autenticare l'identità di ogni utente che tenta di eseguire un TP.

C4: Tutti i TP devono essere certificati per scrivere in un CDI di sola appendice (il log) tutte le informazioni necessarie per permettere di ricostruire la natura dell'operazione.

C5: Ogni TP che prende un UDI come valore di ingresso deve essere certificato per eseguire solo trasformazioni valide, oppure nessuna trasformazione, per ogni possibile valore dell'UDI. La trasformazione dovrebbe prendere l'input da un UDI, o l'UDI viene rifiutata. Tipicamente, questo è un programma di modifica.

E4: Solo l'agente autorizzato a certificare entità può cambiare la lista di tali entità associate ad altre entità: in particolare, la lista dei TP associati con un CDI e la lista degli utenti associati a un TP. Un agente che può certificare un'entità non può avere alcun diritto di esecuzione rispetto a tale entità.

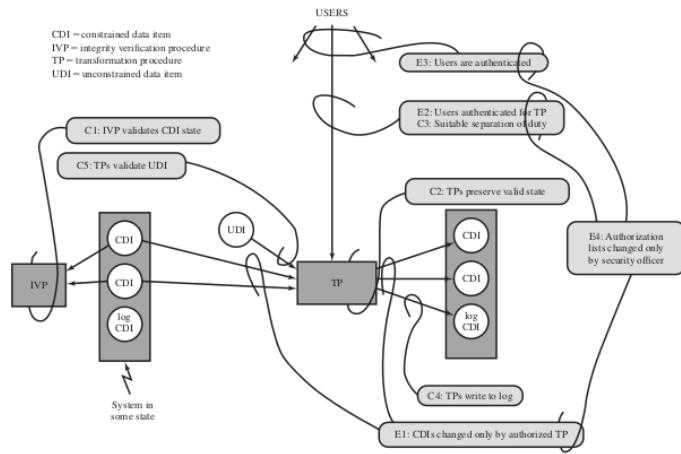


Figure 27.5 Summary of Clark-Wilson System Integrity Rules

9.2.3 Modello Muraglia Cinese

Il Chinese Wall Model (CWM) ha un approccio abbastanza diverso per specificare integrità e riservatezza rispetto a qualsiasi approccio che abbiamo esaminato finora. Il modello è stato sviluppato per applicazioni commerciali in cui possono sorgere conflitti di interesse. Il modello fa uso di concetti di accesso sia discrezionali che obbligatori. L'idea principale dietro il CWM è un concetto che è comune nelle professioni legali, che è quello di usare una cosiddetta muraglia cinese per prevenire un conflitto di interessi. Un esempio dal mondo finanziario è quello di un analista di mercato che lavora per un'istituzione finanziaria che fornisce servizi aziendali. Un analista non può essere autorizzato a fornire consigli a una società quando l'analista ha informazioni riservate (conoscenza privilegiata) sui piani o lo stato di un concorrente. Tuttavia l'analista è libero di consigliare più società che non sono in concorrenza tra loro e di attingere alle informazioni di mercato che sono aperte al pubblico.

Gli elementi del modello sono i seguenti:

- **Soggetti:** Entità attive che potrebbero voler accedere a oggetti protetti. include utenti e processi
- **Informazioni:** Informazioni aziendali organizzate in una gerarchia con tre livelli
 - **Oggetti:** Singoli elementi di informazione, ciascuno riguardante una singola società
 - **Set di dati (DS):** Tutti gli oggetti che riguardano la stessa società
 - **Classe di conflitto di interessi (CI):** Tutti i set di dati le cui società sono in concorrenza
- **Regole di accesso:** Regole per l'accesso in lettura e scrittura

A differenza dei modelli che abbiamo studiato finora, il CWM non assegna livelli di sicurezza a soggetti e oggetti e quindi non è un vero modello sicuro multi-livello. Invece, la storia del precedente accesso di un soggetto determina il controllo dell'accesso. La base della politica della muraglia cinese è che i soggetti possono accedere solo alle informazioni che non è ritenuto in conflitto con qualsiasi altra informazione che già possiedono. Una volta che un soggetto accede alle informazioni di un set di dati, una muraglia è impostata per proteggere le informazioni in altri insiemi di dati nello stesso CI. Il soggetto può accedere alle informazioni su un lato del muro ma non dall'altro lato. Inoltre, le informazioni in altri CI non sono inizialmente. Inoltre, le informazioni in altri IC non sono inizialmente considerate su un lato o l'altro del muro, ma all'aperto. Quando lo stesso soggetto effettua ulteriori accessi in altri IC, la forma del muro cambia per mantenere la protezione desiderata. Inoltre, ogni soggetto è controllato dal proprio pareti per i diversi soggetti che sono diverse.

Per far rispettare la politica della muraglia cinese, sono necessarie due regole. Per indicare la similarità con le due regole BLP, gli autori hanno dato loro gli stessi nomi.

La prima regola è la **semplice regola di sicurezza**

Regola di sicurezza semplice:

- O si trova nello stesso DS di un oggetto a cui S ha già avuto accesso
- O appartiene a un CI da cui S non ha ancora avuto accesso a nessuna informazione.

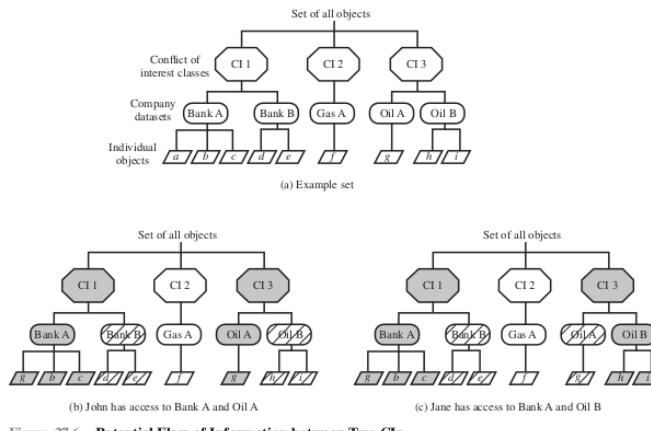


Figure 27.6 Potential Flow of Information between Two CIs

John ha fatto la sua prima richiesta di lettura a qualsiasi oggetto in questo set per un oggetto nella Banca A DS. Poiché John non ha precedentemente avuto accesso ad un oggetto in nessun altro DS in CI 1, l'accesso accesso è concesso. Inoltre, il sistema deve ricordare che l'accesso è stato concesso in modo che ogni successiva richiesta di accesso a un oggetto nella Banca B DS sarà negata. Qualsiasi richiesta di accesso ad altri oggetti nella Banca A DS è concessa. In un momento successivo, John richiede l'accesso ad un oggetto nella Banca A DS. Poiché non c'è conflitto, questo accesso viene concesso, ma viene creato un muro che proibisce il successivo accesso all'Oil B DS, come mostrato nella Figura 27.6b. Allo stesso modo, la Figura 27.6c riflette la storia di accesso alternativo di Jane. Nel nostro esempio, John ha accesso a Oil A DS e Bank A DS; Jane ha accesso a Oil B DS e Bank A DS. Se John è autorizzato a leggere dall'Oil A DS e scrivere nel Bank A DS, John può trasferire informazioni sull'olio A nel Bank A DS; ciò è indicato dal cambiamento del valore del primo oggetto sotto il Bank A DS a g. I dati possono poi essere letti da Jane. Così, Jane avrebbe accesso alle informazioni sia sul petrolio A che sul petrolio B, creando un conflitto di interessi.

Per prevenire questo, il CWM ha una **seconda regola**

Regola della proprietà: Un soggetto S può scrivere un oggetto O solo se:

- S può leggere O secondo la regola di sicurezza semplice, E
- Tutti gli oggetti che S può leggere sono nella stessa DS di O.

Detto altrimenti, o il soggetto non può scrivere affatto, o l'accesso di un soggetto (sia lettura e scrittura) è limitato ad un singolo set di dati. Così, nella figura 27.6, né John né Jane ha accesso in scrittura a qualsiasi oggetto nell'universo complessivo dei dati.

La regola proprietà è abbastanza restrittiva. Tuttavia, in molti casi, un utente ha solo bisogno dell'accesso in lettura perché l'utente sta eseguendo qualche ruolo di analisi.

Per facilitare in qualche modo la restrizione di scrittura, il modello include il concetto di dati sanificati. In sostanza, i dati sanificati sono dati che possono essere derivati dai dati aziendali ma che non possono essere usati per scoprire l'identità della società. Qualsiasi DS che consiste esclusivamente di dati sanificati non ha bisogno di essere protetto da un muro; quindi, le due regole CWM non si non si applicano a tali DS.

9.3 Il concetto di sistemi fidati

9.3.1 Applicazione sicurezza multilivello

Sicurezza multilivello (MLS) è un modo di funzionamento del sistema in cui:

- **Due o più livelli di sicurezza** delle informazioni possono essere gestiti simultaneamente all'interno dello stesso sistema quando alcuni utenti che hanno accesso al sistema non hanno né un nulla osta di sicurezza né la necessità di sapere per alcuni dei dati gestiti dal sistema.
- **La separazione degli utenti e del materiale classificato sulla base** dell'autorizzazione e del livello di classificazione dipendono dal controllo del sistema operativo.

La sicurezza multilivello è interessante quando c'è la necessità di mantenere una risorsa, come un file system o un database in cui sono definiti più livelli di sensibilità dei dati. La gerarchia potrebbe essere semplice come due livelli (ad esempio, pubblico e proprietario) o potrebbe avere molti livelli (ad esempio, il militare non classificato, riservato, confidenziale, segreto, top secret). Le tre sezioni precedenti ci hanno introdotto agli elementi essenziali della sicurezza multilivello.

In questa sezione, esaminiamo due applicazioni in cui sono stati applicati i concetti di MLS:

1. Sistema di controllo degli accessi basato sui ruoli.
2. La sicurezza dei database.

9.3.2 Sicurezza multilivello per il controllo dell'accesso basato sui ruoli

Mostra come un sistema di controllo degli accessi basato su regole (RBAC) può essere usato per implementare le regole di sicurezza multilivello BLP. Ricordiamo che la specifica

ANSI standard RBAC ANSI includeva il concetto di funzioni amministrative, che forniscono la capacità di creare, cancellare e mantenere elementi e relazioni RBAC. È utile quindi assegnare ruoli amministrativi speciali a queste funzioni. Con questo in mente, La

Tabella 27.2 riassume i componenti di un RBAC.

La seguente specifica formale indica come un sistema RBAC può essere usato per implementare l'accesso MLS:

- **Vincolo sugli utenti:**

Per ogni utente u nell'insieme degli utenti U , viene assegnato un nulla osta di sicurezza $L(u)$. Formalmente, qualsiasi $u \in U$ $[L(u) \text{dato}]$.

- **Vincoli sui permessi:**

Ogni permesso assegna un permesso di lettura o scrittura a un oggetto o , e ogni oggetto ha un permesso di lettura e uno di scrittura. Tutti gli oggetti hanno una classificazione di sicurezza. Formalmente, $P = \{(o, r), (o, w) \mid o \in \text{oggetto nel sistema}\}$ qualsiasi $o \in P$ $[L(o)]$ è dato].

- **Definizioni:**

Il livello di lettura di un ruolo r , denotato $r\text{-level}(r)$, è il minimo limite superiore dei livelli di sicurezza degli oggetti per i quali (o, r) è nelle autorizzazioni di r . Il livello w di un ruolo r (denotato $w\text{-level}(r)$) è il massimo limite inferiore (glb) dei livelli di sicurezza degli oggetti o per i quali (o, w) è nelle permessi di r , se tale glb esiste. Se il glb non esiste, il livello w è indefinito.

- **Vincoli su UA:**

Ogni ruolo r ha un livello di scrittura definito, denotato $w\text{-level}(r)$. Per ogni assegnazione dell'utente, l'autorizzazione dell'utente deve dominare il livello r del ruolo ed essere dominata dal livello w del ruolo. Formalmente, qualsiasi $r \in UA$ [$w\text{-level}(r)$ è definito]; qualsiasi $(u, r) \in UA$ $[L(u) \geq r\text{-livello}(r)]$; qualsiasi $(u, r) \in UA$ $[L(u) \leq w\text{-livello}(r)]$.

Le definizioni e i vincoli precedenti applicano il modello BLP. Un ruolo può includere permessi di accesso per più oggetti. Il livello r del ruolo indica la più alta classificazione di sicurezza per gli oggetti assegnati al ruolo. Così, la semplice proprietà di sicurezza (nessuna lettura) richiede che un utente possa essere assegnato a un ruolo solo se l'autorizzazione dell'utente è almeno pari al livello r del ruolo. Allo stesso modo, il livello w del ruolo indica la classificazione di sicurezza più bassa dei suoi oggetti. La proprietà sicurezza (no write down) richiede che un utente sia assegnato a un ruolo solo se il suo non è superiore al livello w del ruolo.

Table 27.2 RBAC Elements

U , a set of users
R and AR , disjoint sets of (regular) roles and administrative roles
P and AP , disjoint sets of (regular) permissions and administrative permissions
S , a set of sessions
$PA \subseteq P \times R$, a many-to-many permission to role assignment relation $APA \subseteq AP \times AR$, a many-to-many permission to administrative role assignment relation
$UA \subseteq U \times R$, a many-to-many user to role assignment relation $AUA \subseteq U \times AR$, a many-to-many user to administrative role assignment relation
$RH \subseteq R \times R$, a partially ordered role hierarchy $ARH \subseteq AR \times AR$, partially ordered administrative role hierarchy (both hierarchies are written as \geq in infix notation)
<i>User:</i> $S \rightarrow U$, a function mapping each session s_i to the single user $user(s_i)$ (constant for the session's lifetime) <i>Roles:</i> $S \rightarrow 2^{RUAR}$ maps each session s_i to a set of roles and administrative roles <i>Roles:</i> $(S_i \subseteq \{r \mid \exists r' \geq r\} [user(s_i).r') \in UA \cup AUA])$ (which can change with time) sessions s_i has the permissions $\bigcup_{r \in \text{roles}(s_i)} \{p \mid \exists r'' \leq r \in PA \cup APA\}$
There is a collection of constraints stipulating which values of the various components enumerated above are allowed or forbidden.

9.3.3 Sicurezza dei database e sicurezza multilivello

L'aggiunta della sicurezza multilivello a un sistema di database aumenta la complessità della funzione di controllo dell'accesso e del design del database stesso. Una questione chiave è la granularità della classificazione. I seguenti sono possibili metodi per imporre la sicurezza multilivello su un database relazionale, in termini di granularità di classificazione (vedi Figura 27.9):

- **Intero database:** Questo semplice approccio è facilmente realizzabile su una piattaforma MLS. Un intero database, come un database finanziario o personale, potrebbe essere classificato come confidenziale o riservato e mantenuto su un server con altri file.
- **Tabelle individuali (relazioni):** Per alcune applicazioni, è appropriato assegnare classificazione a livello di tabella. Nell'esempio della Figura 27.9a, sono definiti due livelli di classificazione:
 - Unrestricted (U)
 - Restricted (R)

La tabella Employee contiene informazioni sensibili sullo stipendio ed è classificata ristretta, mentre la tabella tabella Department è illimitata. Questo livello di granularità è relativamente facile da implementare e applicare.

- **Colonne individuali (attributi):** Un amministratore della sicurezza può scegliere di determinare la classificazione sulla base degli attributi, in modo che le colonne selezionate sono classificate. Nell'esempio della Figura 27.9b, l'amministratore determina che le informazioni sullo stipendio e l'identità dei responsabili di reparto sono informazioni riservate.

- **Righe individuali (tuple):** altre circostanze, può avere senso assegnare livelli di classificazione sulla base di singole righe che corrispondono a certe proprietà. Nell'esempio della figura 27.9c, tutte le righe della tabella Department che contengono informazioni relative al dipartimento di contabilità (Dept. ID = 4), e tutte le righe nella tabella Employee per le quali lo stipendio è maggiore di 50K sono limitate.
- **Elementi individuali:** Lo schema più difficile da implementare e gestire è uno in cui i singoli elementi possono essere classificati selettivamente. Nell'esame Figura 27.9d, le informazioni sullo stipendio e l'identità del manager del eparto contabilità sono limitate

La granularità dello schema di classificazione influisce sul modo in cui il controllo dell'accesso viene applicato. In particolare, gli sforzi per prevenire l'inferenza dipendono dalla granularità della classificazione.

Department Table - U			Employee Table - R			
Did	Name	Mgr	Name	Did	Salary	Eid
4	accts	Cathy	Andy	4	43K	2345
8	PR	James	Calvin	4	35K	5088

(a) Classified by table

Department Table			Employee Table			
Did - U	Name - U	Mgr - R	Name - U	Did - U	Salary - R	Eid - U
4	accts	Cathy	Andy	4	43K	2345
8	PR	James	Calvin	4	35K	5088

(b) Classified by column (attribute)

Department Table				Employee Table				
Did	Name	Mgr		Name	Did	Salary	Eid	
4	accts	Cathy	R	Andy	4	43K	2345	U
8	PR	James	U	Calvin	4	35K	5088	U

(c) Classified by row (tuple)

Department Table			Employee Table				
Did	Name	Mgr	Name	Did	Salary	Eid	
4 - U	accts - U	Cathy - R	Andy - U	4 - U	43K - U	2345 - U	
8 - U	PR - U	James - R	Calvin - U	4 - U	35K - U	5088 - U	

(d) Classified by element

Figure 27.9 Approaches to Database Classification

9.4 Trusted Computing e il Trusted Platform Module

Il trusted platform module (TPM) è un concetto standardizzato da un consorzio industriale consorzio industriale, il Trusted Computing Group. Il TPM è un modulo hardware che è al cuore di un approccio hardware/software all'informatica di fiducia. Infatti, il termine trusted computing (TC) è ora usato nell'industria per riferirsi a questo tipo di approccio hardware/approccio hardware/software. L'approccio TC impiega un chip TPM nella scheda madre del personal computer o una smart card o integrato nel processore principale, insieme all'hardware e al software che in qualche modo è stato approvato o certificato per lavorare con il TPM.

Il TPM genera chiavi che condivide con i componenti vulnerabili che passano dati all'interno del sistema, come i dispositivi di archiviazione, i componenti di memoria e l'hardware audio/visivo. hardware audio/video. Le chiavi possono essere usate per crittografare i dati che fluiscono attraverso la macchina. Il TPM funziona anche con il software abilitato a TC, incluso il sistema operativo e le applicazioni.

Il software può essere sicuro che i dati che riceve sono affidabili, e il sistema può essere sicuro che il software stesso sia affidabile.

Per ottenere queste caratteristiche, TC fornisce tre servizi di base: avvio autenticato, certificazione e crittografia.

9.4.1 Servizio di avvio autenticato

Il servizio di avvio autenticato è responsabile dell'avvio dell'intero sistema operativo in fasi e assicurando che ogni porzione del sistema operativo, quando viene caricata, sia una versione che è approvato per l'uso. Tipicamente, l'avvio di un sistema operativo inizia con un piccolo pezzo di codice nella ROM. Questo pezzo porta altro codice dal blocco di avvio sul disco rigido e trasferisce l'esecuzione a quel codice. Questo processo continua con blocchi sempre più grandi del codice del sistema operativo fino a quando l'intera procedura di avvio del sistema operativo è completa e il sistema operativo residente è avviato. Ad ogni stadio, l'hardware del TC controlla che il software valido sia stato portato dentro.

Questo può essere fatto verificando una firma digitale associata al software.

Il TPM tiene un registro a prova di manomissione del processo di caricamento, usando una funzione di hash crittografica per rilevare qualsiasi manomissione del registro. Quando il processo è completato, il registro a prova di manomissione contiene un record che stabilisce esattamente quale versione del sistema operativo e i suoi vari moduli sono in esecuzione. È ora possibile espandere il confine di fiducia per includere ulteriore hardware e applicazioni e software di utilità. Il sistema abilitato TC mantiene una lista approvata di componenti hardware e software approvati. Per configurare un pezzo di hardware o caricare un software, il sistema controlla se il componente è nella lista approvata, se è firmato digitalmente (dove applicabile), e se il suo numero di serie non è stato revocato. Il risultato è una configurazione di hardware, software di sistema e applicazioni che è in uno stato ben definito con componenti approvati.

9.4.2 Servizio di certificazione

Una volta che una configurazione è raggiunta e registrata dal TPM, il TPM può certificare la configurazione ad altre parti. Il TPM può produrre un certificato digitale firmando una descrizione formattata delle informazioni di configurazione usando la chiave privata del TPM. Così, un altro utente, sia un utente locale che un sistema remoto, può avere fiducia che una configurazione inalterata è in uso perché:

1. Il TPM è considerato affidabile. Non abbiamo bisogno di un'ulteriore certificazione del TPM stesso.
2. Solo il TPM possiede la chiave privata di questo TPM. Un destinatario della configurazione può usare la chiave pubblica del TPM per verificare la firma (vedi Figura 2.7b).

Per assicurare che la configurazione sia puntuale, un richiedente emette una "sfida" sotto forma di un numero casuale quando richiede un certificato firmato dal TPM.

Il TPM firma un blocco di dati che consiste nelle informazioni di configurazione con il numero casuale aggiunto ad esso. Il richiedente può quindi verificare che il certificato sia valido e aggiornato.

Lo schema TC prevede un approccio gerarchico alla certificazione. Il TPM certifica la configurazione hardware/OS. Poi il sistema operativo può certificare la presenza e la configurazione dei programmi applicativi. Se un utente si fida del TPM e si fida della versione certificata del sistema operativo, allora l'utente può avere fiducia nella configurazione dell'applicazione.

9.4.3 Servizio di crittografia

Il servizio di crittografia permette la crittografia dei dati in modo tale che i dati possano essere decifrati solo da una certa macchina, e solo se questa macchina è in una certa configurazione. Ci sono diversi aspetti di questo servizio.

In primo luogo, il TPM mantiene una chiave segreta principale unica per questa macchina. Da questa chiave, il TPM genera una chiave di crittografia segreta per ogni possibile configurazione di quella macchina. Se i dati sono criptati mentre la macchina è in una configurazione, i dati possono essere decifrati solo usando quella stessa configurazione. Se una configurazione diversa viene creata sulla macchina, la nuova configurazione non sarà in grado di decifrare i dati crittografati da una configurazione diversa.

Questo schema può essere esteso verso l'alto, come si fa con la certificazione. Così, è possibile fornire una chiave di crittografia ad un'applicazione in modo che l'applicazione possa criptare i dati, e la decriptazione possa essere fatta solo dalla versione desiderata dell'applicazione desiderata che gira sulla versione desiderata del sistema operativo desiderato. Questi dati crittografati possono essere memorizzati localmente, recuperabili solo dall'applicazione che li ha memorizzati, o trasmessi a un'applicazione peer su una macchina remota. L'applicazione peer dovrebbe essere nella stessa configurazione per decifrare i dati.

9.4.4 Funzioni TPM

Per dare un'idea del funzionamento di un sistema TC/TPM, guardiamo la funzione di memorizzazione protetta. Il TPM genera e memorizza un certo numero di chiavi crittografia in una gerarchia di fiducia. Alla radice della gerarchia c'è una chiave radice di memorizzazione generata dal TPM e accessibile solo per l'uso del TPM. Da questa chiave, altre chiavi possono essere generate e protette dalla crittografia con chiavi più vicine alla radice della gerarchia.

Una caratteristica importante delle Trusted Platforms è che un oggetto protetto dal TPM può essere "sigillato" ad un particolare stato del software in una piattaforma. Quando l'oggetto protetto TPM viene creato, il creatore indica lo stato del software che deve esistere se il segreto deve essere rivelato. Quando un TPM scarta l'oggetto protetto TPM (all'interno del TPM e nascosto alla vista), il TPM controlla che lo stato attuale del software corrisponda allo stato del software indicato. Se corrispondono, il TPM permette l'accesso al segreto. Se non corrispondono, il TPM nega l'accesso al segreto.

La figura 27.12 fornisce un esempio di questa protezione. In questo caso, c'è un file criptato sulla memoria locale a cui un'applicazione utente desidera accedere. I seguenti passi da compiere:

1. La chiave simmetrica che è stata usata per criptare il file è memorizzata con il file. La chiave stessa è criptata con un'altra chiave a cui il TPM ha accesso. La chiave protetta è presentata al TPM con una richiesta di rivelare la chiave all'applicazione.
2. Associata alla chiave protetta è una specifica della configurazione hardware/software che può avere accesso alla chiave. Il TPM verifica che la configurazione corrente corrisponda alla configurazione richiesta per rivelare la chiave. Inoltre, l'applicazione richiedente deve essere specificamente autorizzata ad accedere alla chiave. Il TPM usa un protocollo di autorizzazione per verificare l'autorizzazione.
3. Se la configurazione corrente permette l'accesso alla chiave protetta, allora il TPM decifra la chiave e la passa all'applicazione.
4. L'applicazione usa la chiave per decifrare il file. L'applicazione è affidabile per poi scartare la chiave in modo sicuro.

La crittografia di un file procede in modo analogo. In quest'ultimo caso, un processo richiede una chiave simmetrica per cifrare il file. Il TPM fornisce quindi una versione criptata della chiave da memorizzare con il file.

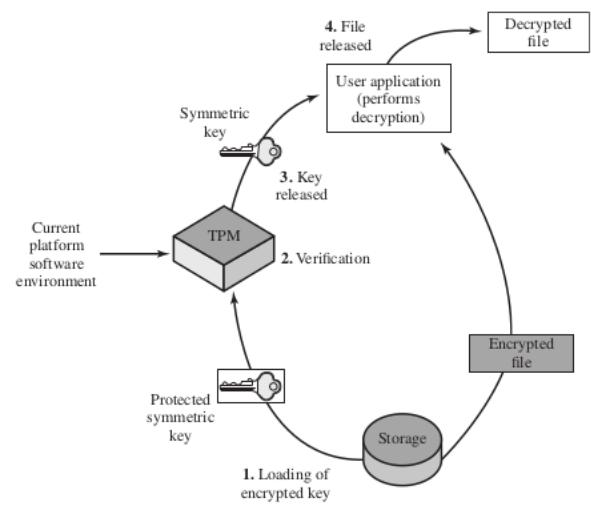


Figure 27.12 Decryption a File Using a Protected Key

9.4.5 Requisiti

Il CC definisce un insieme comune di potenziali requisiti di sicurezza da usare nella valutazione. Il termine obiettivo della valutazione (TOE) si riferisce a quella parte del prodotto o sistema che è soggetto alla valutazione. I requisiti rientrano in due categorie:

1. Requisiti funzionali

Definiscono il comportamento di sicurezza desiderato. I documenti CC stabiliscono un insieme di componenti funzionali di sicurezza che forniscono un modo standard di esprimere i requisiti funzionali di sicurezza per un TOE.

2. Requisiti di sicurezza

La base per ottenere la fiducia che le misure di sicurezza dichiarate misure di sicurezza dichiarate siano efficaci e implementate correttamente. I documenti CC stabiliscono un insieme di componenti di garanzia che forniscono un modo standard di esprimere i requisiti di garanzia per un TOE.

Sia i requisiti funzionali che quelli di garanzia sono organizzati in classi:

Una classe è una collezione di requisiti che condividono un obiettivo o un intento comune.

Le tabelle 27.3 e 27.4 definiscono brevemente le classi per i requisiti funzionali e di garanzia. Ciascuna di queste classi contiene un certo numero di famiglie. I requisiti all'interno di ogni famiglia condividono obiettivi di sicurezza, ma differiscono per enfasi o rigore. Per esempio, la classe di audit contiene sei famiglie che si occupano di vari aspetti dell'auditing (ad es, generazione di dati di audit, analisi di audit e memorizzazione di eventi di audit). Ogni famiglia, a sua volta, contiene uno o più componenti. Un componente descrive un insieme specifico di requisiti di sicurezza requisiti di sicurezza ed è il più piccolo insieme selezionabile di requisiti di sicurezza da includere nelle strutture definite nel CC.

Table 27.3 CC Security Functional Requirements

Class	Description
Audit	Involves recognizing, recording, storing, and analyzing information related to security activities. Audit records are produced by these activities and can be examined to determine their security relevance.
Cryptographic support	Used when the TOE implements cryptographic functions. These may be used, for example, to support communications, identification and authentication, or data separation.
Communications	Provides two families concerned with nonrepudiation by the originator and by the recipient of data.
User data protection	Specifies requirements relating to the protection of user data within the TOE during import, export, and storage, in addition to security attributes related to user data.
Identification and authentication	Ensure the unambiguous identification of authorized users and the correct association of security attributes with users and subjects.
Security management	Specifies the management of security attributes, data, and functions.
Privacy	Provides a user with protection against discovery and misuse of his or her identity by other users.
Protection of the TOE security functions	Focused on protection of TSF (TOE security functions) data rather than of user data. The class relates to the integrity and management of the TSF mechanisms and data.
Resource utilization	Supports the availability of required resources, such as processing capability and storage capacity. Includes requirements for fault tolerance, priority of service, and resource allocation.
TOE access	Specifies functional requirements, in addition to those specified for identification and authentication, for controlling the establishment of a user's session. The requirements for TOE access govern such things as limiting the number and scope of user sessions, displaying the access history, and modifying access parameters.
Trusted path/channels	Concerned with trusted communications paths between the users and the TSF and between TSFs.

Table 27.4 CC Security Assurance Requirements

Class	Description
Configuration management	Requires that the integrity of the TOE is adequately preserved. Specifically, configuration management provides confidence that the TOE and documentation used for evaluation are the ones prepared for distribution.
Delivery and operation	Concerned with the measures, procedures, and standards for secure delivery, installation, and operational use of the TOE, to ensure that the security protection offered by the TOE is not compromised during these events.
Development	Concerned with the refinement of the TSF from the specification defined in the ST to the implementation, and a mapping from the security requirements to the lowest level representation.
Guidance documents	Concerned with the secure operational use of the TOE, by the users and administrators.
Life cycle support	Concerned with the life cycle of the TOE include life cycle definition, tools and techniques, security of the development environment, and remediation of flaws found by TOE consumers.
Tests	Concerned with demonstrating that the TOE meets its functional requirements. The families address coverage and depth of developer testing, and requirements for independent testing.
Vulnerability assessment	Defines requirements directed at the identification of exploitable vulnerabilities, which could be introduced by construction, operation, misuse, or incorrect configuration of the TOE. The families identified here are concerned with identifying vulnerabilities through covert channel analysis, analyzing the configuration of the TOE, examining the strength of mechanisms of the security functions, and identifying flaws introduced during development of the TOE. The second family covers the security categorization of TOE components. The third and fourth cover the analysis of changes for security impact and the provision of evidence that procedures are being followed. This class provides building blocks for the establishment of assurance maintenance schemes.
Assurance maintenance	Provides requirements that are intended to be applied after a TOE has been certified against the CC. These requirements are aimed at assuring that the TOE will continue to meet its security target as changes are made to the TOE or its environment.

9.4.6 Profili e Obiettivi

Il CC definisce anche due tipi di documenti che possono essere generati usando i requisiti definiti dal CC.

- **Profili di protezione (PP):** Definiscono un insieme indipendente dall'implementazione dei requisiti e obiettivi di sicurezza per una categoria di prodotti o sistemi che soddisfano esigenze simili dei consumatori per la sicurezza informatica.

Un PP è inteso essere riutilizzabile e definire requisiti che sono noti per essere utili ed efficaci nel soddisfare gli obiettivi identificati. Il concetto di PP è stato sviluppato per supportare la definizione di standard funzionali e come aiuto alla formulazione di specifiche di approvvigionamento. Il PP riflette la sicurezza dell'utente esigenze degli utenti.

- **Obiettivi di sicurezza (ST):** Contengono gli obiettivi e i requisiti di sicurezza IT di uno specifico TOE identificato e definiscono le misure funzionali e di garanzia offerte da quel TOE per soddisfare i requisiti dichiarati.

9.4.7 Esempio di protezione di un profilo

Il profilo di protezione per una smart card, sviluppato dallo Smart Card Security User Group, fornisce un semplice esempio di PP.

Questo PP descrive i requisiti di sicurezza IT per una smart card da usare in connessione con applicazioni sensibili, come i sistemi di pagamento finanziario dell'industria bancaria. Il livello di garanzia per questo PP è EAL 4, che è descritto nella seguente sottosezione. Il PP elenca le minacce che devono essere affrontate da un prodotto che dichiara di essere conforme a questo PP. Le minacce includono seguenti:

- **Sondaggio fisico:** Può comportare la lettura di dati dal TOE attraverso tecniche comunemente impiegate nell'analisi dei guasti IC e negli sforzi di reverse engineering IC.
- **Input non valido:** L'input non valido può assumere la forma di operazioni che non sono correttamente, richieste di informazioni oltre i limiti del registro, o tentativi di trovare ed eseguire comandi non documentati. Il risultato di un tale attacco può essere una compromissione delle funzioni di sicurezza, la generazione di errori sfruttabili nel funzionamento, o il rilascio di dati protetti.
- **Collegamento di più operazioni:** Un attaccante può osservare usi multipli di risorse o servizi e, collegando queste osservazioni, dedurre informazioni che possono rivelare dati sulla funzione di sicurezza.

Dopo un elenco di minacce, il PP passa alla descrizione degli obiettivi di sicurezza. Questi riflettono l'intento dichiarato di contrastare le minacce identificate e/o conformarsi a qualsiasi politiche di sicurezza organizzativa identificate. Sono elencati diciannove obiettivi, tra cui i seguenti:

- **Audit:** Il sistema deve fornire i mezzi per registrare determinati eventi rilevanti per la sicurezza eventi rilevanti per la sicurezza, in modo da assistere un amministratore nell'individuazione potenziali attacchi o configurazioni errate delle caratteristiche di sicurezza del sistema che lo lascerebbero suscettibile di attacco.
- **Inserimento dei guasti:** Il sistema deve essere resistente a sondaggi ripetuti attraverso inserimento di dati errati.
- **Perdita di informazioni:** Il sistema deve fornire i mezzi per controllare e limitare la perdita di informazioni nel sistema in modo che nessuna informazione utile sia rivelata attraverso le linee di alimentazione, terra, clock, reset o I/O.

I requisiti di sicurezza sono forniti per contrastare minacce specifiche e per supportare politiche specifiche sotto specifiche ipotesi. Il PP elenca requisiti specifici in tre aree generali: requisiti funzionali di sicurezza del TOE, requisiti di sicurezza del TOE, e requisiti di sicurezza per l'ambiente IT.

Nell'area dei requisiti funzionali di sicurezza, il PP definisce 42 requisiti delle classi disponibili di requisiti funzionali di sicurezza (vedi tabella 27.3).

Per esempio, per l'auditing di sicurezza, il PP stabilisce cosa deve controllare il sistema; quali informazioni devono essere registrate; quali sono le regole per monitorare, operare e proteggere i registri, e così via. I requisiti funzionali sono anche elencati da le altre classi di requisiti funzionali, con dettagli specifici per il funzionamento della smart card.

Il PP definisce 24 requisiti di garanzia della sicurezza dalle classi disponibili di requisiti di garanzia della sicurezza (vedi tabella 27.4). Questi requisiti sono stati scelti per dimostrare:

- La qualità della progettazione e della configurazione del prodotto
- Che viene fornita una protezione adeguata durante la progettazione e l'implementazione del prodotto
- Che il test del prodotto da parte del fornitore rispetta parametri specifici
- Che la funzionalità di sicurezza non è compromessa durante la consegna del prodotto
- che la guida per l'utente, compresi i manuali del prodotto relativi all'installazione, alla manutenzione e all'uso, siano di una qualità specifica, la manutenzione e l'uso, siano di una specifica qualità e adeguatezza

Il PP elenca anche i requisiti di sicurezza dell'ambiente IT. Questi coprono i seguenti argomenti:

- Distribuzione delle chiavi crittografiche
- Distruzione della chiave crittografica
- Ruoli di sicurezza

9.5 Assicurazione e valutazione

La garanzia può essere definita come una misura di fiducia che le caratteristiche di sicurezza e l'architettura di un sistema informativo (IS) mediano e applicano accuratamente la politica di sicurezza. Se si fa affidamento sulle caratteristiche di sicurezza di un IS per proteggere informazioni classificate o sensibili e limitare l'accesso degli utenti, le caratteristiche devono essere testate per assicurare che la politica di sicurezza sia applicata. Come per qualsiasi altro aspetto della sicurezza informatica, le risorse dedicate alla garanzia devono essere sottoposte a una sorta di analisi costi-benefici per determinare quale quantità di sforzo sia ragionevole per il livello di garanzia desiderato.

9.5.1 Destinatari

Il design delle misure di garanzia dipende in parte dal pubblico a cui queste misure. Cioè, nello sviluppare un grado di fiducia nelle misure di sicurezza, dobbiamo specificare quali individui o gruppi possiedono quel grado di fiducia. Il documento del CC sull'assicurazione elenca i seguenti destinatari:

- **Consumatori:** Selezionano le caratteristiche e le funzioni di sicurezza per un sistema e determinano i livelli richiesti di garanzia di sicurezza.
- **Sviluppatori:** Rispondono ai requisiti di sicurezza reali o percepiti dai consumatori; interpretare le dichiarazioni dei requisiti di sicurezza e determinare gli approcci e livello di sforzo.
- **Valutatori:** Usano i requisiti di garanzia come una dichiarazione obbligatoria di criteri di valutazione quando valutano le caratteristiche e i controlli di sicurezza.

I valutatori possono essere nella stessa organizzazione dei consumatori o un team di valutazione di terze parti.

9.5.2 Ambito di garanzia

La garanzia si occupa delle caratteristiche di sicurezza dei prodotti IT, come computer, database, sistemi operativi e sistemi completi. La garanzia si applica a i seguenti aspetti di un sistema:

- **Requisiti:** Questa categoria si riferisce ai requisiti di sicurezza di un prodotto
- **Politica di sicurezza:** Sulla base dei requisiti, può essere definita una politica di sicurezza
- **Progettazione del prodotto:** Sulla base dei requisiti e della politica di sicurezza
- **Implementazione del prodotto:** Basato sulla progettazione
- **Funzionamento del sistema:** Include l'uso ordinario più la manutenzione

In ogni area, si possono adottare diversi approcci per fornire garanzie.

I possibili approcci:

- Analisi e controllo dei processi e delle procedure
- Verifica dell'applicazione dei processi e delle procedure
- Analisi della corrispondenza tra le rappresentazioni del progetto TOE
- Analisi della rappresentazione del progetto TOE rispetto ai requisiti
- Verifica delle prove
- Analisi dei documenti di guida
- Analisi dei test funzionali sviluppati e dei risultati forniti
- Test funzionali indipendenti
- Analisi delle vulnerabilità (inclusa l'ipotesi di difetti)
- Penetration Testing

Siccome viene fornita una visione un po' diversa degli elementi di garanzia. Questa relazione è basata sull'esperienza con le valutazioni di Orange Book, ma è rilevante per gli attuali sforzi di sviluppo di prodotti affidabili. L'autore vede la garanzia come comprendente i seguenti requisiti:

- **Architettura del sistema**

Riguarda sia la fase di sviluppo del sistema che la fase operativa del sistema. Esempi di tecniche per aumentare il livello di garanzia durante la fase di sviluppo includono la progettazione modulare del software, la stratificazione e l'astrazione dei dati/nascondere le informazioni.

- **Integrità del sistema**

Riguarda il corretto funzionamento dell'hardware e del firmware ed è tipicamente soddisfatto dall'uso periodico di software diagnostico.

- **Test del sistema**

Assicura che le caratteristiche di sicurezza siano state testate a fondo. Questo include il test delle operazioni funzionali, il test dei requisiti di sicurezza, e test di possibili penetrazioni.

- **Specifiche e verifica del design**

Affronta la correttezza del design e dell'implementazione del sistema progettazione e implementazione del sistema rispetto alla politica di sicurezza del sistema. Idealmente, possono essere usati metodi formali di verifica.

- **Gestione fidata della struttura**

Si occupa dell'amministrazione del sistema. Un approccio è quello di separare i ruoli di operatore del sistema e di amministratore della sicurezza. Un altro approccio è la specificazione dettagliata di politiche e procedure con meccanismi per la revisione.

- **Recupero affidabile**

Fornisce il corretto funzionamento delle funzioni di sicurezza dopo che un sistema si riprende da guasti, crash o incidenti di sicurezza.

- **Distribuzione fidata**

Assicura che hardware, firmware e software protetti non subiscano modifiche non autorizzate durante il transito dal fornitore al cliente

- **Gestione della configurazione**

I requisiti sono inclusi per la configurazione controllo, audit, gestione e contabilità

Così, vediamo che la garanzia si occupa della progettazione, dell'implementazione e del funzionamento delle risorse protette e delle loro funzioni e procedure di sicurezza. È importante notare che la garanzia è un processo, non un risultato. Cioè, la garanzia deve essere un'attività continua, che include test, verifiche e revisioni.

9.5.3 Processo di valutazione

Lo scopo della valutazione di un prodotto IT, un TOE, rispetto a uno standard informatico affidabile è quello di garantire che le caratteristiche di sicurezza nel TOE funzionino correttamente ed efficacemente, e non mostrino vulnerabilità sfruttabili. Il processo di valutazione viene eseguito sia in parallelamente o dopo lo sviluppo del TOE, a seconda del livello di garanzia richiesto.

Più alto è il livello, maggiore è il rigore richiesto dal processo, e più tempo e spese si dovranno sostenere.

I principali input per la valutazione sono l'obiettivo di sicurezza, un insieme di prove sul TOE e il TOE attuale. Il risultato desiderato del processo di valutazione è confermare che l'obiettivo di sicurezza è soddisfatto per il TOE, confermato da prove documentate nel rapporto di valutazione tecnica. Il processo di valutazione metterà in relazione l'obiettivo di sicurezza con uno o più dei seguenti elementi progettazione di alto livello, progettazione di basso livello, specifiche funzionali, implementazione del codice sorgente, codice oggetto e realizzazione hardware del TOE. Il grado di rigore utilizzato e la profondità dell'analisi sono determinati dal livello di garanzia desiderato per la valutazione. Ai livelli più alti, si usano modelli semiformali o formali per confermare che il TOE implementa effettivamente l'obiettivo di sicurezza desiderato. Il processo di valutazione comporta anche il processo di valutazione implica anche un attento test del TOE per confermare le sue caratteristiche di sicurezza.

La valutazione coinvolge una serie di parti:

- **Sponsor:** Di solito o il cliente o il fornitore di un prodotto per il quale è richiesta la valutazione. Gli sponsor determinano l'obiettivo di sicurezza che il prodotto deve soddisfare.
- **Sviluppatore:** Deve fornire prove adeguate sui processi usati per progettare, implementare e testare il prodotto per permetterne la valutazione.
- **Valutatore:** Esegue il lavoro di valutazione tecnica, usando le prove fornite dagli sviluppatori, e ulteriori test del prodotto, per confermare che esso soddisfi i requisiti funzionali e di garanzia specificati nell'obiettivo di sicurezza.
- **Certificatore:** L'agenzia governativa che controlla il processo di valutazione e in seguito certifica che un prodotto è successivamente certifica che un prodotto è stato valutato con successo. I certificatori generalmente un registro dei prodotti valutati, che può essere consultato dai clienti.

Il processo di valutazione ha tre grandi fasi:

1. **Preparazione:** Involge il contatto iniziale tra lo sponsor e gli sviluppatori di un prodotto e i valutatori che lo valuteranno. Confermerà che lo sponsor e gli sviluppatori

sono adeguatamente preparati a condurre la valutazione e includerà una revisione dell'obiettivo di sicurezza e possibilmente altre consegne di valutazione.

2. **Conduzione della valutazione:** Un processo strutturato e formale in cui i valutatori conducono una serie di attività specificate dal CC. Queste includono la revisione dei prodotti forniti dallo sponsor e dagli sviluppatori, e altri test del prodotto, per confermare che soddisfa l'obiettivo di sicurezza. Durante questo processo, possono essere identificati nel prodotto, che vengono riportati agli sviluppatori per la correzione.
3. **Conclusione:** I valutatori forniscono il rapporto tecnico di valutazione finale ai certificatori per l'accettazione. I certificatori usano questo rapporto, che può contenere informazioni confidenziali, per convalidare il processo di valutazione e per preparare un rapporto di certificazione pubblico. Il rapporto di certificazione viene poi elencato nel relativo registro dei prodotti valutati.

Il processo di valutazione è normalmente monitorato e regolato da un'agenzia governativa in ogni paese.

Capitolo 10

Blockchain e Bitcoin

10.1 Blockchain

La blockchain (letteralmente "catena di blocchi") è una struttura dati condivisa e "immutabile". È definita come un registro digitale le cui voci sono raggruppate in "blocchi", concatenati in ordine cronologico, e la cui integrità è garantita dall'uso della crittografia. Sebbene la sua dimensione sia destinata a crescere nel tempo, è immutabile in quanto, di norma, il suo contenuto una volta scritto non è più né modificabile né eliminabile, a meno di non invalidare l'intera struttura.

Tali tecnologie sono incluse nella più ampia famiglia delle “Distributed Ledger”, ossia sistemi che si basano su un registro distribuito, che può essere letto e modificato da più nodi di una rete. Non è richiesto che i nodi coinvolti conoscano l’identità reciproca o si fidino l’uno dell’altro. Difatti, per garantire la coerenza tra le varie copie, l’aggiunta di un nuovo blocco è globalmente regolata da un protocollo condiviso. Una volta autorizzata l’aggiunta del nuovo blocco, ogni nodo aggiorna la propria copia privata: la natura stessa della struttura dati garantisce l’assenza di una sua manipolazione futura. Le caratteristiche che accomunano i sistemi sviluppati con le tecnologie Blockchain e Distributed Ledger sono digitalizzazione dei dati, decentralizzazione, disintermediazione, tracciabilità dei trasferimenti, trasparenza/verificabilità, immutabilità del registro e programmabilità dei trasferimenti. (Grazie a tali caratteristiche, la blockchain è considerata pertanto un’alternativa in termini di sicurezza, affidabilità, trasparenza e costi alle banche dati e ai registri gestiti in maniera centralizzata da autorità riconosciute e regolamentate (pubbliche amministrazioni, banche, assicurazioni, intermediari di pagamento, ecc.).)

Descrizione: Una blockchain è un registro digitale aperto e distribuito, in grado di memorizzare record di dati (solitamente, denominati "transazioni") in modo sicuro, verificabile e permanente. Una volta scritti, i dati in un blocco non possono essere retroattivamente alterati senza che vengano modificati tutti i blocchi successivi ad esso e ciò, per la natura del protocollo e dello schema di validazione, necessiterebbe del consenso della maggioranza della rete. La blockchain è quindi rappresentabile come una lista, in

continua crescita, di "blocchi" collegati tra loro e resi sicuri mediante l'uso della crittografia. Ad un blocco possono essere associate una o più transazioni e ogni blocco, inoltre, contiene un puntatore hash al blocco precedente e una marca temporale. La natura distribuita e il modello cooperativo rendono robusto e sicuro il processo di validazione, ma presentano tempi non trascurabili, dovuti in gran parte al processo di validazione dei blocchi e alla sincronizzazione delle reti. L'autenticazione avviene tramite collaborazione di massa ed è alimentata da interessi collettivi. L'utilizzo di questa tecnologia consente anche di superare il problema dell'infinita riproducibilità di un bene digitale e della doppia spesa, senza l'utilizzo di un server centrale o di un'autorità. Talvolta risulta possibile che alcuni nodi della rete producano simultaneamente più blocchi "concorrenti" (ossia collegati a uno stesso blocco già esistente, ma diversi tra loro nel contenuto): ciò dà origine a una biforcazione (fork) nella catena. Il protocollo di aggiornamento specifica quale regola i nodi debbano adottare per selezionare il blocco da accettare, tra quelli concorrenti. I blocchi non selezionati per l'inclusione nella catena sono chiamati "blocchi orfani"

10.2 Bitcoin

Il bitcoin si comporta più come "oro" che come "moneta" perché il valore cambia nel tempo. Più che proprietà o possesso di bitcoin/monete si parla di diritto di spesa, che si ottiene conoscendo una determinata chiave di cifratura.

Come faccio a ricevere bitcoin? Utilizzando un exchange (piattaforma per lo scambio di criptovalute) si ha la possibilità di inviare/ricevere queste chiavi di cifratura segrete che determinano il diritto di possesso di un certo numero di monete/criptovalute virtuali.

Esempio transazione fatto a lezione:

- A paga in euro
- A comunica la propria chiave pubblica
- B riceve gli euro da A e cede i suoi bitcoin
- La cessione avviene sbloccando i bitcoin dalla chiave privata di B per poi assegnarli alla chiave pubblica di A
- Il nuovo proprietario è A perché è l'unico ad avere una determinata chiave privata associata ad una determinata chiave pubblica

Le chiavi pubbliche e private sono associate 1 a 1, conoscendo la chiave pubblica non è possibile conoscere la chiave privata e viceversa. Quindi nell'esempio io genero una coppia chiave pubblica-privata, divulgo la pubblica e ci ottengo i bitcoin sopra che poi con la chiave privata potrò spendere o cedere a qualcun altro.

Il Bitcoin (codice: BTC o XBT) è una criptovaluta e un sistema di pagamento mondiale creato nel 2009 da un anonimo inventore (o gruppo di inventori), noto con lo pseudonimo

di Satoshi Nakamoto (|| in realtà si dice che ci fosse dietro un intero team), che sviluppò un'idea da lui stesso presentata su Internet a fine 2008. Per convenzione se il termine Bitcoin è utilizzato con l'iniziale maiuscola si riferisce alla tecnologia e alla rete, mentre se minuscola (bitcoin) si riferisce alla valuta in sé.

Dagli esperti di finanza il Bitcoin non viene classificato come una moneta, ma come una riserva di valore attualmente molto volatile. A differenza della maggior parte delle valute tradizionali, il Bitcoin non fa uso di un ente centrale né di meccanismi finanziari sofisticati, il valore è determinato unicamente dalla leva domanda e offerta: esso utilizza un database distribuito tra i nodi della rete che tengono traccia delle transazioni, ma sfrutta la crittografia per gestire gli aspetti funzionali, come la generazione di nuova moneta e l'attribuzione della proprietà dei bitcoin.

La rete Bitcoin consente il possesso e il trasferimento pseudo-anonimo delle monete; i dati necessari a utilizzare i propri bitcoin possono essere salvati su uno o più personal computer o dispositivi elettronici quali smartphone, sotto forma di "portafoglio" digitale, o mantenuti presso terze parti che svolgono funzioni simili a una banca. Il wallet bitcoin ha un indirizzo identificato da un codice alfanumerico che possiede tra i 25 e i 36 caratteri tra numeri e lettere; è l'unico dato da comunicare per ricevere un pagamento che godrà di un certo grado di anonimato, ma sarà allo stesso tempo pubblicamente e immutabilmente visibile sulla blockchain per sempre. Occorre fare molta attenzione nella trasmissione del codice alfanumerico in quanto eventuali errori non consentono di annullare l'operazione e causano la perdita del denaro. È possibile ricevere pagamenti più semplicemente attraverso la scansione di codici QR. In ogni caso, i bitcoin possono essere trasferiti attraverso Internet verso chiunque disponga di un "indirizzo bitcoin". La struttura peer-to-peer della rete Bitcoin e la mancanza di un ente centrale rende impossibile a qualunque autorità, governativa o meno, il blocco dei trasferimenti, il sequestro di bitcoin senza il possesso delle relative chiavi o la svalutazione dovuta all'immissione di nuova moneta. Il Bitcoin è una delle prime implementazioni di un concetto definito come criptovaluta, descritto per la prima volta nel 1998 da Wei Dai su una mailing list.

BitCoin è una scoperta informatica che definisce e implementa un sistema di pagamento sicuro e decentralizzato e uno strumento per l'archiviazione, la verifica e la revisione delle informazioni, comprese le rappresentazioni digitali dei valori. Il protocollo bitcoin definisce una rete overlay su Internet che estrae bitcoin, ogni nodo gestisce un gruppo di indirizzi che detiene monete, ogni indirizzo è un'immagine hash di una sottostante coppia privata-pubblica di chiavi crittografiche e agisce come uno pseudonimo del titolare della moneta. La visione dei nodi di questo stato comune è formata da una BlockChain, un libro mastro condiviso, appendibile, affidabile, di tutte le transazioni delle monete. I limiti del consenso distribuito definiti nel Problema Bizantino e nel Teorema CAP sono risolti usando la tecnica del proof-of-work.

Il bitcoin, visto soprattutto il continuo incremento/decremento del proprio valore, viene difficilmente considerato spendibile. Esiste una versione chiamata bitcoin cash che riesce a

mantenere di più il valore nel tempo, perciò viene considerata “più spendibile”. Con i bitcoin non c’è nessuna terza parte fidata per eseguire le operazioni peer-to-peer, per esempio le banche. Per avere una garanzia che la transazione venga registrata in tempo breve bisogna pagare una commissione, questa commissione verrà incassata da chi si occupa di questo tipo di operazioni, ovvero i miner.

10.2.1 Bitcoin core

Il software Bitcoin Core (evoluzione Bitcoin-Qt) può essere scaricato come qualsiasi altro programma sul nostro computer. Ma prima di ciò, è necessario tenere conto di diversi aspetti. Primo, Bitcoin Core implementa tutti gli aspetti della rete Bitcoin, quindi scaricarlo ti renderà un nodo completo della rete. Ciò include una copia esatta e completa di tutte le operazioni che sono state effettuate con Bitcoin dal suo lancio nel 2009. E, naturalmente, sarà costantemente aggiornato. Quindi la richiesta di spazio di archiviazione disponibile sul disco rigido sarà di almeno 400 GB.

Secondo Bitcoin Core implementa un wallet, attraverso il quale tutte le transazioni effettuate con la copia del file blockchain. Quindi scaricarlo e sincronizzarlo su un computer richiederà alcuni giorni prima di poterlo utilizzare. Pertanto, sebbene offra livelli elevati di sicurezza e privacy, è consigliato solo per utenti esperti. Un’altra caratteristica importante di Bitcoin Core è che utilizza un programma interno (demone) chiamato bitcoind. Un demone (demone in spagnolo) è un programma che viene eseguito in background per essere utilizzato tramite le righe di comando e chiamate di procedura remota (RPC). Il nome "demone" è strettamente correlato ai sistemi UNIX e derivati simili GNU / Linux. Bitcoin Core è anche in grado di creare un file testnet, una testnet in cui gli sviluppatori controllano le modifiche che desiderano apportare. Pertanto, possono analizzare in dettaglio come funzionano i cambiamenti o miglioramenti che desiderano per la rete prima di incorporarli in essa. Inoltre, Bitcoin Core contiene anche un programma chiamato bitcoin-cli. Questa è un’interfaccia a riga di comando, attraverso la quale gli utenti possono inviare comandi RPC a bitcoind ed eseguire qualsiasi operazione supportata da Bitcoin.

Bitcoin ecosystem

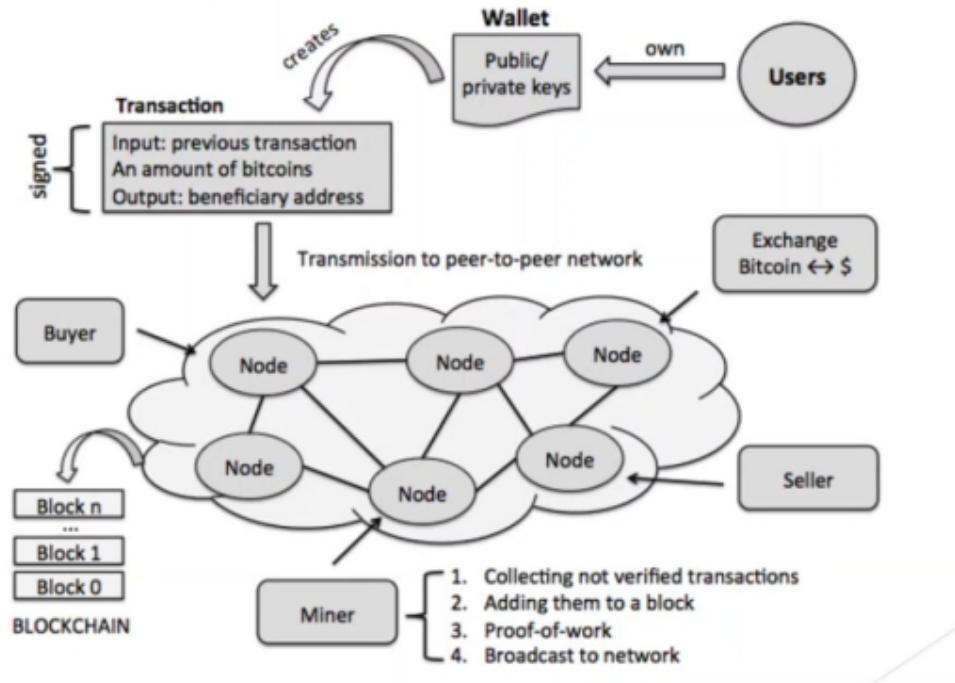


Figura 177: Ecosistema Bitcoin

Come rappresentato in Fig. 1, i principali attori della piattaforma sono gli utenti, che possiedono un portafoglio associato ad una coppia di chiavi crittografiche private/pubbliche. Gli utenti usano queste chiavi per firmare transazioni (irreversibili per progettazione), che sono poi trasmesse alla rete peer-to-peer Bitcoin. Alcuni dei pari in questa rete sono i minatori, il cui compito è quello di aggiornare la blockchain, una struttura dati pubblica e distribuita che implementa il database di ogni transazione mai eseguita. Un proprietario di bitcoin trasferisce la moneta ad un altro proprietario (ad es, un acquirente a un venditore nella Fig. 1) firmando digitalmente un hash di una transazione precedente (dimostrando che questo proprietario è in possesso di bitcoin ricevuti in precedenza) e la chiave pubblica del prossimo proprietario. L'uso di input multipli corrisponde all'uso di uso di più monete in una transazione in contanti. Una transazione può anche avere uscite multiple, permettendo al proprietario di effettuare più pagamenti contemporaneamente. Un beneficiario può verificare le firme per verificare la catena di proprietà. La somma delle entrate può superare la somma prevista dei pagamenti, ma, come nelle transazioni in contanti, un'uscita aggiuntiva è usata per restituire il resto al pagatore. Le commissioni di transazione sono volontarie da parte di chi paga, e rappresentano solo un incentivo per i minatori.

Note immagine:

- Exchange: trasformazione bitcoin e moneta
- Nodi per registrare transazioni
- Gli utenti che voglio partecipare all'ecosistema devo essere dotati di un wallet:
 - Un wallet è un software che permette di gestire in maniera facile le copie di chiavi private-pubbliche, che servono per ricevere o dare diritti di spesa/transazioni
- Le transazioni vengono salvate sui database distribuiti e vengono replicate su tutti i nodi della rete. Soprattutto per questo non serve una banca a verificare la validità delle transazioni, perché su ogni pc della rete sarà presente la prova della transazioni.

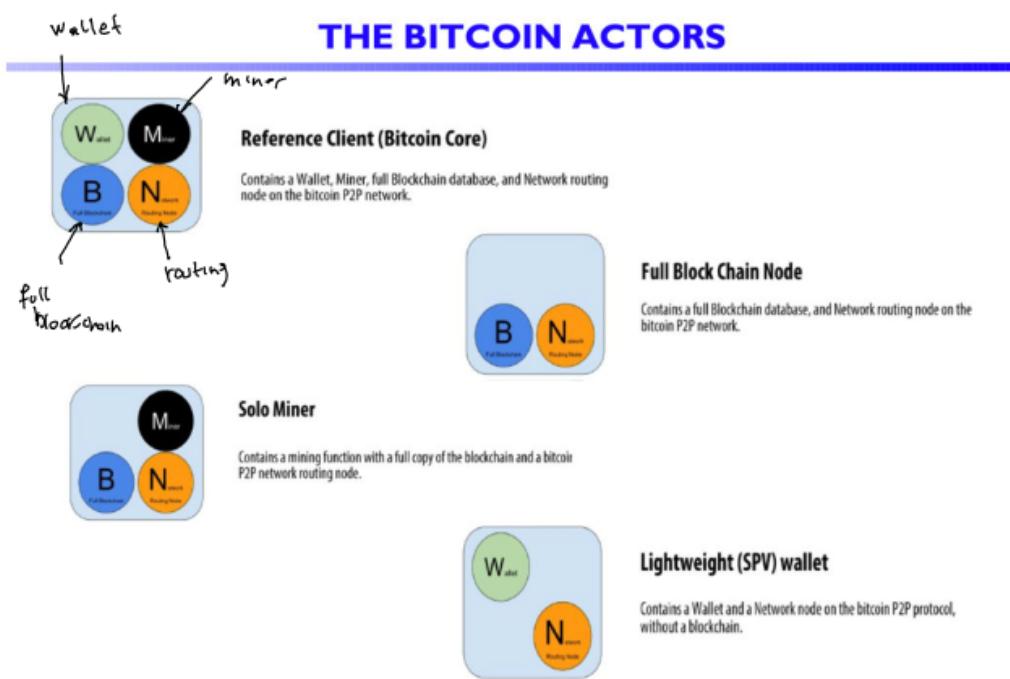


Figura 178: The **Bitcoin** actors

Tipologia di nodi:

- Wallet (verde): mantiene le chiavi pubbliche e private
 - (Quando viene effettuata un'operazione viene messa in chiaro solo la chiave pubblica, ma non gli utenti mittenti e destinari coinvolti)
- Routing (arancione): quando riceve una transazione può fare il broadcast in modo da comunicarlo a tutti

- Full blockchain (blu): tiene il database di tutte le transazioni fatte e/o può creare transazioni
- Miner (nero): in genere le transazioni da inserire su un blocco sono tantissime, il miner ha il compito di selezionarle e metterle in un blocco (nelle blockchain si può inserire solo un blocco di transazioni per volta, e non una transazione alla volta), per poi comunicare le operazioni portate a termine agli altri nodi in modo che tutti aggiornino le informazioni.

Un nodo può essere composto anche da **l'unione** di questi elementi

10.2.2 Rischi di transazioni digitali

Il pagamento richiede l'uso di dispositivi hardware/software

- il barista e il cliente non sono in grado di controllare visivamente la transazione monetaria

Inoltre, la virtualizzazione della moneta introduce il problema della doppia spesa

- il proprietario di una moneta digitale potrebbe farne una copia e cercare di spenderla due (o più) volte

La firma digitale non risolve tutti i problemi, Il problema della doppia spesa rimane!

- un utente può effettuare una transazione se è il proprietario dei bitcoin in ingresso
 - La firma digitale non risolve tutti i problemi, Il problema della doppia spesa rimane!
- Una soluzione possibile
- tracciare e archiviare tutte le transazioni valide effettuate da tutti gli utenti
 - una transazione non sottoposta all'archivio, non può essere considerata valida

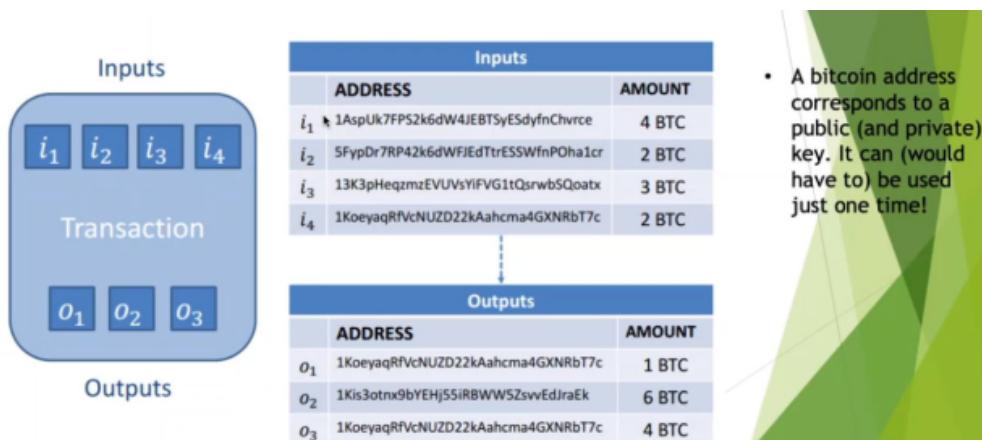


Figura 179: Esempio di transazione

Serie di indirizzi di input (chiavi private) che utilizzo per sbloccare n bitcoin ricevuti in un'unica transazione. Nell'esempio sopra gli 11 bitcoin in input vengono sbloccati da una determinata chiave privata associata ad una chiave pubblica e vengono poi inviati agli indirizzi pubblici riportati in output. Ad ogni transazione i bitcoin in input vengono distrutti e vengono creati quelli in output, in questo modo non ci saranno corrispondenze 1 a 1 e i nuovi non saranno in alcun modo collegati ai vecchi.

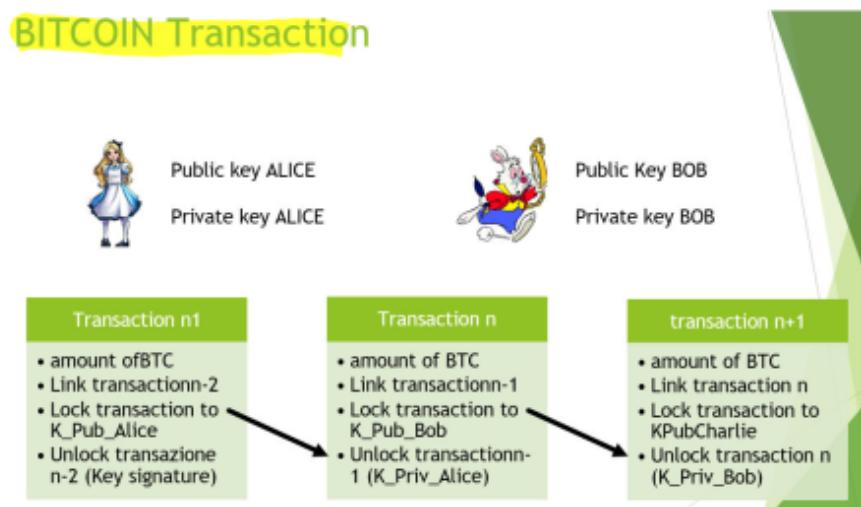


Figura 180: Transazione Bitcoin

Per scambiare dei bitcoin ogni utente deve creare una coppia di chiavi crittografiche (pubblica e privata) e ad esempio: alice che vuole inviare bitcoin a Bob deve formare una cosiddetta transazione bitcoin, nella quale inserire queste 4 informazioni:

1. il numero di bitcoin da inviare
2. indicare al sistema da dove prenderà questi bitcoin inserendo un link ad una transazione che a sua volta ha ricevuto in passato

Poi abbiamo due condizioni crittografiche:

3. lock transaction a favore della chiave pubblica di bob con la quale viene bloccata crittograficamente la transazione a favore della chiave pubblica di Bob indicando di fatto al sistema chi è il nuovo proprietario
4. unlock transaction kprivalice con la quale si da prova al sistema che alice conosce la soluzione alla condizione di lock impostata a suo favore, da qualcun altro, nella transazione $n - 1$.

Bob ora ha la transazione N bloccata crittograficamente a suo favore con la possibilità in futuro di formare la transazione $n+1$ con la quale cederà i bitcoin a favore di un terzo soggetto. Questo è il modo con cui si scambiano le proprietà di bitcoin nel sistema.

Rimane però in sospeso il fatto che non avendo più a disposizione una parte centralizzata come una banca chi verifica che le transazioni sono ben formate per poi salvarle nell'archivio finanziario chiamato blockchain??

Tutte le transazioni non vengono inviate ai miner ma alla rete in un pool generale, poi saranno i miner a decidere quale transazione mettere in un blocco.

Secondo quali regole i miner scelgono queste transazioni? La regola generale dice che si devono inserire nei blocchi prima le transazioni ad importo maggiore e prima le transazioni più vecchie. Ad oggi però le regole sono un po' cambiate vista la mole di transazioni in attesa, infatti se qualcuno vuole può incentivare il miner a prelevare prima la propria transazione. Per incentivare il miner a prelevarla bisognerà lasciare al miner una piccola commissione (o tassa), per esempio 0,5 bitcoin sugli 11 totali dell'esempio fatto in precedenza. Nel protocollo è stabilito che quando i miner inseriscono nuove monete ricevono un compenso, questo compenso è variabile e col tempo verrà decrementato fino ad arrivare a 0 in base al rapporto compenso-numero di blocchi nuovi inseriti, tutto ciò avviene affinchè i miner si concentrino principalmente sulle transazioni (dove riceveranno le commissioni degli utenti) e non sull'inserimento di nuove monete/blocchi. Se così non fosse non sarebbero incentivati e l'ecosistema bitcoin crollerebbe.

2 -Each node add the received transaction to the pool of transactions and select the transactions to put in a block

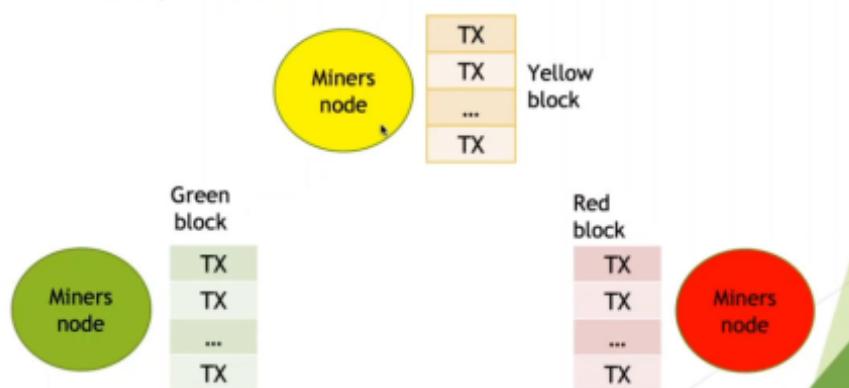


Figura 181: Distributed consensus algorithm

I miner prendono quindi le transazioni dal pool e formano un blocco da inserire nella blockchain. Affinchè tutto questo avvenga in modo sincrono tra i nodi (in maniera tale che a tutti risultino le stesse transazioni prelevate e inserite) c'è bisogno che questi nodi si parlino per capire quali transazioni ogni nodo può prelevare dal pool e inserire nella blockchain. Il protocolli di consenso regolano questo tipo di operazioni, i più utilizzati nell'ambito delle criptovalute sono il proof of work (PoW, bitcoin) e il proof of stack (PoS, ethereum a breve).

- Nel proof of work la ricompensa non viene data a tutti i nodi ma al primo che riesce a risolvere un problema crittografico, questo si aggiudicherà il diritto di scegliere quale transazione inserire nel blocco. Per risolvere il problema il miner dovrà trovare una stringa che aggiunta alle transazioni del blocco dia come risultato 00000xxxxxxxxxxxx una volta hashata. L'unico modo per ottenere la stringa è facendo dei tentativi per indovinarla. Una volta verificata la correttezza della stringa dagli altri nodi l'operazione sarà approvata. La difficoltà per i miner varia in base al numero di 0 iniziali contenuti nella stringa finale, questi 0 possono essere aumentati o diminuiti in base al tempo di risoluzione medio impiegato dai miner (che dovrebbe sempre aggirarsi intorno ai 10 minuti).

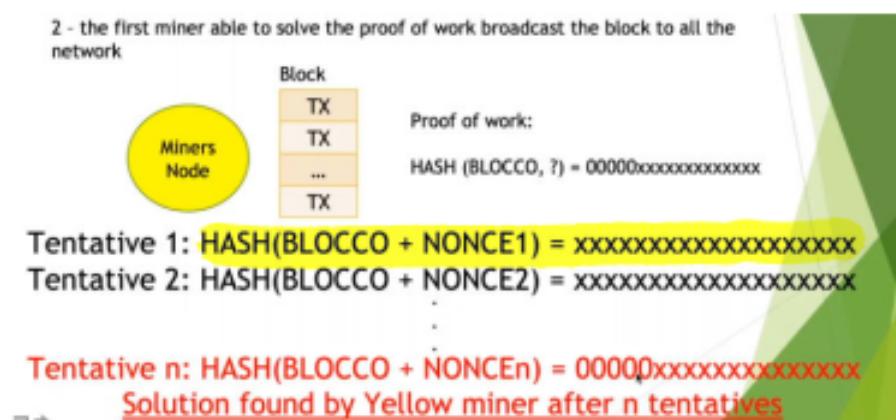


Figura 182: Proof of work

Tra tutte le transazioni inserite il miner aggiunge anche una propria transazione, che viene detta coinbase. A differenza delle normali transazioni contiene solo l'output, ovvero l'indirizzo Bitcoin del minatore che ha eseguito con successo il mining del nuovo blocco (e non input e output), questi tipi di transazioni fanno parte del sistema per mettere in circolazione nuove monete che non sono mai state spese. È grazie a questi tipi di transazioni che l'ecosistema Bitcoin ha iniziato ad avere criptovalute per effettuare pagamenti e scambi di valore.

Le transazioni Coinbase sono anche note come generazione di transazioni. Queste sono una parte fondamentale della generazione delle monete Bitcoin, poiché sono quelle che danno origine a queste nuove monete. Cioè, ogni transazione coinbase è responsabile della trasmissione delle monete vergini al minatore che ha risolto il blocco. In questo modo, il valore base totale di una transazione coinbase, contiene solo ed esclusivamente nuove monete che non sono mai state nel file blockchain

10.2.3 Coinbase

Caratteristiche coinbase

Quando un nuovo blocco viene generato sulla blockchain, ha un elenco di transazioni verificate al suo interno. Ciascuna di queste transazioni è stata generata dagli utenti di criptovaluta di detta blockchain. Ma nonostante, la prima di queste transazioni corrisponde alla transazione coinbase. Il valore di base di questa transazione è equivalente a quello della ricompensa attualmente attiva per il mining di quel blocco. Ciò significa che il valore di questa transazione è legato alla ricompensa del blocco corrente e risente del dimezzamento attivo in quel momento per quella criptovaluta.

Hash	Value	Date
31a5f43e89368606a6eb3c702ddf5ba02365df1fd503d...	6.25000000 BTC	2021-04-30 07:58
COINBASE (Newly Generated Coins)		
	12dRugNcdxK39288N...DV4GX7rMs... 6.52864895 BTC	
	OP_RETURN 0.00000000 BTC	
	OP_RETURN 0.00000000 BTC	
	OP_RETURN 0.00000000 BTC	
Fee	0.00000000 BTC (0.000 sat/B - 0.000 sat/WU - 351 bytes) (0.000 sat/vByte - 324 virtual bytes)	6.52864895 BTC 4 Confirmations
Hash	d691863e958d0633c170ea6d89e64fd5c1a21e3311247...	2021-04-30 07:57
39VgGLvtWEBFnTYHYoUxDHFWyW... 0.11425883 BTC	1J3oViJ43NEwxhvMotTTLJCJHzDStF... 0.01580614 BTC	
	3Q3PqCZE2z79uC2TVLSSok6oTwx... 0.09778069 BTC	
Fee	0.000067200 BTC	0.00000000 BTC

Figura 183: Esempio transazioni: La prima coinbase, la seconda normale

4 - All the node (verify and) accept the block and add the block at the end of the blockchain

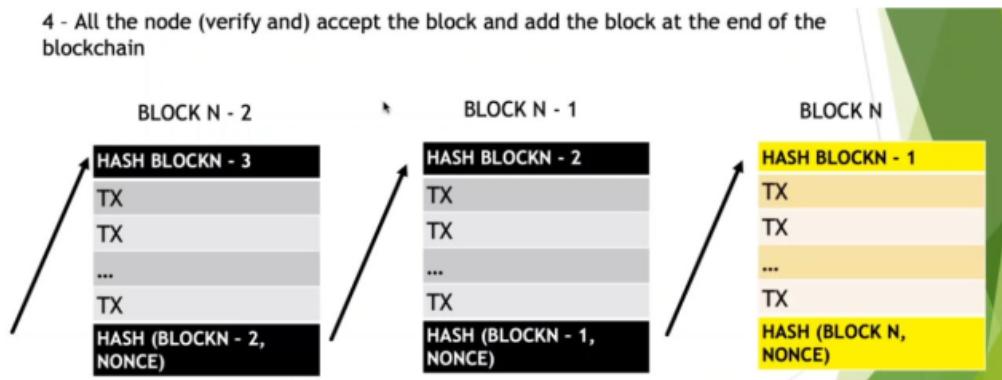
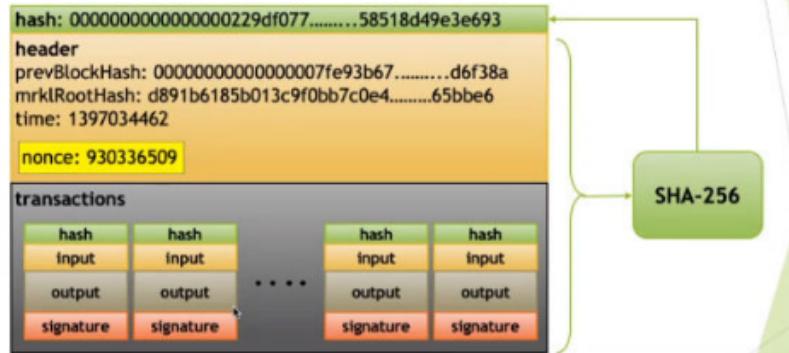


Figura 184: Verifica da parte degli altri nodi

Block (acceptable) of transaction



hash (256 bit) of a block has to start with 64 bit set to 0
⇒ entered the **nonce** field in the header
⇒ at each attempt the **nonce** is increased by one unit

Figura 185: Esempio blocco di transazione

Visualization of a blockchain fork event: a new block extends one fork



Figura 186: Esempio partizionamento della rete

Se un'identica transazione venisse inserita e gestita prima da una parte della rete (nell'esempio rossa o verde) si verrebbero a creare due partizioni della rete non sincronizzate con problemi di integrità e consistenza. Questi problemi si risolvono con una semplice regola: la blockchain a cui fare riferimento è sempre quella più lunga, nel caso dell'esempio il blocco rosso viene scartato e si prosegue sull'altra via, quella viola (blocco x). Queste situazioni durano in media 10 minuti, ovvero il tempo di inserimento/aggiornamento tipico della catena blockchain. Potrebbe in rari casi durare al massimo un'ora, perciò un blocco viene considerato inserito al 100 (quindi non potrà più

essere cancellato) solo quando si trova al livello 6-7 della catena (dopo 6-7 aggiornamenti da 10 minuti).

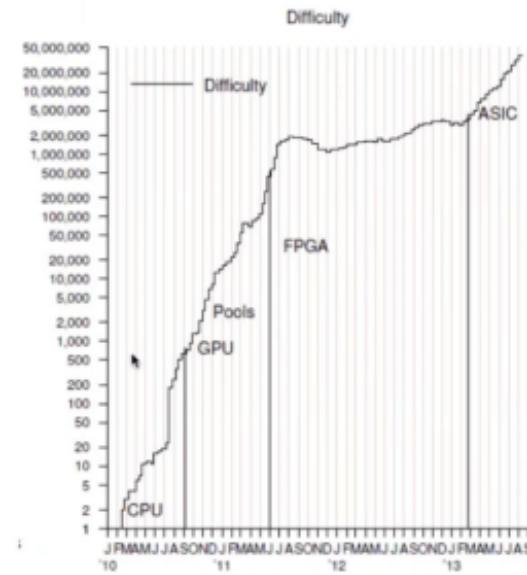


Figura 187: Mining hardware difficulty

Negli anni la difficoltà di risolvere i problemi e minare è sempre aumentata, questo perché la competitività e i requisiti hardware dei miner sono aumentati (e continueranno a farlo) con l'aumentare dei prezzi dei bitcoin. I miner hanno la possibilità di “unire le forze” con altri miner (Pools), in modo tale da compensare o aumentare la potenza di calcolo, in tal caso le ricompense saranno distribuite in base al contributo dato. Gli ASIC sono architetture create apposta per calcolare gli hash. La tecnologia FPGA (Field Programmable Gate Array) è stata praticamente sostituita dall'ASIC.

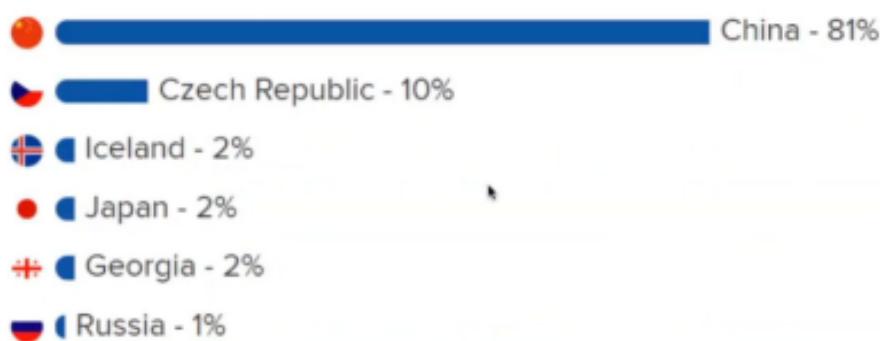


Figura 188: Distribuzione farm mining

La distribuzione della farm è fortemente condizionata dal costo della corrente, per questo motivo l'81farm/miner è situata in Cina. La distribuzione delle potenze di calcolo o dell'ambiente distribuito va tenuta sempre sotto controllo a livello globale, in modo tale da distribuire equamente la potenza e il potere ai vari nodi della rete (che appunto deve mantenere l'equilibrio di una rete distribuita).

10.2.4 Dettagli su come si sbloccano le transazioni

Creation of keys/addresses

- Starting with a private key

32 byte (256 binary digits shown as 64 hexadecimal digits, each 4 bits (mezzo byte))
1E99423A4ED27608A15A2616A280E9E52CED330AC530EDCC32C8FFC6A526AE~~DD~~

Figura 189: Struttura chiave privata

Una chiave privata è composta da 32 byte. A partire dalla chiave privata tramite curve ellittiche si calcola la chiave pubblica, dalla chiave pubblica con la funzione di hash si utilizza un indirizzo bitcoin.

Creation of keys/addresses

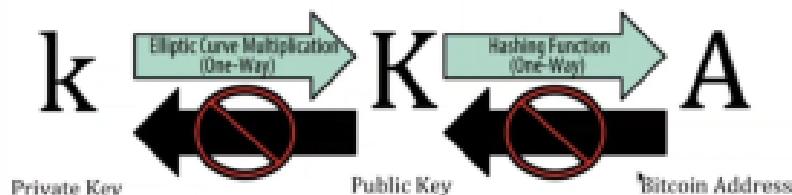


Figura 190: Creazione chiavi / indirizzi

A partire dalla chiave privata per ottenere la chiave pubblica si moltiplica la chiave privata un tot di volte fino ad ottenere la chiave pubblica.

Creation of keys/addresses

- We obtain a public key
Multiply the private key k with the generator point G to find the public key K .
 $K = 1E99423A4ED27608A15A2616A2B0E9E52CED330AC530EDCC32C8FFC6A526AEDD * G$
 Public Key K defined as a point $K = (x, y)$.

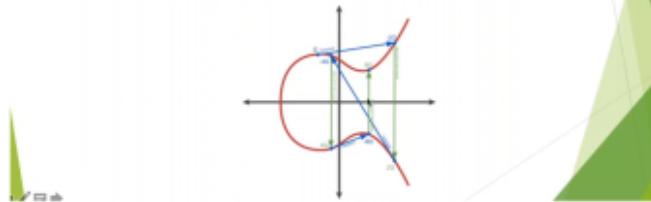


Figura 191: Da chiave privata a chiave pubblica tramite curve ellittiche

- Public key with several representation

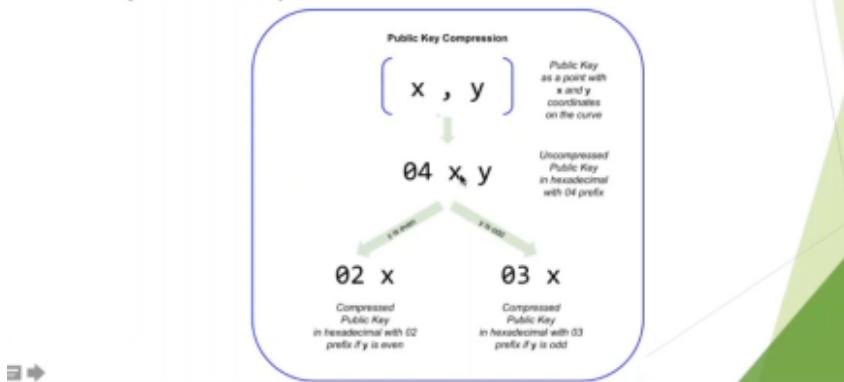


Figura 192: Le 3 possibili rappresentazioni della chiave pubblica

Da questa chiave pubblica con una funzione di hashing ottengo l'indirizzo bitcoin da inserire nelle transazioni.

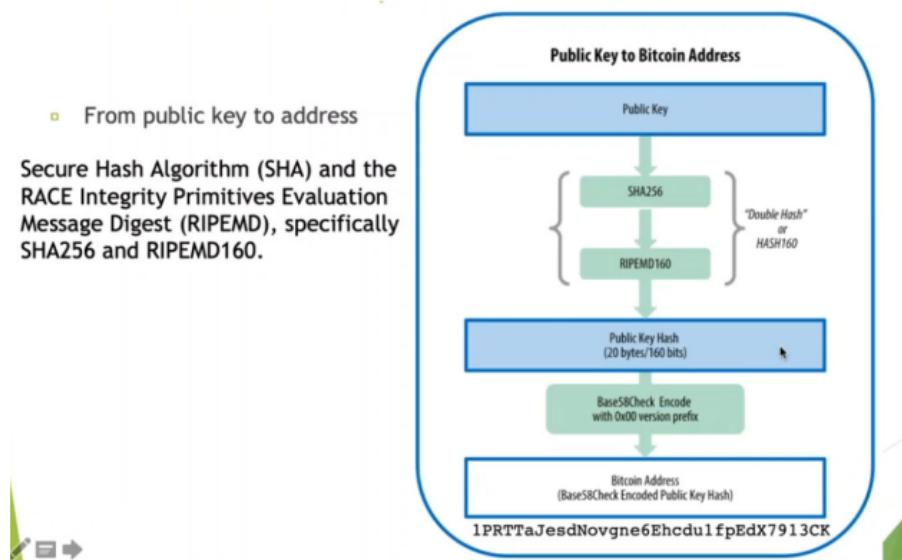


Figura 193: Dalla chiave pubblica all'indirizzo

Create a bitcoin address: 1 and bc1

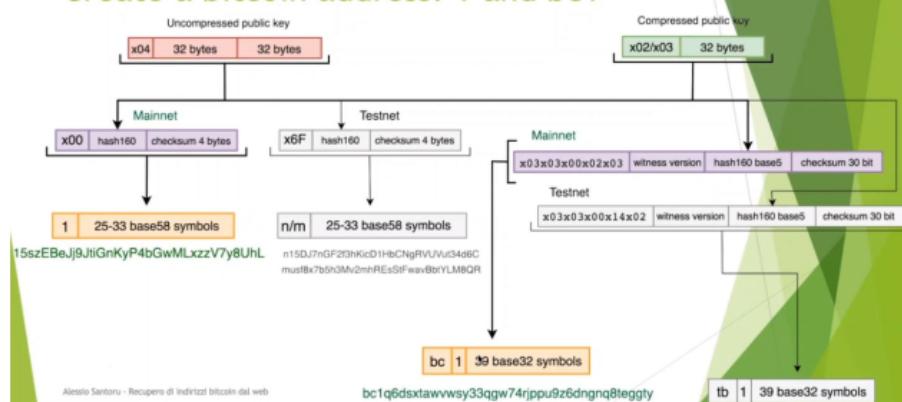


Figura 194: Struttura reti bitcoin

La struttura delle reti bitcoin comprende anche una “Testnet” dove effettuare delle prove locali che non verranno applicate alla rete. Negli anni è cambiata la maniera di calcolare i bitcoin, quella più utilizzata in questo momento è “bc | 1 | 39 base32 symbols”.

Decimal prefix	Hex	Example use	Leading symbol(s)	Example
0	00	Pubkey hash (P2PKH address)	1	17V2NX15N5NtKa8UQFwQbFeFc3iqRYhem
5	05	Script hash (P2SH address)	3	3EktnHQD7RIA6uzMj22IFT9YgRrkSgzQX
128	80	Private key (WIF , uncompressed pubkey)	5	5Hwgr3u458GLafKBgxssHSPqjnYoGrSzgQsPwLFhLNyskDPyyA
128	80	Private key (WIF, compressed pubkey)	K or L	L1aW4aubDFB7yfras251mN3bqg9nwySY8nkoLmjebSLDSBWv3ENZ
4 136 178 30	0488B21E	BIP32 pubkey	xpub	xpub661MyMwAqRbcEYS8w7XLSVeEsBXy79zSzH1J8vCdxAZningWLdn3gtU6LbpB85b3D2yc8fvZU521AAwdZafEz7mnzBBsz4wKY5e4cp9LB
4 136 173 228	0488ADE4	BIP32 private key	xprv	xprv9s21ZrQH143K24Mfq5zL5MhWK9hUhGbd4ShLXo2Pq2oqzMMo63oStZfF93Y5wvdJayhgkkFocicQzcP3y52uPPxfnfoLZB21Teqt1VvEHx
		Bech32 pubkey hash or script hash	bc1	bc1qw508d6qejxtdg4y5r3zarvary0c5xw7kv8f3t4

Figura 195: Tipi di indirizzo

Una transazione può essere sbloccata in 5 modi diversi:

1. Ti pago se tu mi dimostri di avere una determinata chiave privata corrispondente ad una chiave pubblica
2. Ti pago se dimostri di avere questa chiave pubblica
3. Transazioni che non possono essere mai sbloccate, i bitcoin vengono persi
4. Vengono spesso utilizzate per certi tipi di contratti in bitcoin, posso richiedere che la transazione venga sbloccata solo se vengono (per esempio) fornite 2-3 chiavi su 5 che sto richiedendo
5. Io faccio un script hash e ti pago se tu dimostri di conoscere qual è la stringa il cui hash ti da quello che dico io

Transaction types

- Pay to Public Key Hash (P2PKH) and P2WPKH (Pay to Witness Public Key Hash)
- Pay to Public Key (P2PK)
- Data Output (OP RETURN)
- MultiSig
- Pay to Script Hash (P2SH) and P2WSH (Pay to Witness Script Hash)

10.2.5 Pay to Public Key Hash (P2PKH)

Lo sblocco di una transazione inviata ad una chiave pubblica non si fa mostrando e rivelando la chiave privata ma si fa firmando questa transazione con la chiave privata, mostrando poi questa firma al sistema. Infine il sistema verifica se la firma è corretta applicando la chiave pubblica a questa firma per vedere se ottiene questa transazione.

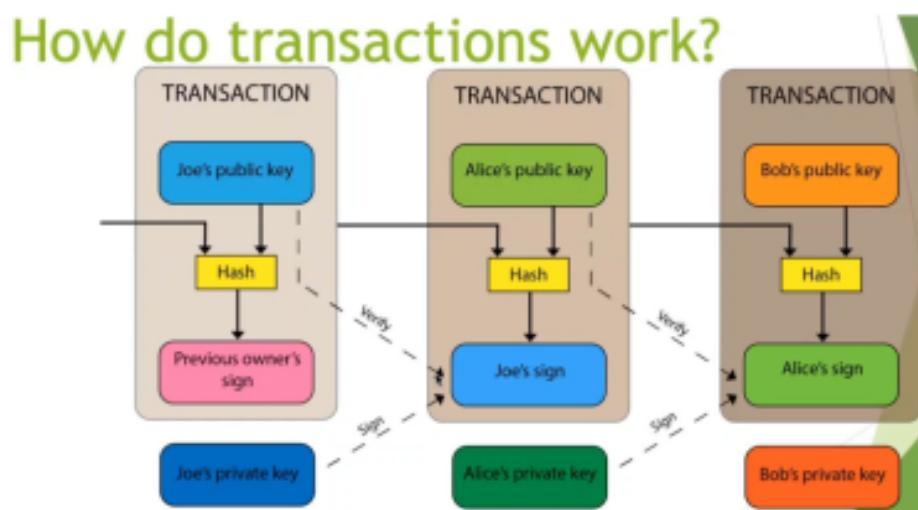


Figura 197: Flusso P2PK e P2PKH

The screenshot shows the Blockchain.com Explorer interface. At the top, there are tabs for Wallet, Exchange, and Explorer, along with buttons for Buy Bitcoin and Trade. Below the tabs, the transaction details are displayed. The transaction includes a Sigscript and a Witness section. The outputs section shows a single output with an index of 0, an address of 17917L8uNxR3tDmwv8CtKCRyMycnWF1r1q, a value of 0.00864183 BTC, and a Pkscript. The Pkscript contains several OP codes and data bytes. A blue circular icon with a white 'B' is visible on the right.

Figura 198: Esempio "traduzione" da Sigscript a Pkscript

- [D1525aa6ad2b9a63caf974921f8524e7b7ce6dfa655f072207882ac7ad3cbf94](#)
 - Is this
 - 0200000001990ec0ae3dd81f945dfa07cb9f78b8453132747c0bbde2b8d4c03f4b97360011010000006b4830450221009fc57570d4ed1ea877e90746020835555d6a5e16633a83bb6000e6c7c754f74c02206ac7b8958544b13eed4d1c878f4e616d84fa792da23c7c1013feea8791f4ba7c0102dd0a816d257b36e335e218d57110d42617f1d01a32d3286779b4a6d3b1b101413b1b10141fefffff02b72f0d000000000001976a91443542c6a7bc41f188dfa92815b64a07ecc88937c88acec110100000000001976a914eb67f69b77058ce33729389b7cd40d9642401ef988accdd90700

Figura 199: Transazione [Bitcoin](#), verde script in input e rosso script in output

How do transactions work?

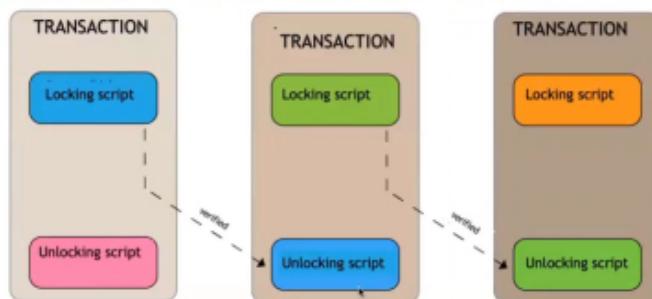


Figura 200: Locking e Unlocking script

Come fa una transazione con un determinato input a sbloccare un determinato output? Si

deve intrepretare l'input a coppie esadecimale come delle istruzioni assembler.

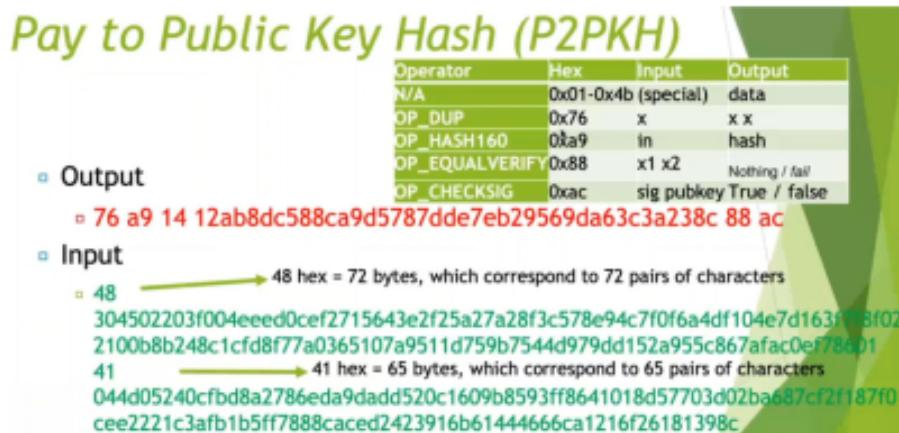


Figura 201: Da esadecimale ad assembler

Pay to Public Key Hash (P2PKH)

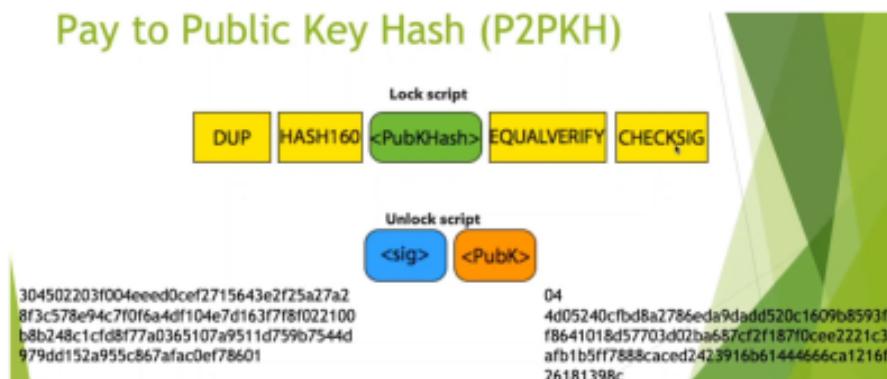


Figura 202: Struttura "programma" P2PKH

Se concateniamo l'unlock script al lock script (output e input) otteniamo un programma con il flusso di istruzioni riportato nell'immagine precedente. Il linguaggio di scripting utilizzato è chiamato bitcoin script.

E' non touring complete perché non ha i cicli, utile come meccanismo di difesa (ddos, brute force). Ethereum li ha ma ha un meccanismo per limitarli e non renderli infiniti.

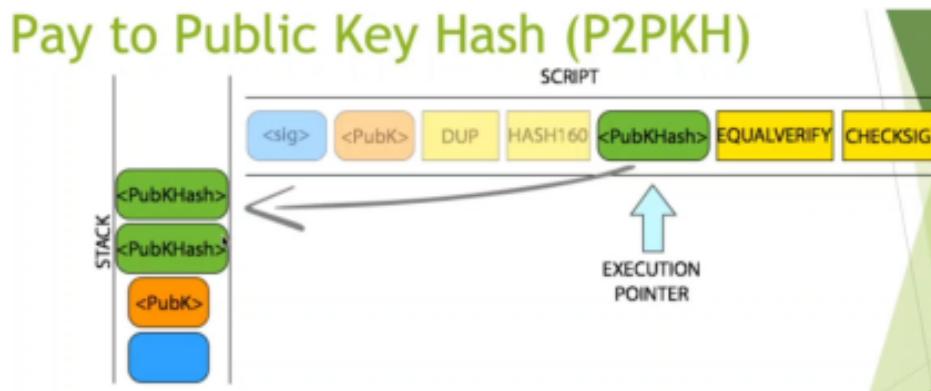


Figura 203: Flusso programma assembler P2PKH

1. Metto nello stack i 48 hex (72 bytes dell'input)
2. Inserisco e duplico la chiave pubblica
3. Effettuo l'hash sulla chiave pubblica
4. Metto sullo stack la chiave pubblica
5. Eseguo equal verify per vedere se la chiave pubblica è uguale alla chiave pubblica duplicata
6. Eseguo equal verify per vedere se la chiave pubblica corrisponde alla signature
7. Se è tutto true la transazione verrà sbloccata e completata

chainquery.com -> dato un id di una transazione restituisce la transazione

Pay-to-Public-Key(P2PK)

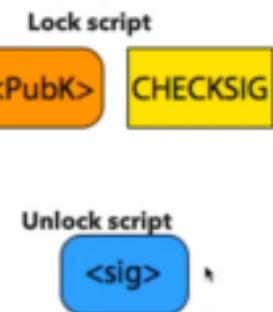


Figura 204: Struttura programma P2PK

La P2PKH è la transazione più utilizzata, è preferita alla P2PK perché occupa meno spazio nella transazione. Lo spazio nel blocco è fisso, quindi utilizzando una P2PKH e non una P2PK un blocco potrebbe contenere un numero maggiore di transazioni.

10.2.6 M-of-N Multi-signature (multisig)

Il pagamento avviene a condizione di fornire n signature su un tot di m chiavi pubbliche fornite.

M-OF-N MULTI-SIGNATURE (MULTISIG)

- a m-of-n multisig (multisignature) locking script
 - specifies **n** public keys
 - requires a valid unlocking script to provide **m** signatures created using any **m out of the n private keys** corresponding to these public keys.
- the m-of-n challenge script is of the form
$$m \text{ <Public Key 1>} \dots \text{ <Public Key n>} n \text{ OP_CHECKMULTISIG}$$
- a valid unlock script pushes m signatures onto the stack and is of the form
$$\text{OP}_0 \text{ <Signature 1>} \text{ <Signature m>}$$
- CHECKMULTISIG checks that the signatures provided by the response script are valid.

Figura 205: M-OF-N MULTI-SIGNATURE (MULTISIG)

Multi-Signature

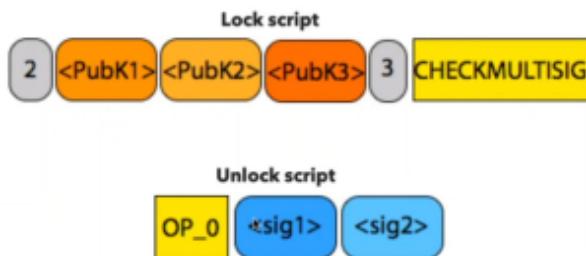


Figura 206: Struttura programma Multi-Signature

The screenshot shows a detailed view of a Bitcoin transaction on the Blockchain.com website. At the top, there are navigation links for 'Blockchain.com', 'Wallet', 'Exchange', and 'Explorer'. On the right, there are buttons for 'Buy Bitcoin' and 'Trade'. Below the header, a search bar says 'Search your transaction, an address or a block' with a dropdown set to 'USD'. A summary section is titled 'Summary' with a link to 'Raw Transaction'. The main content area displays the transaction details:

Hash	1AB.JiWcwvp7tAopUkSnGuEYHmzGYfZPiq	1AB.JiWcwvp7tAopUkSnGuEYHmzGYfZPiq	1AB.JiWcwvp7tAopUkSnGuEYHmzGYfZPiq	Fee	0.00500000 BTC (658.762 sat/B - 164.690 sat/WU - 759 bytes)	0.00500000 BTC	0.47952304 BTC	0.47952304 BTC	
	0.00500000 BTC	0.00500000 BTC	0.47452304 BTC				Unable to decode output address	1KET1UAB76CphMfukC28uUstsUduexvsSB	0.46452304 BTC
							I	1AB.JiWcwvp7tAopUkSnGuEYHmzGYfZPiq	0.00500000 BTC

Below the table, a note states: "This transaction was first broadcast to the Bitcoin network on January 30, 2012 at 2:58 AM GMT+1. The transaction currently has 517,892 confirmations on the network. At the time of this transaction, 0.47952304 BTC was sent with a value of \$2.74. The current value of this transaction is now \$26,935.34. Learn more about how transactions work."

Figura 207: Transazione Multi-Signature [Bitcoin.com](#)

La parte di transazione evidenziata è multi-signature perché non si riesce a risalire agli indirizzi di output. Si verrà a conoscenza dell'indirizzo solo quando verrà sbloccato, ovvero quando 2 indirizzi su 3 verranno sbloccati e il denaro verrà inviato a quei 2 indirizzi, nel momento iniziale in cui inserisce la transazione non si è a conoscenza degli indirizzi di output. Uso delle multi-signature:

- 1-of-n: qualunque utente (che fa parte degli n utenti) può spendere il denaro ma è importante tenere traccia di chi l'ha fatto
- 2-of-2: solo se entrambe le parti coinvolte è d'accordo con la spessa la transazione viene sbloccata
- 2-of-3: contratti di vendita garantiti da una terza parte fidata, quindi 3 chiavi pubbliche coinvolte. Anche detti escrow contract (in italiano deposito di garanzia).
Es. Se A deve inviare soldi a B, C tiene i soldi di A e non li da a B finché B non da ad A quello che gli spetta, in questo caso scambio di soldi e signature.

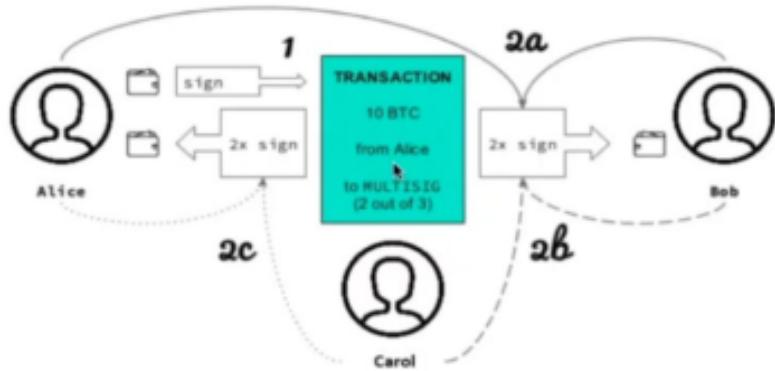
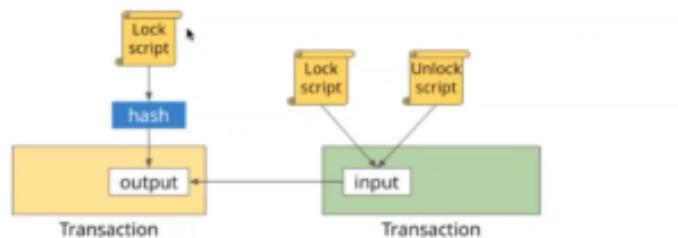


Figura 208: Schema di esempio multi-signature

Esistono anche le transazioni con il lock time, dove A paga B dopo solo dopo un tempo T.

La multi-signature permette di fare una prima forma di smart contact, però occupando tanti "posti" (chiavi numerose) non viene preferita, un'alternativa è fare l'hash dello script in modo tale che occupi meno spazio (Pay to Script Hash, P2SH).

PAY TO SCRIPT HASH (2SH)



- a solution for multi-sig scripts (and, generally complex bitcoin scripts)
 - the sender specifies just a hash of the script
 - include the hash of the script, instead of the script, in the locking script.
- to unlock and redeem the transaction, the receiver presents
 - the original script
 - the signatures, in the unlock script

Figura 209: Pay to Script Hash P2SH

Nella transazione di output che blocca inserisco uno script di Lock e poi faccio l'hash, dalla parte dell'input invece viene fatto il lock script per verificare che le stringhe hash generate dai due Lock Script corrispondano e poi viene eseguito lo script di unlock se tutto da TRUE.

Pay to Script Hash (P2SH)

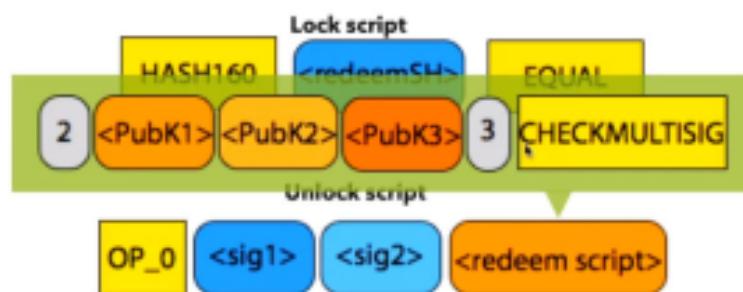


Figura 210: Struttura Pay to Script Hash

Di solito nello script di redeem (riscatto) c'è una P2PK .

10.2.7 Data Output (OP Return)

Vengono utilizzate normalmente per scrivere in blockchain ed avere garanzie di notarizzazione.

Ogni elemento scritto di blockchain ha una data e orario, quindi hashato sarà univoco

Come funziona la OP Return? Si mette un indirizzo a caso di 20 byte che è un pagamento fake che non può essere mai speso e si inserisce una parte di dato che corrisponde per esempio all'hash del programma/transazione di cui voglio dimostrare l'esistenza.

`OP_RETURN <data>` dove “data” sono i dati che io voglio dimostrare.

Data Output (OP RETURN)

- Output

- 6a 20 d68bdab455902dcc59f4e8f775a59c58ea8ae8f0a6cb7f3b96f8a3cf84c9af7

20 hex = 32 bytes, which correspond to 32 pairs of characters

Operator	Hex	Input	Output
N/A	0x01-0x4b (special)	data	
OP_RETURN	0x6a	Nothing	fail

Figura 211: Data Output (OP Return)

Data Output (OP RETURN)



Figura 212: Struttura Data Output (OP Return)

|| Nelle transazioni OP_Return possono essere inserite delle immagini

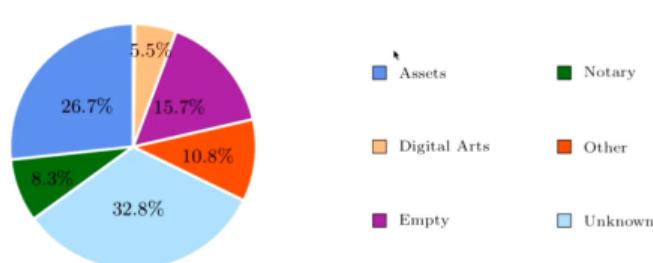


Figura 213: Grafico contenuti OP_Return

Pay to Witness Public Key Hash P2PKH | Pay to Witness Script Hash P2WSH

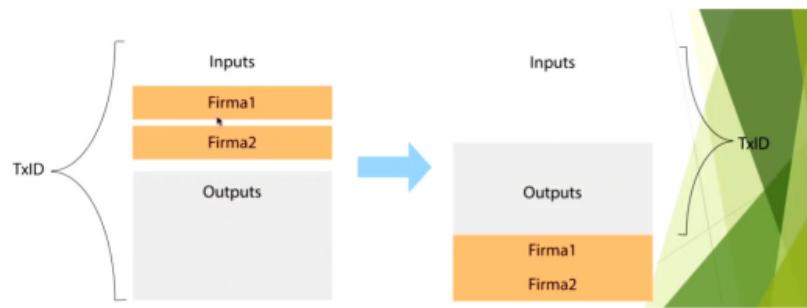


Figura 214: Transazione classica -> Transazione con Pay to Witness

Se le firme di una transazione vengono scritte in maniera diversa (ci sono più modi per scriverle) l'hash generato sarà ovviamente diverso anche se la transazione è la stessa, questa problematica viene chiamata transaction malleability.

Per evitare questo problema si è deciso di spostare le firme fuori dalle transazioni da firmare/hashare. L'altra modifica che viene fatta è togliere gli script di input e di output, perché si sa qual è l'operazione da fare per controllare la firma, Quindi perché viene adottata? Risparmio di spazio e difesa da attacchi/problematiche di transaction malleability.

P2WPKH (Pay to Witness Public Key Hash)

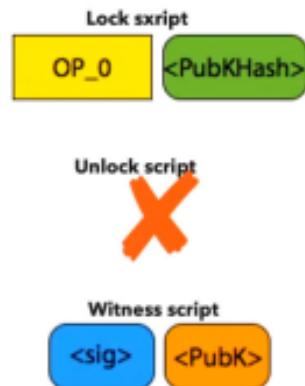


Figura 215: Struttura programma P2WPKH

Pay to witness Script Hash (P2WSH)

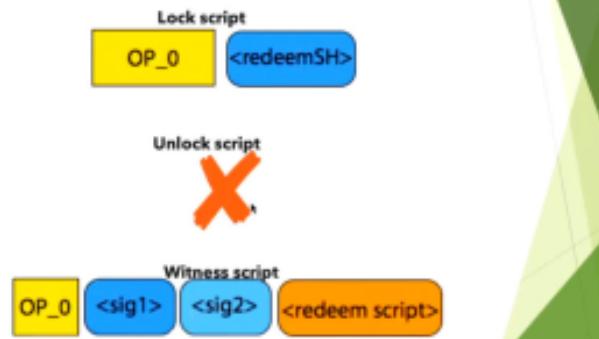


Figura 216: Struttura programma P2WSH

I 5 tipo di transazione appena descritte vengono chiamate “transazioni standard”. Dal momento che uno script bitcoin potrebbe contenere qualsiasi cosa, i miner posso anche decidere di minare “non standard”.

10.2.8 Contenuti tipici transazioni non standard

OnlyHash



Figura 217: OnlyHash

Lock script fatti solo di hash, in tal caso si può definire "Pay to Hash". Es. per bloccarlo c'è l'hash di qualcosa e per sbloccarlo basta dare qualcosa il cui hash fa questo, che appunto non deve essere per forza uno script.

P2Pool Bug



Figura 218: P2Pool Bug

Questo tipo di transazione non può essere sbloccata in alcun modo, ciò si traduce in "soldi buttati".

CHECKLOCKTIMEVERIFY OP DROP (CLTV)

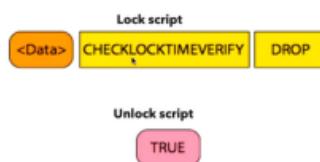


Figura 219: CHECKLOCKTIMEVERIFY OP DROP (CLTV)

Si sblocca con il valore true ma solo dopo che è passato un certo tempo. Es. do un tot di bitcoin solo a chi inserisce il true entro un tot di tempo.

Pay to Public Key Hash 0 (P2PKH0)

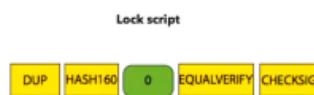


Figura 220: Pay to Public Key Hash 0 (P2PKH0)

Fornire l'unlock il cui hash da 0, soldi buttati anche in questo caso.

OP MIN OP EQUAL

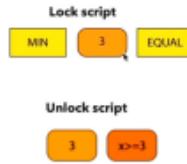


Figura 221: Op Min Op Equal

Operazioni matematiche. Es. trovare l'elemento tale che $6+n$ fa 8.

UnLocked(UL)



Figura 222: UnLocked (UL)

E' sufficiente inserire semplicemente true per sbloccarle.

P2PKH NOP

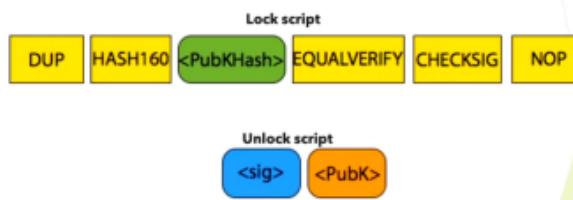


Figura 223: P2PKH NOP

E' a tutti gli effetti una P2PKH. La differenza con le classiche P2PKH è che al termine dello script è presente l'istruzione assembly "NOP", che però non fa niente ed è quindi inutile.

Pay to Hash(P2H)

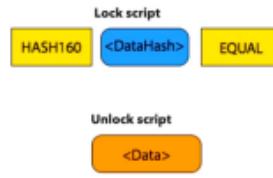


Figura 224: Pay to Hash (P2H)

Pago a condizione di fornire un determinato dato. Nel lock script rispetto alla P2PKH ha DataHash e non <redeemSH>.

| | Nel 2018 le blockchain erano formate dal 93% di transazioni standard e dal 7% di transazioni non standard.

Distribution of Standard transactions

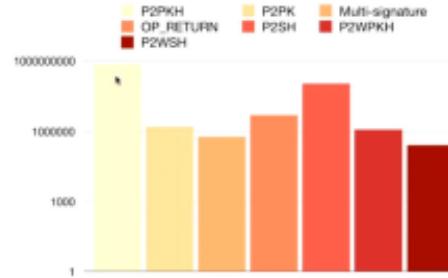


Figura 225: Utilizzo transazioni standard (2018)

Distribution of NON Standard transactions

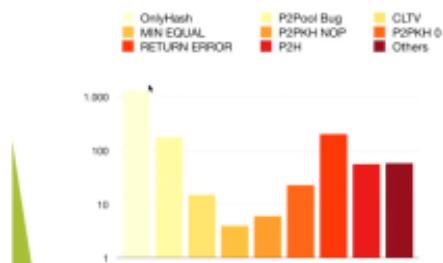


Figura 226: Utilizzo transazioni standard (2018)

MULTISIGNATURES

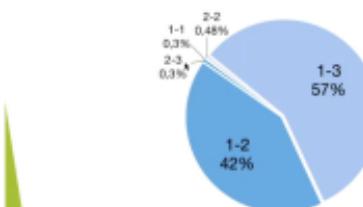


Figura 227: Utilizzo transazioni multisig 2018

Mining time of non-standard transactions

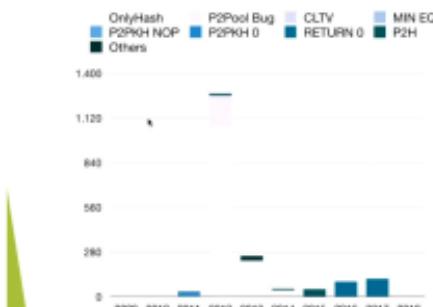


Figura 228: Numero di transazioni non standardminate negli anni

Come si fa a capire chi ha minato una transazione?

682367	6 minutes	Poolin	1,330,053 bytes	3b09587b034f1a6874aa...	10:31	0.00654830 BTC	\$367.83
682366	9 minutes	Unknown	1,177,658 bytes	133113ce94dc99931db...	10:31	0.01473000 BTC	\$827.40
682365	14 minutes	F2Pool	1,244,810 bytes	1ff7ea8c221488a7af6a8a...	10:31	0.00507393 BTC	\$285.01
682364	36 minutes	Poolin	1,854,689 bytes	8a3908926e66d119412...	10:31	0.02346059 BTC	\$1,317.81
682363	39 minutes	Unknown	1,796,671 bytes	25f56984a566d9de45b...	10:31	0.00330141 BTC	\$185.44
682362	40 minutes	F2Pool	376 bytes	74ae068973cbfa126546...	10:31	3.41919717 BTC	\$192,060.07

Figura 229: Miner transazioni Blockchain.com

Command result: bitcoin-cli getrawtransaction

Figura 230: Miner transazione in dettaglio

Avendo a disposizione solo il file json per capire chi ha minato una transazione, possiamo risalire al miner convertendo la stringa selezionata nella figura precedente da esadecimale ad ASCII.

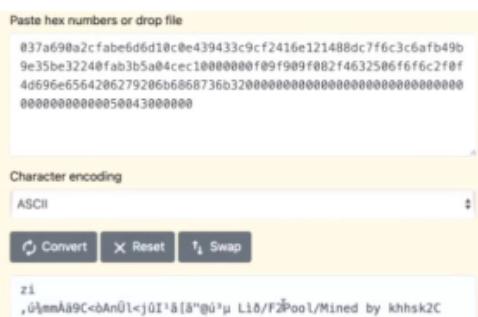


Figura 231: Hex to ASCII

|| Mastering bitcoin, libro per imparare i bitcoin

|| Blockchain-bitcoin possibile argomento di tesi

10.2.9 Sicurezza della blockchain

Ransom attack: sfrutta il fatto che bitcoin è pseudoanonimo e non completamente anonimo. (non si sa a chi è associata una chiave pubblica) (Attacco + riscatto)

51% attack: sfrutta problema dovuto al protocollo di consenso (es. proof of work PoW)

CryptoCurrencies (Bitcoin) weakness

- Double Spending
- Mining attack
 - 51%, Block withholding, Bribery
- Wallet attack (endpoint security)
- Network attack
 - Partitioning, Sybil, Eclipse
- Code Vulnerability
- Data Protection

Figura 232: Panoramica vulnerabilità

Common Vulnerabilities and Exposures

CVE	Announced	Affects	Severity	Attack Is..	Raw	Net
Pre-BIP protocol changes	n/a	All Bitcoin clients	NetSpill ^[2]	Implicit ^[2]	Various hardforks and softforks	100%
CVE-2010-5137	2010-07-28	wwwBitcoin and bitcoind	DoS ^[2]	Easy	OP_LSHFT crash	100%
CVE-2010-5141	2010-07-28	wwwBitcoin and bitcoind	Theft ^[4]	Easy	OP_RETURN could be used to spend any output.	100%
CVE-2010-5138	2010-07-29	wwwBitcoin and bitcoind	DoS ^[2]	Easy	Unlimited SigOp DoS	100%
CVE-2010-5139	2010-08-15	wwwBitcoin and bitcoind	Inflation ^[3]	Easy	Combined output overflow	100%
CVE-2010-5140	2010-08-29	wwwBitcoin and bitcoind	DoS ^[2]	Easy	Never confirming transactions	100%
CVE-2011-4447	2011-11-11	wwwBitcoin and bitcoind	Exposure ^[5]	Easy	Wallet non-encryption	100% ↗
CVE-2012-1909	2012-02-07	Bitcoin protocol and all clients	NetSpill ^[2]	Very hard	Transaction overwriting	100% ↗
CVE-2012-1910	2012-03-17	bitcoind & Bitcoin-Qt for Windows	Unknown ^[1]	Very	Non-thread safe MingW exceptions	100% ↗
BIP-0016	2012-04-01	All Bitcoin clients	Fake Conf ^[6]	Miners ^[2]	Softfork: P2SSH	100% ↗
CVE-2012-2459	2012-05-14	bitcoind and Bitcoin-Qt	NetSpill ^[2]	Easy	Block hash collision (via merkle root)	100%
CVE-2012-3789	2012-06-26	bitcoind and Bitcoin-Qt	DoS ^[2]	Easy	(Lack of) orphan tx resource limits	100%
CVE-2012-4682		bitcoind and Bitcoin-Qt	DoS ^[2]			100% ↗
CVE-2012-4683	2012-08-23	bitcoind and Bitcoin-Qt	DoS ^[2]	Easy	Targeted DoS by CPU exhaustion using alerts	100% ↗
CVE-2012-4684	2012-08-24	bitcoind and Bitcoin-Qt	DoS ^[2]	Easy	Network-wide DoS using malleable signatures in alerts	100% ↗
CVE-2013-2272	2013-01-11	bitcoind and Bitcoin-Qt	Exposure ^[6]	Easy	Remote discovery of node's wallet addresses	100% ↗
CVE-2013-2273	2013-01-30	bitcoind and Bitcoin-Qt	Exposure ^[6]	Easy	Predictable change output	100% ↗
CVE-2013-2292	2013-01-30	bitcoind and Bitcoin-Qt	DoS ^[2]	Very	A transaction that takes at least 3 minutes to verify	100% ↗
CVE-2013-2293	2013-02-14	bitcoind and Bitcoin-Qt	DoS ^[2]	Easy	Continuous hard disk seek	100% ↗
CVE-2013-3219	2013-03-11	bitcoind and Bitcoin-Qt 0.8.0	Fake Conf ^[6]	Miners ^[2]	Unenforced block protocol rule	100% ↗
CVE-2013-3220	2013-03-11	bitcoind and Bitcoin-Qt	NetSpill ^[2]	Easy	Inconsistent BDB lock limit interactions	100% ↗
BIP-0034	2013-03-25	All Bitcoin clients	Fake Conf ^[6]	Miners ^[2]	Softfork: Height in coinbase	100% ↗
BIP-0050	2013-05-15	All Bitcoin clients	NetSpill ^[2]	Implicit ^[2]	Hard fork to remove txid limit protocol rule	100% ↗
CVE-2013-4627	2013-06-17	bitcoind and Bitcoin-Qt	DoS ^[2]	Easy	Memory exhaustion with excess tx message data	100% ↗

Figura 233: Common Vulnerabilities and Exposures

Friend (to protect)

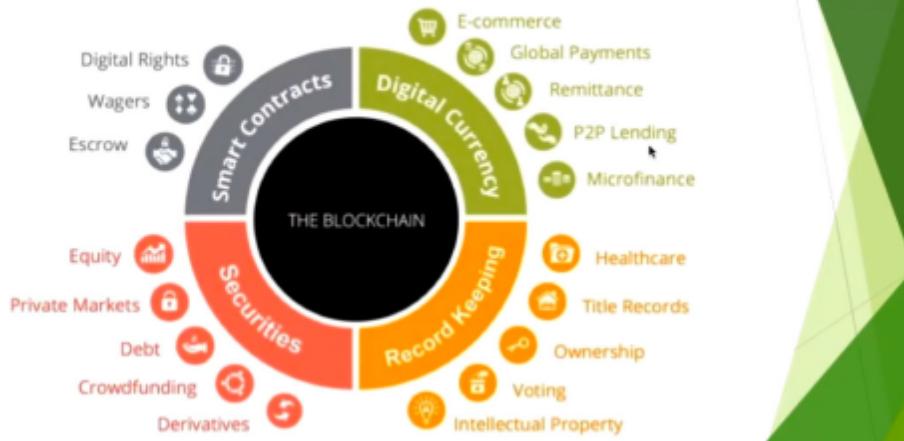


Figura 234: Categorie che utilizzano bitcoin

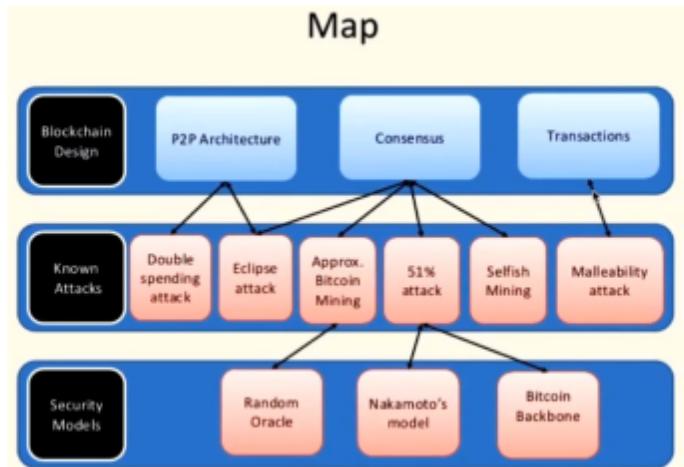


Figura 235: Mappa: possibili attacchi->superficie da attaccare

10.2.10 Double Spending

A acquista un bene tramite internet con pagamento bitcoin, A riceve il bene da B e poi invia/divulga tanti blocchi minati (dopo che i blocchi sono stati minati vengono messi da parte aspettando di essere inviati tutti in una volta) in modo tale che la catena diventi tanto lunga da sovrascrivere le operazioni precedenti e annullare la transazione. Ad oggi è molto difficile realizzare questo tipo di attacchi vista la dimensione della rete e le elevate potenze di calcolo richieste. Questo è attacco è stato fatto da "Finney".

La soluzione più comune prevede che B consegni la merce acquistata da A soltanto dopo che la transazione è stata scritta in blockchain dai miner, quindi non basta che la

transazione venga inviata da A al pool (il passaggio successivo al pool è appunto l'aggiunta in blockchain da parte dei miner). Per evitare anche il caso estremo del partizionamento della rete si potrebbe anche aspettare che la transazione raggiunga prima il sesto-settimo blocco, quindi 60-70 minuti in modo tale che non possa essere sovrascritta dal percorso più lungo.

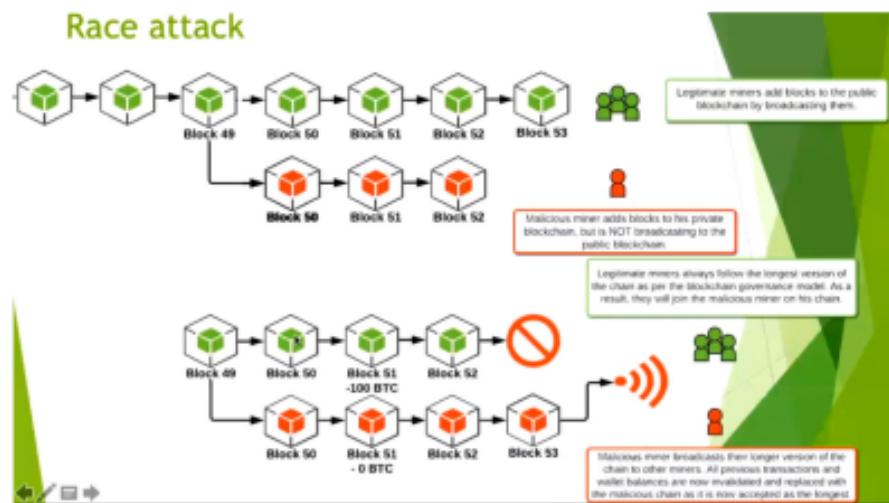


Figura 236: Flusso double spend attack

A porta avanti due flussi di transazioni in parallelo minando le transazioni a due livelli (verde e rosso), appena A riceve la merce da B (flusso verde) effettua il broadcast delle transazioni minate di nascosto sul flusso rosso. Appena il broadcast viene diffuso il flusso verde viene sovrascritto in quanto più corto e non divulgato, quindi i soldi non vengono realmente spesi. E' rarissimo se non impossibile che nel bitcoin (a differenza di monete emergenti dove le blockchain sono ancora di piccole dimensioni) questo attacco vada a buon fine, nessuno disporrebbe della potenza di calcolo richiesta per attaccare una blockchain di tali dimensioni.

>50% or Goldfinger attack

51% Attack (double-spend)

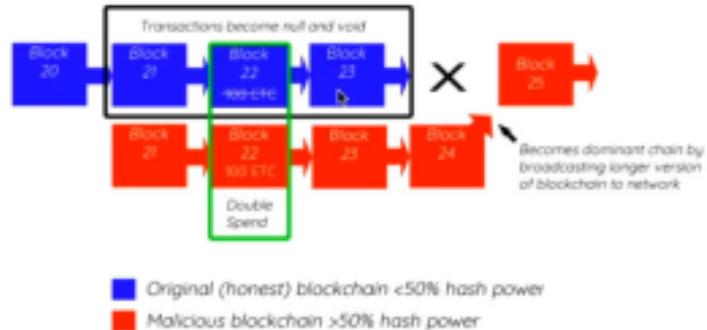


Figura 237: 51% Attack double-spend

Se ho il 51% di potenza di calcolo, oltre al double spend posso fare molti altri tipi di attacco, posso praticamente riscrivere le regole della blockchain. Es. posso cambiare dimensioni e regole delle transazioni. Es. posso effettuare short selling, ovvero pilotare il prezzo del bitcoin a mio piacimento. Es. posso distruggere completamente la moneta.

Il governo cinese potrebbe decidere di impossessarsi dell'80% della rete visto che la maggior parte delle farm è situata in Cina, così facendo avrebbe il controllo assoluto sul bitcoin.

Per evitare il 51% attack i detentori dei pool hanno stretto un patto fissando dei limiti alla distribuzione delle potenze di calcolo. || Nel 2018 la potenza computazionale della rete bitcoin era 7648 volte più grande del computer più potente del mondo.

MINING POOLS HISTORY

- first pool appears in late-2010, back in the GPU era !
 - by 2014: around 90% of mining pool-based
 - miners tend to flock to the largest pools.
- june 2014: one pool, Ghash.io, in 2014 reached 55%
 - possibility of 50% attack: total control of bitcoin by one single entity.
 - they have publicly said: please leave, do not join !

pool	percentage
BTCGuild.com	26%
GHash.io	40%
Eligius.st	10%
Bitcoin.cz	6%
Bitminter.com	5%

Figura 238: Mining pools history

HASH POWER DISTRIBUTION: CURRENT SCENARIO

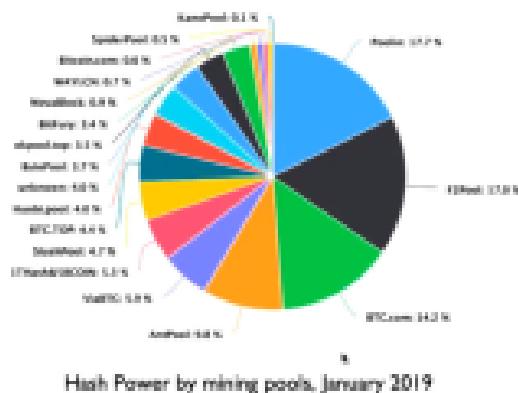


Figura 239: Distribuzione pool-potenze di calcolo

Esistono siti che vendono anche le potenze di calcolo. Per un utente quasi impossibile acquistarne una quantità per la moneta bitcoin.

10.2.11 Wallet Attack

Sono attacchi che colpiscono il software che gestisce le chiavi pubbliche e private, il wallet.

Avere il wallet non è una cosa obbligatoria per gestire le blockchain, però vista la mole delle chiavi da gestire è praticamente indispensabile. L'attacco può avvenire sulla macchina dell'utente che utilizza il wallet oppure sull'exchange server dove l'utente è registrato.

Come si fa a gestire bene un wallet? Quando avviene uno scambio di denaro in criptovaluta e si deve fare una transazione bisogna mandarla a un nostro indirizzo pubblico gestito da noi sui nostro wallet. Il wallet potrebbe essere un software sul cellulare o su un computer, ma la cosa migliore è avere dei wallet esterni/token crittografici esterni da sbloccare con pin (è come quando accediamo ad una smart card), così se uno ruba il dispositivo e riesce a sbloccarlo non può comunque accedere ai dati contenuti dentro. Con la tecnica della split key la chiave viene memorizzata su device diversi, quindi l'attacco andrebbe portato a termine su più device e non soltanto su uno.

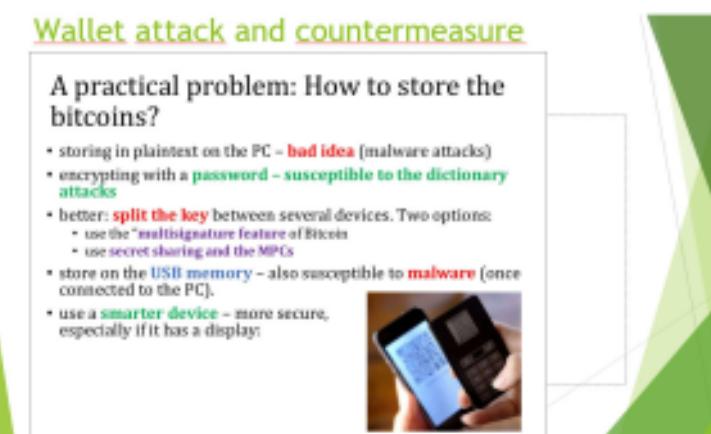


Figura 240: Wallet attack and countermeasure

10.2.12 Transaction Malleabilit

Sfrutta il fatto che una parte della transazione è firmata con la chiave privata del possessore dei fondi. Ogni transazione è unicamente identificata da un hash, questo hash non è calcolato solo sui dati finanziari ma anche sulla firma (che può essere scritta in modi diversi). Soluzione possibile: witness, transazioni SegWit (Segregated Witness).

Network attack

IDENTITY ATTACKS

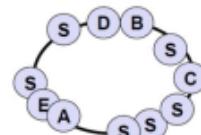
- how are identities assigned in a P2P network?
 - every entity (peer, content) has a unique identity
 - generated by a cryptographic hash
- how to exploit identities to attack the network? a malicious user may
 - obtain multiple identities
 - **Sybil attack**
 - obtain a specific ID, next to the value of the key of a resource or a to node it wants to control
 - **ID Mapping Attack**

Figura 241: Identity attacks

Sybil attack:

IDENTITY ATTACKS IN P2P NETWORKS

- inject **one or multiple fake identities (sybils)** into the network
 - use them as a starting point to perform further attacks.
- potential goals
 - routing attacks
 - control data replica
 - network connectivity
 - in case of majority votes, be the majority.
 - cryptocurrencies consensus
- efficient attack against
 - peer-to-peer
 - other decentralized networks.



node S (sybil) is in the network
with 6 different identities.

Figura 242: Sybil attack

ATTACKING THE ROUTING

- attacks to the routing process
 - routing table poisoning
 - propagate wrong information for routing
 - eclipse attack
 - Strictly related to sybil and a routing table posoning attacks
 - misroute messages
 - change target while forwarding
 - either randomized or according to some plan
 - pretend to be the key manager

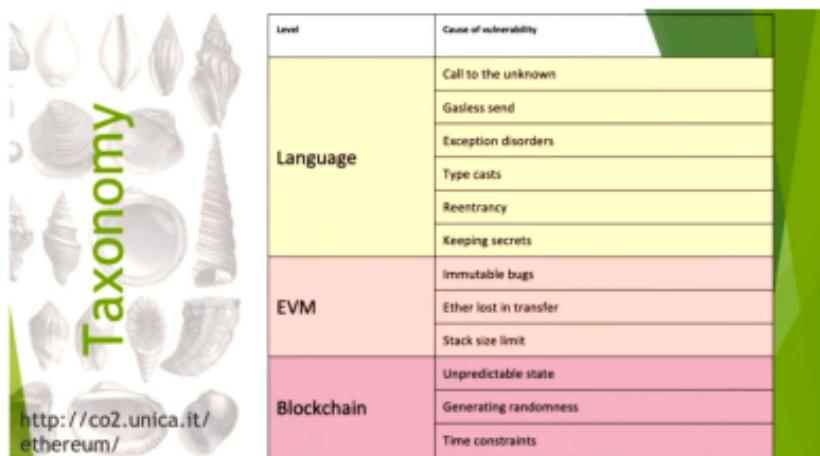
Figura 243: Sybil attack, routing

Routing: creando ed impersonando diversi nodi della rete l'attaccante può collegarli ad un determinato nodo in modo tale che venga isolato dagli altri.

Eclipse: nasconde il resto della rete al nodo stesso, così tutti i nodi che sono collegati a lui (quelli che ha creato l'attaccante) possono gestire a loro piacimento il routing (routing table poisoning).

10.2.13 Code vulnerability

Vulnerabilità del codice e/o del software.



Level	Cause of vulnerability
Language	Call to the unknown
	Gasless send
	Exception disorders
	Type casts
	Reentrancy
	Keeping secrets
EVM	Immutable bugs
	Ether lost in transfer
	Stack size limit
Blockchain	Unpredictable state
	Generating randomness
	Time constraints

<http://co2.unica.it/ethereum/>

Figura 244: Esempi di attacchi subiti da Ethereum

Gli smart contracts sono i principali destinatari degli attacchi software. I contratti una volta messi in blockchain sono immutabili, trasparenti (tutti possono guardare il codice), vengono eseguiti in maniera autonoma (quando per esempio scattano degli eventi) e si possono controllare sempre visto che rimangono in blockchain. Con gli smart contract si può eseguire della computazione o trasferire del denaro, in ethereum hanno anche un indirizzo pubblico dedicato come per gli utenti.

Smart contracts characteristics



Figura 245: Caratteristiche smart contract

Caso DAO attack (Decentralized Autonomous Organization)

Era uno smart contract che permetteva di effettuare transazioni senza utilizzo di moneta, uno dei metodi era bacato e poteva essere chiamato più volte senza fare un controllo. Un utente sfruttò questo metodo per indirizzare tanti smart contract a se stesso, i contratti però non potevano essere cambiati in quanto immutabili. L'unica soluzione era fare una sorta di roll back di tutti i blocchi inseriti prima e ripartire, non tutti però erano

ovviamente d'accordo. Alla fine la rete si è divisa in due mondi: Ethereum ed Ethereum Classic, i primi hanno effettuato il roll back ed hanno corretto il baco, i secondi no.

DAO – Decentralized Autonomous Organization



Figura 246: DAO - Decentralized Autonomous Organization