

# Appunti di Introduction to Computer and Network Security

Luca Facchini

Matricola: 245965

Corso tenuto dal prof. Raniese Silvio

Università degli Studi di Trento

A.A. 2024/2025

## **Sommario**

Appunti del corso di Introduction to Computer and Network Security tenuto dal prof. Raniese Silvio presso l'Università degli Studi di Trento nell'anno accademico 2024/2025.

# Indice

<b>1</b>	<b>Basic Notions - Nozioni Base</b>	<b>3</b>
1.1	The CIA Triad . . . . .	3
1.2	Security Policy, Mechanism, Service . . . . .	4
1.3	CIA, Security Violation, and Mitigation . . . . .	5
1.4	Risk . . . . .	6
1.5	Fine della diffusione degli attacchi . . . . .	7
1.6	Security & Human Factor - Sicurezza & Fattore Umano . . . . .	7
<b>2</b>	<b>Authentication I: Passwords &amp; co</b>	<b>9</b>
2.1	User Authentication & Digital Identity - Autenticazione e Identità Digitale . . . . .	9
2.2	An Introduction to Passwords - Introduzione alle Password . . . . .	10
2.3	Multi Factor Authentication - Autenticazione a più fattori . . . . .	11
2.4	Outsourcing Authentication . . . . .	13
<b>3</b>	<b>Cryptography Introduction</b>	<b>15</b>
3.1	Cryptosystem . . . . .	15
3.2	Types of cryptography - Tipologie di crittografia . . . . .	16

# Capitolo 1

## Basic Notions - Nozioni Base

### 1.1 The CIA Triad

#### 1.1.1 Introduzione

**L'acronimo** L'acronimo CIA in inglese sta per **Confidentiality**, **Integrity** e **Availability**. Questi tre concetti sono alla base della sicurezza informatica e della protezione dei dati.

**Confidentiality - Riservatezza** Il principio della Riservatezza prescrive che i non autorizzati non devono poter accedere ai dati e i dati sono visibili solo a chi è autorizzato a vederli.

**Integrity - Integrità** Il principio di Integrità prescrive che i dati non devono essere modificati se non da persone autorizzate a farlo, i tentativi di modifica non autorizzati devono essere rilevati e prevenuti.

**Availability - Disponibilità** Il principio della Disponibilità prescrive che i dati non devono essere mantenuti inaccessibili ed inoltre bisogna permettere l'accesso agli autorizzati alle informazioni e ai servizi.

#### 1.1.2 In Pratica

**Confidentiality - Riservatezza** L'accesso alle informazioni riservate può essere:

**Intenzionale** Quando un intruso cerca di accedere a dati sensibili.

**Non Intenzionale** Quando un utente autorizzato accede a dati sensibili senza volerlo causato dal fatto che chi detiene i dati non è competente e/o non gli importa della sicurezza.

Come garantire la riservatezza:

**Crittografia** La **crittografia** è una tecnica che permette di limitare l'accesso ai dati solo a chi possiede la chiave per decifrarli (e quindi è autorizzato).

**Access Control** Il **controllo degli accessi** è una parte integrante per mantenere la **riservatezza**, ciò andando a gestire quali utenti possono accedere a quali risorse.

**Integrity - Integrità** Per garantire l'integrità bisogna prevenire la modifica e/o la distruzione, inoltre bisogna assicurare informazioni autentiche e non invalidate.

Inoltre ulteriore prescrizione è quella di individuare anche le cosiddette "minute changes" ovvero le modifiche minime che possono essere fatte ai dati per alterarne il significato.

L'integrità può essere anche di tipo **data integrity** ovvero la garanzia che i dati siano corretti e completi, oppure di tipo **system integrity** ovvero la garanzia che il sistema sia protetto da attacchi e che funzioni correttamente.

L'integrità può essere compromessa da:

- errore umano
- attacchi come malware distruttivi e ransomware

Come garantire l'integrità:

**Version Control** Il **controllo delle versioni** assieme agli **audit trails** permettono ad una organizzazione di garantire la veridicità dei dati.

**Compliance requirements** I **requisiti di conformità** sono regole e normative che un'organizzazione deve seguire per garantire l'integrità dei dati.

**Availability - Disponibilità** Le violazioni alla disponibilità includono:

- Fallimento dell'infrastruttura
- Overload dell'infrastruttura
- Blackout
- Attacchi come DDoS

Come garantire la disponibilità:

**Backup** Bisogna sempre avere un piano di **backup** per mantenere disponibile in caso di disastro, attacco o altre minacce si verificassero.

**Cloud** Il **cloud** è una soluzione per garantire la disponibilità dei dati, in quanto i dati sono replicati su più server e quindi in caso di guasto di uno di essi i dati sono comunque disponibili.

## 1.2 Security Policy, Mechanism, Service

### 1.2.1 Security Policy

#### Definizione

Una **security policy** è un insieme di regole e requisiti stabilita da una azienda che definisce l'uso accettabile delle informazioni e dei servizi, inoltre il livello indica il livello di confidenzialità, integrità e disponibilità delle informazioni.

#### Esempi

Un esempio di possibili regole di una **security policy** potrebbero essere:

- I dipendenti devono completare il corso su sicurezza informatica e accettare la policy di sicurezza.
- I visitatori dell'azienda devono essere accompagnati da un dipendente autorizzato in tutti i momenti, questo dipendente deve mantenere i visitatori in una area appropriata senza dati sensibili
- I dipendenti devono mantenere una "scrivania pulita" dunque senza lasciare documenti sensibili accessibili e incustoditi.
- I dipendenti devono usare una password sicura e queste non devono essere usate in altri servizi esterni.
- Gli ex-dipendenti non devono mantenere alcun dato dell'azienda.

### 1.2.2 Security Service

#### Definizione

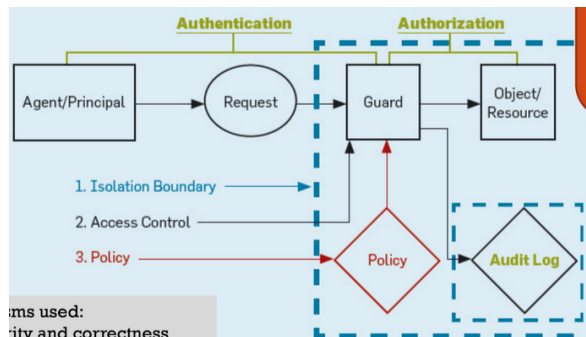
La capacità di supportare uno o più dei requisiti di sicurezza (Confidentiality, Integrity, Availability) e.s.: gestione chiavi, controllo accessi, autenticazione. Come già detto i **Security Services** implementano uno o più **Security Mechanism** che servono per far rispettare la **Security Policy**.

#### Esempi

##### HTTP(S)

**HTTP** Se si usa il protocollo **HTTP** per la trasmissione dei dati qualsiasi informazione trasmessa può venire intercettata da hacker nel mezzo.

**HTTPS** Per ovviare a questo problema si usa il protocollo **HTTPS** che è una versione sicura di **HTTP** che usa la crittografia per proteggere i dati, in questo modo i dati trasmessi vengono cifrati e se intercettati non possono essere letti.



### Access Control

**Audit Log** è un registro importante che registra tutte le richieste di accesso ai dati e se la richiesta è stata accettata o rifiutata, in questo modo si può monitorare se una richiesta che è stata fatta ha avuto un risultato non aspettato e quindi che vadano modificate le **policy**.

## 1.2.3 Security Mechanism

### Definizione

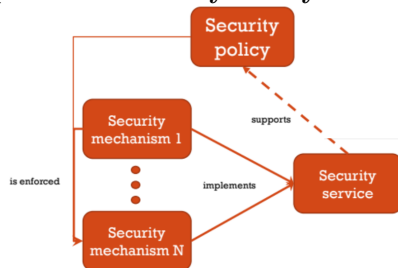
Un **security mechanism** un dispositivo o funzione designato per provvedere uno o più servizi di sicurezza classificate in termini di potenza

Inoltre è l'implementazione della **security policy**

### Esempi

## 1.2.4 Come si relazionano

In conclusione i **Security Services** implementano uno o più **Security Mechanism** che servono per far rispettare la **Security Policy**.



## 1.3 CIA, Security Violation, and Mitigation

La triade CIA non è solo essenziale per rendere sicuri i dati ma aiuta anche a capire cosa è andato storto nel caso di una **security violation**

### 1.3.1 Attacco Ransomware

**L'attacco** Nel caso di un attacco ransomware i dati vengono tenuti in ostaggio e la vittima viene ricattata con la pubblicazione di dati personali o il mancato accesso ad essi fino a che un riscatto non viene pagato.

Questo attacco cripta i dati rendendoli inaccessibili e per ripristinarli è necessaria una chiave di decodifica, i pagamenti solitamente vengono eseguiti in crypto valute poco tracciabili.

Questo genere di attacco viola la **confidentiality** e la **availability** dei dati.

**Mitigation** Per mitigare il rischio (che comunque non può essere eliminato) è utile procedere con un approccio **Zero Thrust** ovvero un "protocollo" che richiede che tutti gli utenti all'interno e all'esterno della rete aziendale siano autenticati e che la connessione è continuamente validata, prima di consentire accesso a applicazioni e dati.

## 1.4 Risk

Non esiste un sistema sicuro, esistono solo diversi livelli di insicurezza, si punta sempre ad ottenere il miglior livello di sicurezza possibile con il budget a disposizione.

### 1.4.1 Vulnerability

#### Definizione

Una **vulnerabilità** è una debolezza in un sistema informativo, un sistema di sicurezza, nei controlli interni o nell'implementazione di questi. Questa debolezza può essere sfruttata o compromessa da una minaccia alla sicurezza.

#### Examples

Un esempio di **vulnerabilità** è una backdoor nascosta nel software e/o hardware, oppure dei bug non conosciuti di un software, o anche delle password deboli.

### 1.4.2 Threat

#### Definizione

Qualunque circostanza o evento con il potenziale per causare perdite o danni a: operazioni dell'azienda, risorse o individui che possono accedere a tali risorse tramite un accesso non autorizzato, distruzione, pubblicazione, modifica e/o DoS. Inoltre un **threat** è un potenziale attacco che sfrutta una **vulnerabilità** del sistema.

#### Examples

Un esempio di **threat** sono:

- Hacker:
  - Trova e/o **sniff** una password.
  - Usa il **social engineering** per ottenere informazioni sensibili e password.
  - Occupa le risorse di sistema con richieste inutili (attacchi DoS).
- Virus e worms:
  - I **virus** sono programmi auto replicanti che richiedono una azione dell'utente per essere attivati, esempio ne è una Email, un allegato o un link o un Floppy/CD/USB infetto.
  - I **worms** sono programmi auto replicanti che non richiedono una azione dell'utente per essere attivati, esempio ne è un worm che sfrutta una vulnerabilità di un sistema operativo.

### 1.4.3 Attack

#### Definizione

Un **attacco** è un qualsiasi attività dannosa che prova a raccogliere, degradare, negare o distruggere informazioni o servizi. Può anche manifestarsi sotto un tentativo di ottenere un accesso ad una risorsa alla quale non si ha diritto

### 1.4.4 Risk

#### Definizione

Il **rischio** è la **probabilità** che una particolare **minaccia** sfrutti una particolare **vulnerabilità**. Il **rischio** è inoltre una misura di quanto una particolare circostanza o evento possa essere potenzialmente minacciato ed è tipicamente in funzione di:

1. L'impatto che questa circostanza o evento potrebbe avere
2. La probabilità che questa circostanza o evento si verifichi

#### Riassumendo

Riassumendo le **minacce** sono una combinazione di **intento** e **capability** di un attacco con successo. Invece le **vulnerabilità** sono caratterizzate da quanto è facile **identificarle** e **sfruttarle**.

C'è da dire che l'impatto di un attacco deve essere valutato dal punto di vista di ogni soggetto coinvolto, ad esempio un attacco che compromette la privacy di un singolo individuo può avere un impatto molto diverso se lo si guarda dal punto di vista dell'individuo o dell'azienda.

#### Risk Matrix

		Severity of Impact				
		1	2	3	4	5
Likelihood	5 Highly Probable	1x5=5	2x5=10	3x5=15	4x5=20	5x5=25
	4 Probable	1x4=4	2x4=8	3x4=12	4x4=16	5x4=20
	3 Occasional	1x3=3	2x3=6	3x3=9	4x3=12	5x3=15
	2 Remote	1x2=2	2x2=4	3x2=6	4x2=8	5x2=10
	1 Improbable	1x1=1	2x1=2	3x1=3	4x1=4	5x1=5

La presente matrice è una rappresentazione grafica del rischio, in cui si valuta la probabilità di un attacco e l'impatto che questo avrebbe, solitamente un rischio per essere accettabile deve avere un valore risultante tra basso (1-4) e medio-basso (5-9), i valori quali medio-alto (10-14) e alto (15-25) sono considerati inaccettabili e quindi devono essere mitigati abbassando la probabilità di attacco e/o l'impatto che questo avrebbe.

## 1.5 Fine della diffusione degli attacchi

**Generalmente** Il rischio è causato dunque dalla somma di una minaccia e una vulnerabilità, la minaccia è la capacità di sfruttare una vulnerabilità per causare un danno.

**Pianificazione** Quando si studiano i rischi sia prima che dopo un attacco bisogna considerare che se la **Minaccia - Threat** riesce a sfruttare la **Vulnerabilità - Vulnerability** allora si ha un **Rischio - Risk** se questo risulta inaccettabile allora si deve procedere con la **Mitigazione - Mitigation** del rischio, ciò riduce la **Minaccia - Threat** e il ciclo riprende fino a quando il **Rischio residuo - Residual Risk** è accettabile.

## 1.6 Security & Human Factor - Sicurezza & Fattore Umano

Il fattore umano è uno dei fattori più importanti nella sicurezza informatica, infatti la maggior parte degli attacchi sono causati da password poco sicure o da una mancata preparazione degli utenti.

### 1.6.1 Passwords

C'è chi dice che le password andrebbero trattate come la biancheria intima, ovvero:

- Cambiate regolarmente - magari senza usare un pattern

- Non condivise con nessuno - nemmeno con colleghi o familiari
- Non lasciate sulla scrivania - non scritte su post-it o in chiaro

### 1.6.2 Security

Ogni persona che ha accesso ad un sistema informatico è un potenziale punto di attacco, quindi è importante che ogni utente sia formato sulla sicurezza, ogni utente al suo livello a partire dall'utente base che necessita di una formazione base fino ad arrivare all'amministratore di sistema che necessita di una formazione più avanzata.



## Capitolo 2

# Authentication I: Passwords & co

## 2.1 User Authentication & Digital Identity - Autenticazione e Identità Digitale

### 2.1.1 Introduzione e Definizioni

**Identità** è un insieme di attributi relazionati ad una entità

**Attributo** è una caratteristica o proprietà di una entità che può essere usata per descriverne lo stato, apparenza o ogni altro aspetto.

**Identità digitale** è una identità i cui attributi sono conservati e trasmessi in forma digitale

Ma perché la identità digitale è importante? In pratica questa può essere una soluzione per diversi aspetti che i governi mondiali vorrebbero risolvere.

### 2.1.2 Digital Identity Lifecycle - Ciclo di Vita dell'Identità Digitale

**Creazione** L'identità viene creata, in genere da un ente di certificazione che assicura l'identità dell'utente e fornisce/chiede una forma di accesso per riconoscere in futuro l'utente. Viene creato un ID univoco per l'utente

**Autenticazione** L'identità viene autenticata tramite il metodo concordato in fase di creazione

**Autorizzazione/ Accesso** L'utente ha accesso a risorse e servizi in base ai permessi assegnati, i dati necessari dell'utente vengono trasmessi dall'ente di certificazione al servizio.

**Cancellazione o Disattivazione** L'identità viene cancellata o disattivata quando non è più necessaria e i dati devono essere cancellati dopo un certo periodo di tempo per garantire la privacy dell'utente.

#### Enrollment / On-Boarding - Creazione

La fase di creazione di una utenza porta un **Applicant** a diventare un **Subscriber** tramite una serie di passaggi:

**Resolution** Vengono raccolti gli **attributi** essenziali dell'utente (nome, indirizzo, data e luogo di nascita, ...) vengono ora raccolti anche due documenti di identità (patente, passaporto, C.I.,...) - ora l'utente viene distinto univocamente su un contesto

**Validation** Gli **attributi** raccolti vengono validati sulla base di **prove** tramite una fonte autorevole- gli attributi vengono ora associati ad una persona fisica

**Verification** Le **prove** vengono verificate, in questo punto si verifica la corrispondenza tra le diverse foto, viene mandato un codice ai contatti per verificare che siano i suoi, etc - l'utente viene ora confermato e la sua identità è certificata

**Identity Assurance Levels - Livelli di Assicurazione dell'Identità**

**IAL1** GLi attributi, se presenti, vengono auto-dichiarati, o considerati come tali

**IAL2** Una persona o di remoto o in presenza, verifica gli attributi

**IAL3** Gli attributi vengono verificati da una persona autorizzata e i documenti fisici vengono esaminati

**Authentication - Autenticazione**

**Definizione** L'**autenticazione** è il processo di verifica dell'identità di un utente, processo o dispositivo. Il **richiedente** deve dimostrare al **verificante** che è chi dice di essere.

**Base** Quando esegui il login solitamente inserisci un **username** e una **password** che vengono confrontati con quelli memorizzati nel sistema.

L'autenticazione tramite **password** è molto diffusa e semplice da implementare.

## 2.2 An Introduction to Passwords - Introduzione alle Password

### 2.2.1 User Authn - Autenticazione Utente

Quando esegui il login inserisci un **username** per annunciare chi sei e una **password** per dimostrare chi dici di essere, questo tipo di autenticazione si chiama **user authn** ovvero il processo di verifica dell'identità di un utente.

**Entollment - Creazione**

Le **password** dovrebbero essere conosciute solo dall'utente e dal sistema. Spesso però, soprattutto nelle grandi aziende, le password vengono mandata via mail o scritte dall'utente su una pagina web. In questi casi da chi potrebbe essere intercettata la password?

**La teoria rispetto alla realtà**

Le password hanno dominato il mondo dell'informatica sin dall'avvento dei computer, ma da allora la richiesta verso queste è che siano più sicure ma allo stesso tempo più "User Friendly", siamo in ricerca di alternative ma per ora delle proposte alternative come *criteri di complessità* ma studi hanno rilevato che spesso non vengono rispettati e/o che non sono efficaci.

### 2.2.2 Password Security - Sicurezza delle Password

All'inizio di internet le password venivano conservate in file appositi in chiaro, ma questo comportava un rischio molto alto, infatti se un attaccante riusciva ad accedere a quel file poteva facilmente accedere a tutte le password.

**Hashing**

Per risolvere questo problema si è pensato di "**hash-are**" le password, ovvero di applicare una funzione di hash alla password e conservare il risultato. In questo modo se un attaccante accede al file delle password non può risalire alla password originale.

**Proprietà della funzione di hash** Una funzione di hash ottimale dovrebbe avere le seguenti proprietà:

**Deterministica** dato un input la funzione restituisce sempre lo stesso output

**Rapida** la funzione deve essere veloce da calcolare

**Difficile da invertire** data l'output è difficile risalire all'input

**Biettiva** due input diversi devono avere output diversi

**Lunghezza fissa** l'output deve avere una lunghezza fissa indipendentemente dalla lunghezza dell'input

Molto spesso però capita che i bit disponibili per l'hash siano limitati rispetto a tutti i possibili input, per questo motivo si è cercato di creare funzioni di hash che generano due output uguali solo in rarissimi casi, inoltre si aggiunge a questo che per input simili l'output sia molto diverso.

### Crack of Passwords

In caso di un "leak" di password che sono state opportunamente codificate con una funzione di hash, un attaccante può tentare di decifrarle ma questo è molto difficile. Questo diventa più semplice se vengono usate password poco sicure e/o comuni. infatti si stima che per trovare una password da 8 caratteri composti da lettere minuscole e numeri ci vogliano "solo" 155€ su una istanza di AWS.

### 2.2.3 Hash & Salt

Per rendere più difficile il lavoro degli attaccanti si è pensato di aggiungere un **salt** alle password, ovvero un valore casuale che viene aggiunto alla password prima di calcolare l'hash. Questo rende più difficile per l'attaccante decifrare la password.

## 2.3 Multi Factor Authentication - Autenticazione a più fattori

### 2.3.1 Definizione

L'autenticazione a più fattori è un metodo di autenticazione che richiede l'uso di più metodi di autenticazione per verificare l'identità di un utente.

### 2.3.2 Fattori di Autenticazione

I fattori di autenticazione sono:

**Qualcosa che sai** (password, PIN, ...)

**Qualcosa che hai** (smartphone, token, ...)

**Qualcosa che sei** (impronte digitali, riconoscimento facciale, ...)

**Vantaggi/Svantaggi quello che sai**

**Vantaggi**

- Facile da implementare - non richiede hardware aggiuntivo
- Facile da usare - Basta ricordare la password
- Facile da resettare se dimenticato - basta fare il reset della password

**Svantaggi**

- Facile da rubare - se un attaccante riesce a scoprire la password può accedere al sistema
- Facile da dimenticare - se la password è complessa è facile dimenticarla
- Facile da indovinare - se la password è semplice ed è stata usata in altri contesti è facile indovinarla

### Vantaggi/Svantaggi quello che hai

#### Vantaggi

- Difficile da rubare - se un attaccante non ha il dispositivo non può accedere al sistema
- Difficile da indovinare - se il dispositivo è protetto da PIN o password è difficile indovinare l'accesso
- Difficile da clonare - se il dispositivo è protetto da un token è difficile clonarlo inoltre la parte di autenticazione è fatta dal dispositivo stesso e non dal sistema

#### Svantaggi

- Facile da perdere - se il dispositivo viene perso l'utente non può accedere al sistema
- Difficile da resettare - se il dispositivo viene perso l'utente deve contattare l'amministratore per resettare l'accesso
- Costoso - i dispositivi possono essere costosi

### Vantaggi/Svantaggi quello che sei

#### Vantaggi

- Difficile da rubare - le impronte digitali o il riconoscimento facciale sono unici e difficili da rubare
- Difficile da indovinare - è difficile indovinare le impronte digitali o il riconoscimento facciale
- Difficile da clonare - è difficile clonare le impronte digitali o il riconoscimento facciale

#### Svantaggi

- Se il fattore viene compromesso non può essere cambiato - se le impronte digitali vengono rubate non possono essere cambiate
- Costoso - i dispositivi possono essere costosi
- Non sempre preciso - il riconoscimento facciale può essere ingannato

### PSD2 Compliance

**Cos'è PSD2** La **PSD2** è una direttiva europea che regola i servizi di pagamento e che richiede l'autenticazione a più fattori per i pagamenti online. Questo per ridurre il rischio di frodi.

**How to comply MFA in PSD2** Una idea sarebbe quella di includere nella challenge anche i dati della particolare transazione come:

- L'identificativo del destinatario della transazione
- L'importo della transazione
- L'istante nella quale la transazione è stata inizializzata
- ...

Purtroppo però ciò non è abbastanza in quanto i dati sopra indicati possono essere ricavati dal contesto il che rende la challenge prevedibile.

### 2.3.3 FIDO: Phishing Resistant Authentication

#### Cos'è FIDO

**FIDO (Fast Identity Online)** è un insieme di standard aperti per l'autenticazione a più fattori che mira a ridurre il rischio di phishing. Il suo scopo è quello di assicurare una **autenticazione forte** e di **ridurre l'uso di password**.

#### Come funziona

1. Viene chiesto all'utente di scegliere un ente FIDO
2. L'utente sblocca il dispositivo FIDO usando un'impronta digitale, un pulsante su un dispositivo di secondo fattore, un PIN o un qualsiasi altro metodo di autenticazione supportato
3. Il dispositivo crea una chiave pubblica e una chiave privata univoca per il dispositivo, il servizio online e l'utente
4. La chiave pubblica è inviata al servizio online ed associata all'account dell'utente
5. Il servizio online chiede all'utente di autenticarsi con il dispositivo precedentemente registrato
6. L'utente sblocca il dispositivo FIDO usando lo stesso metodo di autenticazione usato in precedenza
7. Il dispositivo FIDO usa l'account dell'utente identificato per inviare la corretta chiave al servizio online
8. Infine il dispositivo invia la challenge ricevuta dal servizio online firmata con la chiave privata e il servizio online verifica la firma con la chiave pubblica, l'utente è autenticato.

## 2.4 Outsourcing Authentication

### 2.4.1 Definizione

L'outsourcing dell'autenticazione è il processo di affidare l'autenticazione degli utenti ad un servizio esterno, spesso chiamato **Identity Provider**. Questo servizio si occupa di verificare l'identità dell'utente e di inviare un token di autenticazione al servizio che richiede l'autenticazione. Esempio in Italia per l'accesso ai servizi pubblici si usa SPID.

### 2.4.2 Che problema risolve

L'outsourcing dell'autenticazione risolve diversi problemi:

**Sicurezza** L'Identity Provider è specializzato in autenticazione e può offrire un livello di sicurezza più alto

**User Experience** L'Identity Provider può offrire un'esperienza utente migliore in quanto l'utente non deve ricordare le password per ogni servizio ma solo quella dell'Identity Provider - SSO

**Compliance** L'Identity Provider può aiutare a rispettare le normative sulla privacy e la sicurezza

### 2.4.3 La soluzione

L'outsourcing dell'autenticazione è delegato ad una terza parte, l'**Identity Provider**, che si occupa di verificare l'identità dell'utente e di inviare le informazioni di autenticazione al servizio che richiede l'autenticazione. A questo punto il servizio invia all'utente un token di autenticazione che può essere usato per accedere al servizio senza dover inserire nuovamente le credenziali del SSO.

#### 2.4.4 Potenziali Problemi

L'outsourcing dell'autenticazione può comportare alcuni problemi:

**Single Point of Failure** Se l'Identity Provider è compromesso tutti i servizi che usano l'Identity Provider per l'autenticazione sono compromessi

**Privacy** L'Identity Provider può raccogliere informazioni sull'utente e sulle sue abitudini di navigazione

## Capitolo 3

# Cryptography Introduction

### 3.1 Cryptosystem

**Definizione** Un **cryptosystem** è una tupla di 5 elementi (E,D,M,K,C):

**E** è un *algoritmo di cifratura*

**D** è un *algoritmo di de-cifratura*

**M** è un insieme di *messaggi in chiaro*

**K** è un insieme di *chiavi*

**C** è un insieme di *messaggi cifrati*

In modo astratto i punti **E** e **D** possono essere espresse come funzioni:

$$\begin{aligned} E : M \times K &\rightarrow C \\ D : C \times K &\rightarrow M \end{aligned} \quad D(E(m, k), k) = m$$

Nella crittografia base ogni chiave  $k \in K$  può essere usata per cifrare e de-cifrare i messaggi, nella crittografia simmetrica la chiave è la stessa per entrambi i processi, mentre nella crittografia asimmetrica le chiavi sono diverse.

**Quali componenti sono pubbliche?** Solitamente non abbiamo necessità di mantenere segrete le funzioni E e D o il messaggio cifrato C, quello che davvero deve essere segreto è la chiave K. Il motivo per il quale le funzioni non devono essere segrete è che il sistema deve essere pubblico e quindi tramite processi di *reverse engineering* è possibile ricavare le funzioni (es: il sistema di cifratura di un DVD è stato ricavato tramite *reverse engineering* dopo solo due giorni).

#### 3.1.1 Why Cryptography?

**Sicurezza** La crittografia è usata nei **meccanismi di sicurezza** per garantire **confidenzialità** nascondendo il contenuto dei messaggi, **integrità** provvedendo al controllo di integrità tramite funzioni di **hash** e **la verifica dell'origine** dei dati tramite firme digitali verificabili da una fonte autorevole.

#### 3.1.2 Cryptography on rented servers

**Problema** Se si usa un server di terze parti per conservare dati, ad esempio su un database, è possibile che il proprietario del server possa accedere ai dati in chiaro. Per evitare ciò si può cifrare i dati prima di inviarli al server, in modo che il proprietario non possa leggerli, ciò comporta però un aumento del carico computazionale lato client in quanto tutti i dati prima di essere letti devono essere de-cifrati, per questo esistono meccanismi di ricerca su dati cifrati che permettono di effettuare ricerche su dati cifrati senza de-cifrarli, una volta trovati i dati desiderati si de-cifrano solo quelli.

### 3.1.3 Come definiamo "Computazionalmente sicuro" nelle comunicazioni

**Definizione** Definiamo un sistema di comunicazione **Computazionalmente sicuro** quando la decifrazione di un messaggio cifrato senza conoscere la chiave è molto difficile, o addirittura impossibile, e richiede molto tempo e risorse computazionali.

$E(k, P) = C$  Calcolare  $C$  da  $P$  deve essere difficile senza  $k$ , inoltre calcolare  $C$  da  $P$  sapendo  $k$  deve essere facile.

$D(k, C) = P$  Calcolare  $P$  da  $C$  deve essere facile sapendo  $k$ , ma deve essere difficile senza  $k$ .

**Trapdoor - funzione a senso unico** Una **trapdoor** è una funzione a senso unico che richiede una ulteriore informazione. Ed una **funzione a senso unico** è descritta come una funzione che è facile da calcolare in una direzione, ma molto più difficile nell'altra.

### 3.1.4 Hash v/s Encryption

**Quando l'una e quando l'altra** Usiamo funzioni di hash quando non abbiamo bisogno di accedere all'informazione originale, ma solo di verificare l'integrità dei dati. Ricordiamo che le funzioni di **hash** per definizione sono **one-way** e **deterministiche**, quindi non possiamo de-cifrarle e non necessitiamo di una chiave per cifrare i dati.

D'altro canto usiamo la cifratura quando i dati che vogliamo proteggere devono poter essere letti e ne vogliamo preservare la **confidenzialità**, bisogna prima concordare una **chiave** in maniera sicura, poi calcolare il **messaggio cifrato** con suddetta chiave e infine inviare il messaggio cifrato, il destinatario potrà de-cifrare il messaggio con la chiave concordata. In questo modo proteggiamo la **confidenzialità** e la **riservatezza** dei dati ma non la loro **integrità**.

### 3.1.5 La Criptografia non è la soluzione a tutti i problemi

**Perchè?** La crittografia non è la soluzione a tutti i problemi di sicurezza, in quanto comunque è sensibile a delle chiavi conservate su supporti digitali, i quali devono proteggere questa informazione. Inoltre da sola la crittografia non è mai usata come soluzione a problemi di sicurezza, ma sempre in combinazione con altri meccanismi di sicurezza.

## 3.2 Types of cryptography - Tipologie di crittografia

### 3.2.1 Cifrari di sostituzione

**In breve** I **cifrari di sostituzione** sono cifrari che sostituiscono un simbolo del *dizionario* con un altro simbolo. In questo contesto la *chiave* è la *sostituzione* dei simboli.

**In lungo** I **cifrari di sostituzione** sono dei **metodi di criptazione** per i quali ogni simbolo del **messaggio in chiaro** è rimpiazzato nel **messaggio cifrato** rispetto ad un prefissato sistema. I "simboli" possono essere lettere, coppie di lettere, triple di lettere, o anche una combinazione di questi, e anche altro. Il ricevente decifra il testo svolgendo le sostituzioni inverse.

#### Cifrario di cesare

Il **cifrario di Cesare** è un cifrario di sostituzione in cui il **dizionario** del testo in chiaro è spostato di un numero fisso di posizioni nell'alfabeto. Per decifrare il testo cifrato si sposta indietro dello stesso numero di posizioni.

Questo tipo di sistema viene anche chiamato: **rotazione di  $k$  posizioni**, in quanto il dizionario viene ruotato di  $k$  posizioni.



**Problemi e proprietà** Il cifrario descritto è il più semplice tra tutti come decodifica in quanto esistono solo 25 chiavi e con un semplice attacco di **brute force**. Inoltre se non si vuole usare un attacco di **brute force** si può comunque cercare di trovare lettere comuni del messaggio cifrato e provare a ricondurle a lettere comuni dell'alfabeto.



Figura 3.1: Esempio di cifrario di Cesare spostato di 3 posizioni

### Cifrario di Vigenère

Il **cifrario di Vigenère** è un cifrario che prevede l'uso di una "parola chiave", se necessario ripetuta per la lunghezza del messaggio, per poi calcolare la "somma" dei valori di ogni lettera del messaggio con la corrispondente lettera della parola chiave (prendendo poi il resto della divisione per 26). Quindi assegnando ad ogni lettera un numero da 0 a 25, si somma il numero della lettera del messaggio con il numero della lettera della parola chiave, si prende il resto della divisione per 26 e si assegna alla lettera corrispondente il numero ottenuto.

A	T	T	A	C	K	A	T	D	A	W	N	plaintext
L	E	M	O	N	L	E	M	O	N	L	E	keyword repeated to match plaintext length
L	X	F	O	P	V	E	F	R	N	H	R	ciphertext

Figura 3.2: Esempio di cifrario di Vigenère chiave "lemon"

Per decifrare il messaggio cifrato si sottrae il numero della lettera della parola chiave al numero della lettera del messaggio e si prende il resto della divisione per 26.

**Vantaggi** Questo cifrario è molto più sicuro rispetto a quello di Cesare in quanto come mostrato nell'esempio la lettera "A" viene codificata in quattro lettere diverse e la lettera "T" che è presente tre volte viene cifrata in due lettere corrispondenti. In quanto la parola chiave ha lunghezza "l" allora dimensione della chiave è  $l^{26}$ . Inoltre se la chiave è della stessa lunghezza del messaggio allora il cifrario di Vigenère è equivalente a un cifrario di sostituzione casuale il che rende impossibile la decifrazione, però ciò comporta ad una ulteriore difficoltà per il mittente e il destinatario nel concordare una chiave.

### 3.2.2 Cifrari di trasposizione

**In breve** I **cifrari di trasposizione** sono cifrari che permutano i simboli del messaggio in chiaro. In questo contesto la *chiave* è la *permutazione* dei simboli.

**In lungo** I **cifrari di trasposizione** sono dei **metodi di criptazione** per i quali i simboli del **messaggio in chiaro** sono permutati in un certo modo per ottenere il **messaggio cifrato**. Il ricevente decifra il testo svolgendo le permutazioni inverse.

#### Cifrario di trasposizione per colonne

Il **cifrario di trasposizione per colonne** è un cifrario che permuta i simboli del messaggio in chiaro per colonne, in modo che il messaggio cifrato sia una matrice di colonne.



Figura 3.3: Esempio di cifrario di trasposizione per colonne

Per decifrare il messaggio cifrato si riordinano le colonne in base alla chiave.

### 3.2.3 Criptografia Simmetrica

Come detto in precedenza la **criptografia simmetrica** prevede la stessa **chiave** per cifrare e de-cifrare i messaggi. La gestione di chi possiede le chiavi determina chi può accedere ai dati.

**Tipi** Esistono nella *criptografia moderna* due tipi di crittografia simmetrica:

**Cifrari a flusso** I **cifrari a flusso** cifrano una "breve" porzione di un blocco di dati alla volta con una chiave che varia nel tempo. Questa tipologia si basa su un **generatore di chiavi**, la cifratura è relativamente semplice (es. XOR) e molto spesso viene usato un bit per blocco.

**Cifrari a blocchi** I **cifrari a blocchi** cifrano un blocco di dati "lungo" alla volta con una chiave fissa. Questa tipologia è più complessa e richiede una funzione di cifratura e una di de-cifratura. Solitamente si usano blocchi di 64/128 bit.

#### Cifrari a flusso

I cifrari a flusso operano su un flusso di *testo in chiaro* e producono un flusso di *testo cifrato*. Lo stesso testo se ripetuto può essere cifrato in modo diverso in base al tempo nel quale è stato cifrato. I circuiti sono molto semplici e progettati per essere molto veloci.

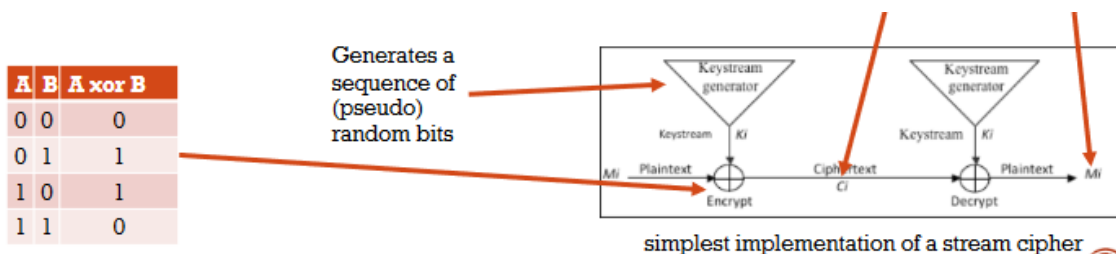


Figura 3.4: Esempio di circuito di cifratura a flusso

**Dove si usano** I cifrari a flusso sono usati in applicazioni in cui la velocità è fondamentale, come ad esempio nelle comunicazioni via radio o nella comunicazione in tempo reale.

### Cifrari a blocchi

Nei **cifrari a blocchi** viene trattata una sezione di *testo in chiaro* come una cosa unica per produrre un *testo cifrato* di egual lunghezza. La dimensione tipica di questa sezione è solitamente tra 64 e 128 bit. Un messaggio  $M$  più lungo di  $n$  bit viene diviso in blocchi di  $n$  bit e ogni blocco  $(M_1, M_2, \dots, M_i)$  viene cifrato separatamente con la stessa chiave  $k$ .

**Esempi** Alcuni esempi di cifrari a blocchi sono:

**DES Data Encryption Standard** è un cifrario a blocchi che opera su blocchi di 64 bit.

**AES Advanced Encryption Standard** è un cifrario a blocchi che opera su blocchi di 128 bit e utilizza chiavi di 128, 192 o 256 bit.

**DES** Il **Data Encryption Standard** è un cifrario a blocchi che opera su blocchi di 64 bit che venne progettato da IBM negli anni '70. Il cifrario è basato su una rete di sostituzione e trasposizione, e utilizza una chiave di 56 bit. Il DES è stato il cifrario più usato fino al 2000 (anche da ANSI<sup>1</sup> e in tre F.I.P.S.<sup>2</sup>), fino al 2005 quando è stato definitivamente dichiarato insicuro.

**Feistel (1973)** Il cifrario DES è basato su una struttura di **Feistel**, la quale prevede che il testo in chiaro venga diviso in due parti uguali, la prima passa attraverso una funzione di cifratura insieme ad una chiave, il risultato viene messo in **XOR** con la seconda parte del testo in chiaro, il risultato viene poi scambiato con la prima parte e il processo viene ripetuto per 16 volte con 16 chiavi diverse (o una chiave divisa in 16 parti). La decodifica avviene tramite lo stesso processo ma con le chiavi e le parti invertite.

**Vantaggi e svantaggi** Primo vantaggio di questo algoritmo è lo stesso meccanismo di cifratura e decifratura oltre alla semplicità, questo rende le componenti hardware molto più semplici e meno costose in quanto non necessitano di componenti diverse per cifrare e de-cifrare.

---

<sup>1</sup>American National Standard X3.92

<sup>2</sup>Federal Information Processing Standards