Report su Autenticazione ed Autorizzazione

Corso di Sicurezza dei Dati
Prof. Christian Esposito

UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA
DIPARTIMENTO DI ECCELLENZA

SOMMARIO

INTRODUZIONE

In questo documento viene presentato lo stato dell'arte relativo alla sicurezza, in particolare vengono analizzati gli standard e le principali soluzioni presenti in letteratura scientifica inerenti a sistemi di autenticazione e autorizzazione, verranno quindi mostrate metodologie e criteri che proteggono i sistemi da accessi non autorizzati.
Nella prima sezione verranno descritti gli aspetti relativi alla crittografia e il suo utilizzo nella realizzazione di certificati e Public Key Infrastracture. Successivamente verranno descritti i differenti metodi di autenticazione e autorizzazione introducendo i concetti chiave del dominio della sicurezza con le relative tecniche. Inoltre, verrà posta una certa attenzione all'adattamento di tali definizioni e concetti all'interno della tecnologia dei servizi Web presentando concetti relativi allo standard WS-Security.

Crittografia

La crittografia è il metodo mediante il quale le informazioni vengono convertite in un codice segreto che nasconde il vero significato delle informazioni, come illustrato in Figura 1. Il processo di codifica viene effettuato utilizzando un determinato algoritmo che prende in input una chiave di crittografia. Tale chiave indica come codificare le informazioni eseguendo una serie di trasformazioni e sostituzioni. L'algoritmo di codifica adottato deve essere sufficientemente potente da evitare che un avversario scopra la chiave di crittografia utilizzata conoscendo l'algoritmo adottato e alcune informazioni codificate. Il lato opposto esegue un altro algoritmo (essenzialmente l'algoritmo di codifica viene eseguito al contrario), fornendo come input una chiave di decrittazione, che in genere viene tenuta segreta e specifica le operazioni opposte da eseguire. Lo scambio sicuro delle chiavi adottate scambiate tra mittente e destinatario impedisce ad utenti non autorizzati di accedere alle informazioni. Dalla letteratura, possiamo distinguere tra schemi simmetrici o asimmetrici, rispettivamente se viene utilizzata un'unica chiave condivisa sia per la crittografia che per la decrittazione o se vengono adottate due chiavi distinte. Nella crittografia a chiave simmetrica è presente una sola chiave che viene usata

sia per cifrare che per decifrare e deve essere nota solo a chi può accedere al dato (quindi bisogna prevedere meccanismi sicuri per lo scambio della chiave). Questo tipo viene molto usato per garantire la confidenzialità sia nella conservazione che nella trasmissione dei dati, uno degli algoritmi più diffusi e sicuri di questo tipo è AES (Advanced Encryption Standard). Nella crittografia simmetrica, nota anche come crittografia a chiave pubblica, viene utilizzata una coppia di chiavi, una privata che deve essere generata localmente e tenuta segreta dal suo proprietario, e una pubblica che quindi è visibile a tutti. Le due chiavi devono essere generate in modo che una venga utilizzata per la cifratura e l'altra per la decifratura. Per garantire confidenzialità nello scambio di un messaggio, il mittente recupera ed utilizza la chiave pubblica del destinatario per cifrare il testo, lo invia e il ricevente lo decodifica utilizzando la sua chiave privata. Solo il destinatario previsto è in grado di decodificare il testo cifrato in quanto è l'unico a possedere la chiave privata richiesta. È possibile utilizzare la crittografia a chiave pubblica anche per fornire autenticità e/o integrità dei dati. Per fare questo, un utente cifra il dato utilizzando la sua chiave privata, e chiunque conosca la chiave pubblica dell'utente può decifrarlo. Se si riesce a decifrare con successo il testo utilizzando la chiave pubblica dell'utente, allora questo significa che solo lui può averlo cifrato in quanto proprietario della chiave privata associata, in questo modo si garantisce l'autenticità. Inoltre, solo l'utente può apportare modifiche al dato in quanto lui è l'unico in grado di cifrarlo con la sua chiave privata. Uno degli algoritmi più importanti che utilizza crittografia a chiave pubblica è RSA (Rivest, Shamir e Adleman).

Gli algoritmi a chiave pubblica sono computazionalmente onerosi e quindi poco indicati allo scambio di dati cifrati di grandi dimensioni. Gli algoritmi a chiave simmetrica offrono prestazioni migliori ma richiedono che le due parti che vogliono comunicare debbano condividere una chiave segreta, a differenza di come avviene nella crittografia a chiave asimmetrica. Per questo motivo, gli algoritmi a chiave pubblica sono molto diffusi nello scambio confidenziale di chiavi simmetriche, le quali vengono poi usate per la cifratura dei messaggi veri e propri.

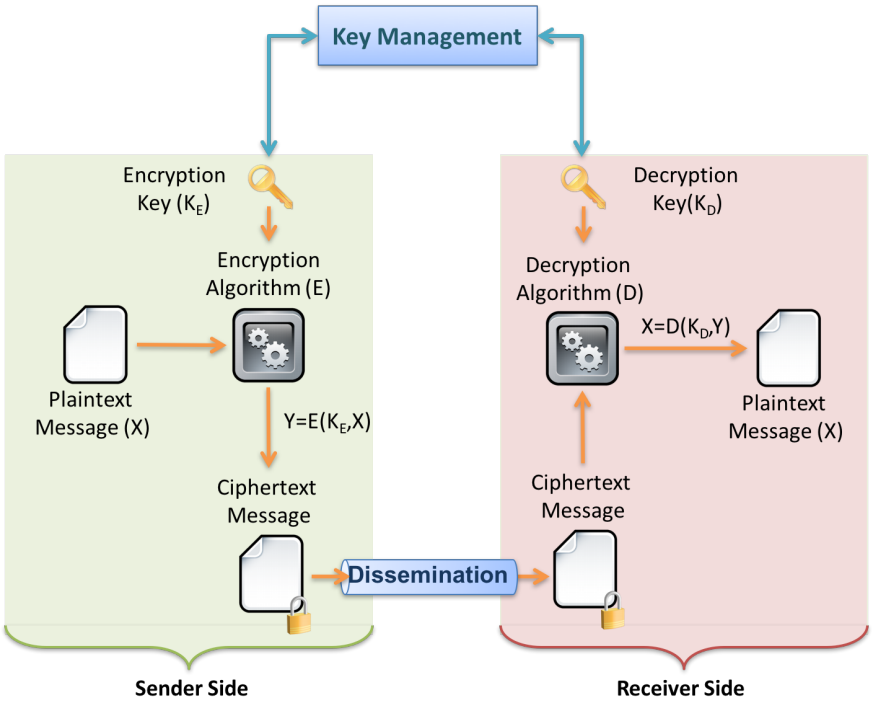

Figura 1: Un semplice modello di crittografia [2]

Un'altra tecnica per il miglioramento della sicurezza è la firma digitale, associata da un mittente a ogni messaggio generato, come illustrato nella Figura 2. Nello specifico, una firma è fondamentalmente un hash del contenuto dell'informazione crittografato utilizzando la chiave privata del publisher. Il subscriber può verificare l'integrità dell'informazione effettuando l'hashing del contenuto del messaggio ricevuto e confrontando tale valore con quello ottenuto decriptando la firma con la chiave pubblica del publisher. La verifica della firma è gestita da uno schema crittografico asimmetrico, dove il messaggio viene cifrato con la chiave privata del produttore del messaggio, mentre il destinatario decifra con la chiave pubblica del produttore del messaggio. Quando un'infrastruttura a chiave pubblica viene utilizzata per associare chiavi pubbliche alle rispettive identità utente, l'identità del firmatario è documentata da un certificato digitale valido, ovvero un artefatto che associa un'identità utente alla propria chiave pubblica, fornito e firmato da una Certificate Authority. Tale certificato viene inserito all'interno del messaggio firmato. A volte una firma digitale viene confusa con un Message Authentication Code (MAC), che sono piccole informazioni allegate ai messaggi al fine di garantirne l'autenticità e l'integrità. In particolare, MAC si affida alla crittografia simmetrica

per generare un tag crittografato collegato alle informazioni al fine di garantirne l'autenticità e l'integrità. Nonostante la differenza nello schema di crittografia adottato, le firme digitali e il MAC differiscono anche perché le firme digitali forniscono anche il non ripudio (utilizzando timestamp e/o nonces per dimostrare l'unicità e la freshness di un messaggio), mentre MAC no, e perché entrambi i publisher e il subscriber condividono la stessa chiave per gestire il MAC.

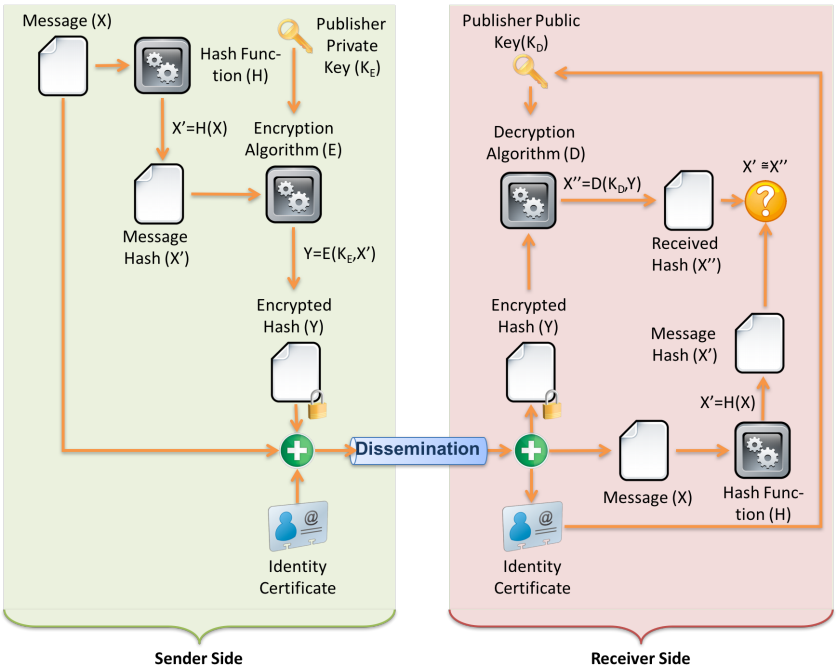

Figura 2: Un semplice modello di firma del messaggio [2]

X.509 e PKI

X.509 è un formato standard per certificati a chiave pubblica, ovvero documenti digitali che associano in modo sicuro coppie di chiavi crittografiche a identità come siti Web, individui o organizzazioni. Lo standard definisce sotto diversi aspetti l'utilizzo dei certificati nelle infrastrutture a chiave pubblica (PKI), ovvero un insieme di processi e mezzi tecnologici che consentono a terze parti fidate di verificare e/o farsi garanti dell'identità di un utente.

X.509 fu proposto dall'Unione internazionale delle telecomunicazioni ITU-T per la prima volta nel 1988 come parte degli standard X.500 per i servizi di directory elettronica. Il formato del certificato nello standard del 1988 è chiamato formato versione 1 (v1).

Quando X.500 è stato rivisto nel 1993, sono stati aggiunti altri due campi, ottenendo il formato versione 2 (v2).
Nello stesso anno con l'*Internet Privacy Enhanced Mail (PEM*) [rfc-1422] è stata proposta un'infrastruttura a chiave pubblica basata su X.509 (v1). L'esperienza fin qui acquisita fece emergere l'esigenza di un'ulteriore estensione, per rendere più dettagliati i certificati e favorirne l'interoperabilità dei sottosistemi con profili diversi, come Internet. Nel 1996 fu, quindi, completata la versione 3 (v3), dalla collaborazione d'ISO/IEC, ITU-T, e ANSI X9, che permette una maggiore flessibilità nelle strutture di creazione dei certificati X.509, con possibilità di usare topologie di PKI in stile mesh o bridge [rfc-4158], anche in una rete di fiducia peer-to-peer come quella di Open-PGP. Questo aiuta a superare la rigida struttura gerarchica dei PKI.
Lo standard che profila il formato e la semantica dei certificati X.509 v3 e degli elenchi di revoche di certificati X.509 v2 (CRL) per l'infrastruttura PKI è *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile* [rfc-5280].
Una PKI fornisce la struttura di base per un'ampia varietà di componenti, applicazioni, politiche e pratiche per combinare e ottenere le tre principali funzioni di sicurezza (integrità, autenticazione e non ripudio). Riassumendo, è una combinazione di strumenti, politiche e procedure hardware e software attraverso le quali viene fornita la sicurezza di base richiesta per comunicazioni sicure in modo che gli utenti che non si conoscono possano comunicare in modo sicuro attraverso una catena di fiducia. I certificati digitali sono una componente fondamentale nell'infrastruttura PKI in quanto agiscono come "passaporti digitali" legando la firma digitale dell'utente alla sua chiave pubblica.

Una PKI è composta da:

1. politiche di sicurezza (Security Policy);
2. certificate Authority (CA);
3. registration Authority (RA);
4. certificate repository;
5. applicazioni per PKI.

Una **politica di sicurezza** definisce la top level direction di un'organizzazione sulla sicurezza delle informazioni, nonché i processi e i principi per l'uso della crittografia. In genere include dichiarazioni su come l'organizzazione gestisce le chiavi e le informazioni preziose e imposta il livello di controllo richiesto per far corrispondere i livelli di rischio.
Alcuni sistemi PKI sono gestiti da Autorità di certificazione commerciale (CCA) o Trusted Third Party (TTP) e pertanto richiedono un Certificate Practice Statement (CPS). Si tratta di un documento dettagliato contenente le procedure operative su come deve essere applicata e supportata la politica di sicurezza. Questa include specifiche su come sono costruite e gestite le CA, come vengono emessi, accettati e revocati i certificati, come le chiavi saranno generate, registrate e certificate, dove saranno archiviate e messe a disposizione degli utenti.

La **CA** è un'entità che emette e revoca i certificati. Un server interno o un TTP (Trusted Third Party) come Entrust, Baltimore o VeriSign, possono fornire una funzione CA. Una CA fornisce la base di attendibilità per una PKI poiché gestisce i certificati di chiave pubblica per l'intero ciclo di vita del certificato.
I compiti delle CA sono:

1. emettere certificati vincolando l'identità di un utente o un sistema ad una chiave pubblica con firma digitale,
2. pianificare le date di scadenza dei certificati,
3. mantenere elenchi di revoca dei certificati (CRL),
4. assicurare che i certificati vengano revocati mediante pubblicazione di CRL (Certificate Revocation Lists).

Quando si implementa una PKI, un'organizzazione può gestire la propria CA o utilizzare i servizi di una CA commerciale o TTP.
Sebbene i principi della PKI siano gli stessi, attualmente esistono due principali modelli di implementazione commerciale che dipendono da chi è la CA:

1. **private CA**: i fornitori vendono un sistema PKI completo a un'organizzazione che diventa la propria CA ed è responsabile dell'emissione e della gestione dei certificati,
2. **public CA**: i certificati vengono acquistati da un'organizzazione CA pubblica.

Una **RA** fornisce l'interfaccia tra l'utente e la CA, inoltre autentica l'identità degli utenti e inoltra la richiesta di certificato alla CA. La qualità di questo processo di autenticazione determina il livello di attendibilità che può essere riposto nei certificati.
Ad esempio, se tutto ciò che una RA richiede è un indirizzo e-mail e un nome, il livello di fiducia che dovrebbe essere riposto in quel certificato sarebbe notevolmente inferiore rispetto alla richiesta di procedure di registrazione più rigorose.

Le **Certificate Repository** forniscono un meccanismo per conservare le chiavi, i certificati e le CRL. Il key recovery è una funzione richiesta per recuperare dati o messaggi quando una chiave è persa e una PKI fornisce questo servizio automatizzato di key recovery.

AUTENTICAZIONE E AUTORIZZAZIONE

In questa sezione saranno affrontate le problematiche relative agli accessi alle funzionalità messe a disposizione da un sistema software. In particolare, saranno definiti i concetti di autenticazione e autorizzazione, e mostrate metodologie e criteri che proteggono i sistemi da accessi non autorizzati.

Autenticazione

Nel contesto dei sistemi informatici, l'autenticazione è il processo per determinare se qualcuno o qualcosa è, in effetti, chi o cosa dice di essere. La tecnologia di autenticazione fornisce il controllo dell'accesso ai sistemi controllando se le credenziali di un utente corrispondono alle credenziali in un database di utenti autorizzati o in un server di autenticazione. In questo modo, l'autenticazione garantisce sistemi sicuri, processi protetti e sicurezza delle informazioni aziendali.
I termini autenticazione e autorizzazione sono spesso usati in modo intercambiabile, sebbene siano spesso implementati insieme, sono due funzioni distinte. L'autenticazione è il processo di convalida dell'identità di un utente registrato o di un processo prima di consentire l'accesso a reti e sistemi protetti. Mentre l'autorizzazione è un processo più granulare che convalida se all'utente o al processo autenticato è stata concessa l'autorizzazione ad accedere alla risorsa specifica che è stata richiesta. Il processo mediante il quale l'accesso a tali risorse è limitato a un certo numero di utenti è chiamato controllo dell'accesso. Il processo di autenticazione inizia quando un utente tenta di accedere alle informazioni e precede sempre il processo di autorizzazione.
Durante l'accesso ad alcuni servizi messi a disposizione dal Web è importante per l'utente definire in modo univoco la propria identità, essere riconosciuto e per questo ottenere l'accesso ai propri servizi. Allo stesso modo è fondamentale anche conoscere l'identità di chi si trova dall'altra parte della "linea" della comunicazione ed essere certi che l'interlocutore con il quale si stanno scambiando delle informazioni sia veramente chi dice di essere e non un impostore.

Con l'evoluzione degli attacchi informatici, sono stati sviluppati una serie di schemi di autenticazione degli utenti al fine di tentare di proteggerli. Esistono diversi tipi di autenticazione. A seconda dei casi d'uso per i quali viene utilizzata l'autenticazione, essa può consistere in SFA, 2FA o MFA. In Figura 3 sono schematizzati i tre tipi di autenticazione citati in precedenza.

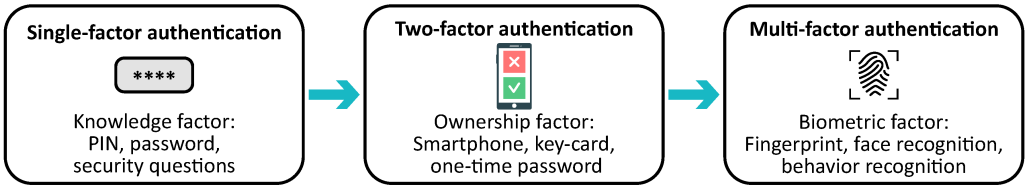

Figura 3: Tipi di autenticazione [12]

Al fine di identificare l'identità degli utenti vengono in genere utilizzati degli ID utente e l'autenticazione si verifica quando l'utente fornisce credenziali come una password che corrisponde al proprio ID utente. La pratica di richiedere un ID utente e una password è nota come autenticazione a fattore singolo (SFA). Negli ultimi anni, le aziende hanno rafforzato l'autenticazione richiedendo ulteriori fattori di autenticazione, come un codice univoco che viene fornito a un utente su un dispositivo mobile quando si tenta un accesso o una firma biometrica, come una scansione facciale o un'impronta digitale. Questo è noto come autenticazione a due fattori (2FA). Inoltre, vi è anche l'autenticazione a più fattori (MFA) che aggiunge più di un livello di autenticazione in modo tale da richiede agli utenti intenzionati ad avere un accesso alle possibili risorse non solo un ID utente e una password, ma anche altre informazioni di autenticazione aggiuntive in maniera tale da fornire un elevato livello di sicurezza.

Nel campo dell'information Technology, le organizzazioni utilizzano l'autenticazione per controllare chi ha accesso alle reti e alle risorse aziendali, nonché per identificare e controllare a quali macchine e server hanno accesso. Le aziende utilizzano anche l'autenticazione per consentire ai dipendenti remoti di accedere in modo sicuro alle proprie applicazioni e reti.

Per le aziende e altre grandi organizzazioni, l'autenticazione può essere eseguita anche utilizzando un sistema single sign-on semplificato, che garantisce l'accesso a più sistemi con un unico set di credenziali di accesso.

L'autenticazione di un utente con un ID utente e una password è generalmente considerata il tipo di autenticazione più semplice e dipende dal fatto che l'utente conosca due informazioni: l'ID utente o il nome utente e la password. Poiché questo tipo di autenticazione si basa su un solo fattore di autenticazione, è un tipo di SFA.

L'autenticazione forte è un termine generalmente utilizzato per descrivere un tipo di autenticazione più affidabile e resistente agli attacchi. L'autenticazione forte in genere utilizza almeno due diversi tipi di fattori di autenticazione e spesso richiede l'uso di password complesse contenenti almeno otto caratteri, un mix di lettere minuscole e maiuscole, simboli e numeri speciali.

Un fattore di autenticazione rappresenta un dato o un attributo che può essere utilizzato per autenticare un utente che richiede l'accesso a un sistema.

Un fattore è considerato una delle seguenti caratteristiche (Figura 4):

- qualcosa che un utente conosce,
- qualcosa che un utente possiede,
- qualcosa che un utente è,
- qualcosa che un utente fa,
- da qualche parte in cui si trova un utente o qualcosa che si trova nell'ambiente dell'utente,

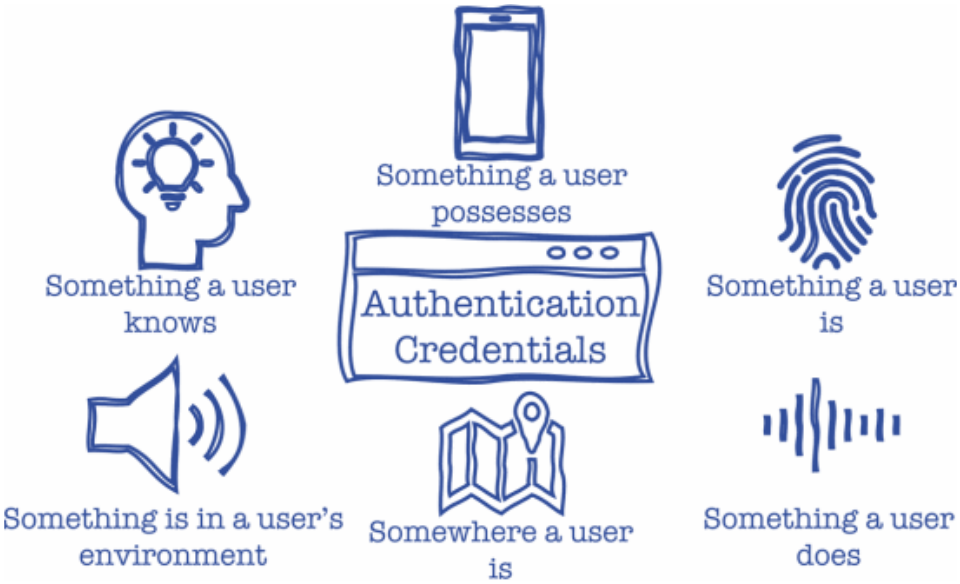

Figura 4: Credenziali di autenticazione [1]

Nelle sezioni seguenti, verranno descritti i vari tipi di fattori di autenticazione proposti dalla letteratura.

Qualcosa che un utente conosce

Quando un utente accede a un sistema informatico, la prima operazione di autenticazione prevede l'identificazione, che può essere eseguita fornendo un nome utente. Per autenticare un utente, il cliente deve dimostrare di essere l'effettivo proprietario di un profilo esistente, cosa che potrebbe essere fatta fornendo qualcosa che nessuno conosce tranne quell'utente, come un numero di identificazione personale (PIN), un nome utente, una password o la risposta a una domanda segreta.

Attualmente l'utilizzo di una password è il mezzo di più comune di autenticazione. Tuttavia, è spesso considerata una forma debole di protezione. Infatti, le password sono inclini a numerosi attacchi che utilizzano l'ingegneria sociale per estrapolare informazioni dell'utente preso di mira per poi applicare attacchi basati su dizionario o di forza bruta che potrebbero portare ad indovinarla facilmente. In risposta a questi attacchi, gli utenti dovrebbero rafforzare le proprie password. Per ottenere una maggiore sicurezza per le informazioni dell'utente, le password dovrebbero essere cambiate frequentemente, dovrebbero essere univoche e dovrebbero contenere lettere, numeri e caratteri speciali

come maiuscole e minuscole. Questo approccio però porterebbe l'utente ad avere password difficili da ricordare e dunque l'utente sarebbe impossibilitato ad accedere, per ovviare a queste problematiche tipicamente si utilizzano meccanismi per recuperare la propria password con l'aiuto della propria e-mail o numero di cellulare con cui si è registrato al sistema. Questo aiuta l'utente a riottenere temporaneamente l'accesso al sistema.

Qualcosa che un utente possiede

Il fattore di possesso, o qualcosa che l'utente ha, può essere qualsiasi credenziale basata su elementi che l'utente può possedere e portare con sé, inclusi dispositivi hardware, come un token di sicurezza o un telefono cellulare utilizzato per accettare un messaggio di testo o per eseguire un'app di autenticazione che può generare una password monouso (OTP) o un PIN. Poiché vengono utilizzati due tipi di autenticazione, ciò che un utente conosce (una password) e ciò che l'utente possiede (ad esempio, un dispositivo mobile che viene generalmente trasportato), questo tipo di autenticazione viene solitamente definito autenticazione a due fattori (2FA). Tuttavia, se un utente dovesse utilizzare solo uno dei fattori, sarebbe considerato SFA (autenticazione a fattore singolo). Al momento, i gadget comunemente utilizzati per autenticare gli utenti sono token, smart card e dispositivi mobili.

Token: un token è un piccolo dispositivo elettronico con una schermata di visualizzazione a diodo a emissione di luce (LED) che può essere utilizzato al posto di una password ed è una forma di autenticazione che dipende da ciò che un utente possiede. Il metodo in cui è possibile utilizzare un token è quello di ricevere una password una tantum (OTP) o generare un OTP. Esistono due tipi di OTP:

1. un OTP (TOTP) basato sul tempo,
2. un codice di autenticazione di messaggi basati su hash (HMAC) OTP (HOTP).

Un TOTP è un passcode temporaneo che cambia dopo un determinato periodo di tempo; di solito, un TOTP viene generato e visualizzato sullo schermo a LED del token per un tempo limitato. Poiché il token potrebbe non avere una connessione diretta all'entità di autenticazione, un algoritmo condiviso tra il token e l'entità di autenticazione

corrispondente potrebbe usare l'orario del giorno come uno dei suoi fattori di autenticazione. Quando si utilizza un token, un utente accede con un nome utente e una password, il codice viene mostrato sul token. Dopo che l'utente ha immesso il token visualizzato, l'entità di autenticazione lo riceve e verifica che siano identici. Il codice deve essere utilizzato all'interno del limite di tempo predefinito del token. Dopo che il limite di tempo del token è scaduto, non può più essere autenticato dall'entità di autenticazione e l'utente non è autorizzato ad accedere a un'altra sessione con quel token. Se l'utente ha già effettuato l'accesso con il token scaduto, l'utente viene disconnesso. I token hanno alcuni vantaggi rispetto alle password. Innanzitutto, una password è statica; non cambia a meno che l'utente non lo cambi (o sia costretto a farlo), il che offre a un attaccante la possibilità di recuperarla e successivamente usarla. In alternativa, un token genera spesso una password dinamica valida per una sola operazione in un tempo specifico, che crea una barriera all'accesso non autorizzato.

Smart Card: è una scheda di plastica che contiene un chip a microprocessore al cui interno sono salvati i dati relativi a un utente. Queste smart card possono essere utilizzate come identificazione personale, per l'autenticazione e persino per l'archiviazione dei dati. Le smart card possono essere implementate come schede di contatto ovvero che richiedono un contatto fisico o schede senza contatto, in cui le informazioni nel chip vengono trasferito elettronicamente. Tali schede utilizzano le tecniche Near-field communication (NFC) e Radio-frequency Identification (RFID).
L'RFID è una tecnologia di comunicazione wireless che viene utilizzata per identificare oggetti o persone taggate in modo univoco. Al momento, RFID viene utilizzato per autenticare gli utenti e prevenire frodi o accessi non autorizzati a un sistema informatico. L'NFC è una comunicazione wireless a corto raggio che consente ai dispositivi intelligenti di scambiare dati toccando o raggiungendo un certo raggio di vicinanza. Affinché due dispositivi intelligenti utilizzino la tecnologia NFC per scambiare informazioni, entrambi i dispositivi devono essere sintonizzati sulla stessa radio frequenza. Inoltre, entrambi i dispositivi devono essere nel raggio operativo, che in genere è compreso tra 3 e 5 cm. Tuttavia, una scheda è facile da perdere, il che lo rende vulnerabile a numerosi tipi di attacchi:

1. attacco di imitazione,

2. rubare la smart card,
3. attacco di offline password cracking,
4. attacco di mascheramento del server,
5. Ecc...

Inoltre, le smart card hanno i loro limiti poiché alcuni computer e dispositivi intelligenti non supportano il software Smart Card (ad esempio, lettori di smart card). Per incorporare una tecnica come le smart card, sono necessarie parti hardware che devono essere installate prima di utilizzarlo, che costerebbe denaro, tempo e sforzi.

Applicazione di **Smart Device**: in generale, un'applicazione Smart Device è semplicemente un'applicazione software progettata per essere eseguita su un dispositivo intelligente (ad es. Smartphone, smartwatch, tablet, ecc.). Sviluppare un'applicazione di dispositivi intelligenti che si comportano in un certo modo per autenticare un utente e dimostrare la propria identità a una risorsa è un qualcosa che è esistito per un decennio. L' applicazione di Smart Device è considerata molto comoda perché l'utente non ha bisogno di avere nient'altro che il proprio dispositivo intelligente. Inoltre, questa tecnica è onnipresente e facile da usare. Come qualsiasi altra tecnica, un'applicazione Smart Device ha i suoi svantaggi. Una volta che un utente perde o rompe il suo dispositivo intelligente, l'utente non può più autenticarsi per accedere ad una risorsa. Inoltre, se un dispositivo intelligente esaurisce la batteria, un utente non può utilizzarlo. I token software per applicazioni per dispositivi intelligenti sono vulnerabili all'attacco man-in-the-middle (MITM), infezione da malware, attacchi di phishing e reverse engineering.

Qualcosa che è un utente

Qualcosa che un utente è, dipende dalle caratteristiche di una persona che possono essere fisiche o non fisiche, dunque, si basa in genere su una qualche forma di identificazione biometrica, comprese impronte digitali, riconoscimento facciale, scansione della retina o qualsiasi altra forma di dati biometrici.
Questa forma di autenticazione è nota come autenticazione biometrica. L'autenticazione biometrica fisica viene definita biometria standard, mentre l'autenticazione biometrica non fisica viene definita biometria cognitiva.

Autenticazione biometrica standard: è un processo di sicurezza che si basa sulle caratteristiche fisiche distintive di un utente (ciò che è un utente) per verificare che un utente sia chi afferma di essere. L'autenticazione biometrica standard può utilizzare le impronte digitali, il riconoscimento del viso, il riconoscimento della retina, la geometria delle mani o il riconoscimento dell'iride; tuttavia, le impronte digitali sono più spesso utilizzate rispetto al riconoscimento del viso. Esistono due diversi tipi di scanner per impronte digitali: uno scanner di impronte digitali statico e uno scanner di impronte digitali dinamico. Con uno scanner di impronte digitali statico, un utente deve posizionare l'intero pollice o dito sullo scanner. Uno scanner di impronte digitali dinamico richiede a un utente di posizionare una parte di un dito su una piccola porzione di scanner.

Autenticazione biometrica cognitiva: è un metodo che riconosce le persone in base ai loro pensieri, percezioni e processi di osservazione usando le risposte dal tessuto nervoso. Il tessuto nervoso è una componente importante del sistema nervoso, che consiste in neuroni. Le risposte dal tessuto nervoso di un utente possono essere ottenute utilizzando molte tecniche; ad esempio, usando un elettroencefalogramma (EEG), un elettrocardiogramma (ECG) o da una risposta elettrodermica (EDR). Queste tecniche sono un modo per registrare l'attività cerebrale prodotta durante un'attività di pratica, che può essere un'attività mentale o fisica.

In generale, l'autenticazione biometrica presenta i suoi vantaggi. Il primo è una migliore qualità e sicurezza, l'utilizzo di una caratteristica unica del corpo umano per

l'autenticazione crea un ostacolo per un utente malintenzionato che deve avere accesso fisico al dispositivo per poter accedere. In secondo luogo, l'identità biometrica di un utente non può essere persa perché questa categoria di autenticazione dipende da una parte del corpo umano, in genere è difficile perderlo o non averla. Tuttavia, l'autenticazione biometrica ha anche i suoi svantaggi. Un sistema di autenticazione biometrica in genere è molto costoso economicamente. Questi costi potrebbero essere dovuti sia ai requisiti software che hardware e il sistema potrebbe persino richiedere costi aggiuntivi di manutenzione nel tempo. Il principale svantaggio di una tecnica di autenticazione biometrica è che una volta che un sistema con una firma biometrica è compromesso, un utente non può "ripristinare" o cambiare il proprio corpo.

Qualcosa che fa un utente

Questo tipo di fattore di autenticazione, nota come biometria comportamentale, si basa sull'identificazione e la misurazione di modelli basati sul comportamento di una persona durante l'esecuzione di determinate attività. La dinamica dei tasti (Keystroke Dynamics) e il riconoscimento vocale sono due esempi di biometria comportamentale.

Keystroke Dynamics: utilizza il riconoscimento dei modelli di ritmo e tempistica creati quando un individuo digita su una tastiera. Questo metodo utilizza molte misurazioni per studiare il comportamento di battitura di una persona, come un tempo di permanenza: la durata del tempo in cui viene premuto un tasto; tempo di volo: la durata del tempo tra il rilascio di un tasto e la pressione del tasto successivo; ed errore comune: gli errori che si verificano quando un utente digita un nome utente o una password.
I vantaggi della Keystroke Dynamics sono la sua convenienza (non richiede ulteriori parti hardware o addirittura un'interfaccia utente), comodità (un utente non ha bisogno di fare passi in più per essere autenticato) e affidabilità (perché questo metodo non può mai essere perso o eliminato). Keystroke Dynamics ha anche i suoi svantaggi; ad esempio, generalmente soffre di bassa precisione (il modello di battitura di un utente può cambiare a causa di lesioni, stanchezza o distrazione) e ha una bassa permanenza (il modello di battitura di un utente può cambiare gradualmente nel tempo perché può abituarsi a

digitare una password, adattarsi a un dispositivo di input o aumentare la loro competenza di battitura).

Riconoscimento vocale: questa tecnica analizza la voce di una persona per identificare le caratteristiche uniche da usare quando si verifica la loro identità. Diverse caratteristiche, come l'età, la forma e le dimensioni della bocca, la forma della testa e il movimento della mascella creano una voce unica per ogni persona. Come altri tipi di tecniche di autenticazione degli utenti, il riconoscimento vocale presenta vantaggi e svantaggi. Alcuni dei vantaggi del riconoscimento vocale includono praticità (ha un'elevata accettabilità sociale), efficacia in termini di costi (non è necessario hardware extra) ed è ampiamente accessibile (la maggior parte dei dispositivi sono dotati di un microfono, ad esempio). Tuttavia, gli svantaggi sono che può essere facilmente falsificato (la voce è soggetta a spoofing, un utente malintenzionato può registrare la voce di un utente e utilizzare la registrazione per l'autenticazione), c'è lo svantaggio del tempo (al fine di verificare l'identità di un utente, il sistema può richiedere un po' di tempo) e gli accenti possono influire sul risultato (il riconoscimento vocale potrebbe non essere in grado di riconoscere la voce di un utente che non è un madrelingua della lingua del sistema, per esempio).

Da qualche parte in cui si trova un utente

Il punto in cui si trova un utente si basa sulla posizione di un utente ed è noto come geolocalizzazione. Il fattore di localizzazione di solito non può essere utilizzato da solo per l'autenticazione, ma può integrare gli altri fattori fornendo un mezzo per escludere alcune richieste. Ad esempio, può impedire a un utente malintenzionato situato in un'area geografica remota di fingere di essere un utente che normalmente accede solo dalla propria casa o dall'ufficio nel paese di origine dell'organizzazione. L'autenticazione della geolocalizzazione viene utilizzata dalla maggior parte delle banche, ad esempio, per bloccare i trasferimenti di denaro dall'estero a meno che l'utente non abbia informato la banca di approvare in anticipo eventuali trasferimenti da un'altra geolocalizzazione. Questo metodo di autenticazione è utilizzato non solo dalle banche; molti siti Web non

consentono a un utente di ottenere l'accesso se la posizione dell'utente non è nella posizione consuetudinaria.
Un vantaggio di questa tecnica di autenticazione è che aiuta a identificare l'accesso non autorizzato nelle prime fasi determinando la posizione degli utenti tramite i loro indirizzi IP. Ad esempio, quando un utente utilizza un sistema con controlli di sicurezza della geolocalizzazione, se nella fase di registrazione, l'account online dell'utente è configurato in Italia, nel caso in cui un utente malintenzionato cerca di accedere all'account di quell'utente da un indirizzo IP situato al di fuori dell'Italia, il sistema rifiuterà la richiesta e avviserà l'utente legittimo. Tuttavia, questo metodo presenta uno svantaggio legato alla privacy degli utenti poiché questa tecnica necessita delle informazioni di una persona che potrebbero essere usate per monitorare gli utenti.

Qualcosa che si trova nell'ambiente dell'utente

Questo fattore, qualcosa che è nell'ambiente dell'utente, è stato utilizzato in una serie di schemi del 2FA. Qualcosa che si trova nell'ambiente dell'utente dipende da qualcosa di unico nell'area circostante all'utente, come il suono ambientale o le informazioni sui punti di accesso ambientali.

Informazioni sui punti di accesso ambientale: un punto di accesso è un componente hardware che aumenta in modalità wireless la copertura della rete trasmettendo periodicamente un messaggio ogni 102,4 ms in una rete locale (LAN). Il messaggio di trasmissione può essere raggiunto nell'intervallo di un punto di accesso e contiene dati e informazioni di rete che possono essere utilizzate per stabilire una connessione, condividere informazioni, individuare i dispositivi o per dimostrare l'identità di un utente. Questa tecnica dipende dall'utilizzo delle informazioni sui punti di accesso ambientale (ovvero identificatore del set di servizi (SSID), identificatore di set di servizi di base (BSSID), indicatore di resistenza al segnale ricevuto (RSSI)) per autenticare un utente richiede un effetto zero. Alcuni vantaggi di questa tecnica sono convenienti poiché utilizza ciò che un utente possiede (ad esempio, dispositivo intelligente degli utenti) e ciò che si trova nell'ambiente dell'utente (ovvero l'impronta Wi-Fi). Inoltre, il principale vantaggio di dipendere dall'utilizzo delle informazioni sui punti di accesso ambientale è che solo

quando un dispositivo si trova all'interno di un intervallo fisico limitato di ciascun punto di accesso, sarà in grado di ricevere tali informazioni. Ciò limita intrinsecamente il potenziale per lo spoofing o l'impersonificazione e rende sicura l'autenticazione del secondo fattore. Tuttavia, questa tecnica non può funzionare in assenza di punti di accesso; inoltre, non può funzionare se un utente accede da un desktop che non ha una scheda Wi-Fi.

Suono ambientale: si riferisce al rumore di fondo di un utente (ad esempio, rumore televisivo, rumore del traffico, allarmi, conversazioni, rumore bio-acustico per animali, suoneria del telefono, ecc.). La registrazione del rumore di fondo di un utente da più di un dispositivo può essere utilizzata per autenticare un utente. Quando si tratta del suono ambientale, in termini di autenticazione, si considerano due dispositivi che registrano da vicino, uno accanto all'altro. Pertanto, rendendo difficile per un attaccante ottenere un accesso non autorizzato. L'attaccante e l'utente normale dovrebbero trovarsi in una posizione fisica simile per ottenere un risultato simile.

Inoltre, quando si utilizzano due diversi dispositivi che sono vicini, i suoni saranno molto probabilmente simili; Significa che se un utente malintenzionato si trova nell'area e registra i suoni circostanti per l'autenticazione, dovrà fornire un sound bite simile a quello che l'utente aveva registrato in precedenza. Questo fa sì che l'attaccante si avvicini alla vittima perché a diverse distanze i sound bite saranno diversi quando si verifica un confronto per l'autenticazione.

Autorizzazione

L’autorizzazione è il passo successivo dopo che un utente è stato autenticato. È il processo che consente l’accesso alle risorse solamente a coloro che hanno i diritti di usarle. Durante l’autorizzazione vengono quindi valutati i privilegi dell’identità digitale associata all’utente autenticato e viene consentito, limitato oppure impedito l’accesso alla risorsa, applicando opportune regole stabilite in precedenza.
Per quanto riguarda l’autorizzazione degli utenti autenticati si associa un ruolo ai vari attori definendo delle regole di accesso alle varie risorse del sistema, siano esse funzioni o dati.

Access Control

Il controllo dell'accesso è una tecnica di sicurezza che regola chi o cosa può visualizzare o utilizzare le risorse in un ambiente informatico. È un concetto fondamentale nella sicurezza che riduce al minimo i rischi per l'azienda o l'organizzazione.
Esistono due tipi di controllo degli accessi: fisico e logico. Il controllo dell'accesso fisico limita l'accesso a campus, edifici, stanze e risorse IT fisiche. Il controllo di accesso logico limita le connessioni a reti di computer, file di sistema e dati.
Per proteggere una struttura, le organizzazioni utilizzano sistemi di controllo degli accessi elettronici che si basano su credenziali utente, lettori di schede di accesso, auditing e report per tenere traccia dell'accesso dei dipendenti a sedi aziendali riservate e aree proprietarie, come i data center. Alcuni di questi sistemi incorporano pannelli di controllo accessi per limitare l'accesso a stanze e edifici, nonché allarmi e funzionalità di blocco, per impedire accessi o operazioni non autorizzati.
I sistemi di controllo degli accessi logici eseguono l'autenticazione e l'autorizzazione per l'identificazione di utenti ed entità valutando le credenziali di accesso.

Componenti del controllo accessi

Ad alto livello, il controllo dell'accesso riguarda la limitazione dell'accesso a una risorsa. Qualsiasi sistema di controllo accessi, fisico o logico, ha cinque componenti principali:

1. **Autenticazione:** l'atto di provare un'asserzione, come l'identità di una persona o di un utente del computer. Potrebbe comportare la convalida di documenti di identità personale, la verifica dell'autenticità di un sito Web con un certificato digitale o il controllo delle credenziali di accesso rispetto ai dettagli memorizzati.
2. **Autorizzazione**: la funzione di specificare diritti di accesso o privilegi alle risorse. Ad esempio, il personale delle risorse umane è normalmente autorizzato ad accedere ai registri dei dipendenti e questa politica è solitamente formalizzata come regole di controllo dell'accesso in un sistema informatico.
3. **Accesso**: una volta autenticato e autorizzato, la persona o il computer possono accedere alla risorsa.

4. **Gestione**: la gestione di un sistema di controllo degli accessi include l'aggiunta e la rimozione dell'autenticazione e dell'autorizzazione di utenti o sistemi.
5. **Audit**: usato frequentemente come parte del controllo di accesso per far rispettare il principio del privilegio minimo. Nel tempo, gli utenti possono ritrovarsi con l'accesso di cui non hanno più bisogno, ad es. quando cambiano ruolo. Controlli regolari riducono al minimo questo rischio.

Importanza del controllo accessi

L'obiettivo del controllo degli accessi è ridurre al minimo il rischio per la sicurezza dell'accesso non autorizzato ai sistemi fisici e logici. Il controllo degli accessi è una componente fondamentale dei programmi di conformità alla sicurezza che garantisce che la tecnologia di sicurezza e le politiche di controllo degli accessi siano in atto per proteggere le informazioni riservate, come i dati dei clienti. La maggior parte delle organizzazioni dispone di infrastrutture e procedure che limitano l'accesso a reti, sistemi informatici, applicazioni, file e dati sensibili, come informazioni di identificazione personale e proprietà intellettuale.
I sistemi di controllo degli accessi sono complessi e possono essere difficili da gestire in ambienti IT dinamici che coinvolgono sistemi locali e servizi cloud.
A seconda dell'organizzazione, il controllo degli accessi può essere un requisito di conformità normativa:

1. **PCI DSS**: Il Payment Card Industry Data Security Standard è uno standard di sicurezza che protegge l'ecosistema delle carte di pagamento. Un sistema di controllo degli accessi è fondamentale per consentire o negare le transazioni e garantire l'identità degli utenti.
   Le organizzazioni dovranno limitare l'accesso fisico ai propri edifici per il personale in loco, i visitatori e i media, oltre a disporre di adeguati controlli logici dell'accesso per mitigare il rischio di sicurezza informatica di individui malintenzionati che rubano dati sensibili. Inoltre, le organizzazioni possono utilizzare soluzioni di sicurezza per tracciare e monitorare i propri sistemi in modo verificabile.

2. **HIPAA**: l'Health Insurance Portability and Accountability Act è stato creato per proteggere i dati sanitari dei pazienti dalla divulgazione senza il loro consenso. Il controllo dell'accesso è fondamentale per limitare l'accesso agli utenti autorizzati, garantire che le persone non possano accedere ai dati che vanno oltre il loro livello di privilegio e prevenire le violazioni dei dati.
3. **SOC 2**: Service Organization Control 2 (SOC 2) è una procedura di controllo progettata per i fornitori di servizi di terze parti per gestire i dati sensibili e prevenire violazioni dei dati, proteggendo la privacy di dipendenti e clienti. Garantisce che i fornitori proteggano la privacy dei propri clienti e richiede alle organizzazioni di implementare e seguire politiche e procedure rigorose relative ai dati dei clienti. I sistemi di controllo degli accessi sono fondamentali per far rispettare questi rigidi processi di sicurezza dei dati. Le aziende che desiderano ottenere l'assicurazione SOC 2 devono utilizzare una forma di controllo degli accessi con autenticazione a due fattori e crittografia dei dati. L'assicurazione SOC 2 è particolarmente importante per le organizzazioni che elaborano informazioni di identificazione personale (PII).
4. **ISO 27001**: l'International Organization for Standardization è uno standard di sicurezza delle informazioni che richiede alla direzione di esaminare sistematicamente i vettori di attacco di un'organizzazione e di controllare tutte le minacce informatiche e le vulnerabilità. L'implementazione dei controlli di accesso è fondamentale per conformarsi a questo standard di sicurezza. Inoltre, richiede un set completo di mitigazione del rischio o protocolli di trasferimento per garantire la sicurezza continua delle informazioni e la continuità aziendale.

Come funziona il controllo accessi

I controlli di accesso identificano un individuo o un'entità, verificano che la persona o l'applicazione sia chi o cosa afferma di essere e autorizzano il livello di accesso e la serie di azioni associate al nome utente o all'indirizzo IP. I servizi e i protocolli di directory, inclusi Lightweight Directory Access Protocol e Security Assertion Markup Language (SAML), forniscono controlli di accesso per autenticare e autorizzare utenti ed entità e consentire loro di connettersi alle risorse del computer, come applicazioni distribuite e server Web.

Le organizzazioni utilizzano diversi modelli di controllo degli accessi a seconda dei requisiti di conformità e dei livelli di sicurezza dell'IT che stanno cercando di proteggere.

Tipi di controllo accessi

I principali modelli di controllo accessi sono i seguenti:

Mandatory Access Control (MAC). Si tratta di un modello di sicurezza in cui i diritti di accesso sono regolati da un'autorità centrale basata su più livelli di sicurezza. Spesso utilizzate in ambienti governativi e militari, le classificazioni sono assegnate alle risorse di sistema e al sistema operativo o al kernel di sicurezza. Il MAC concede o nega l'accesso agli oggetti risorsa in base all'autorizzazione alla sicurezza delle informazioni dell'utente o del dispositivo. Ad esempio, Security-Enhanced Linux è un'implementazione di MAC su Linux.

Discretionary Access Control (DAC). Questo è un metodo di controllo dell'accesso in cui i proprietari o gli amministratori del sistema, dei dati o della risorsa protetti impostano le politiche che definiscono chi o cosa è autorizzato ad accedere alla risorsa. Molti di questi sistemi consentono agli amministratori di limitare la propagazione dei diritti di accesso. Una critica comune ai sistemi DAC è la mancanza di controllo centralizzato.

Role-Based Access Control (RBAC). Si tratta di un meccanismo di controllo degli accessi ampiamente utilizzato che limita l'accesso alle risorse del computer in base a individui o gruppi con funzioni aziendali definite, ad esempio livello esecutivo, livello tecnico 1, ecc., anziché sulle identità dei singoli utenti. Il modello di sicurezza basato sui ruoli si basa su una struttura complessa di assegnazioni di ruoli, autorizzazioni di ruolo e

autorizzazioni di ruolo sviluppate utilizzando l'ingegneria dei ruoli per regolare l'accesso dei dipendenti ai sistemi. I sistemi RBAC possono essere utilizzati per applicare i framework MAC e DAC.

Rule-based access control. Questo è un modello di sicurezza in cui l'amministratore di sistema definisce le regole che regolano l'accesso agli oggetti risorsa. Queste regole sono spesso basate su condizioni, come l'ora del giorno o il luogo. Non è raro utilizzare una qualche forma di controllo dell'accesso basato su regole e RBAC per applicare politiche e procedure di accesso.

Attribute-based access control. Si tratta di una metodologia che gestisce i diritti di accesso valutando un insieme di regole, politiche e relazioni utilizzando gli attributi degli utenti, dei sistemi e delle condizioni ambientali.

STATO DELL'ARTE DELLA SICUREZZA DEI SERVIZI WEB

Questa sezione ha lo scopo di introdurre i concetti chiave del dominio della sicurezza con le relative tecniche. Inoltre, dedicherà una certa attenzione all'adattamento di tali definizioni e concetti all'interno della tecnologia dei servizi Web presentando concetti relativi allo standard WS-Security.

WS-Security

WS-Security è uno standard composito realizzato combinando altre specifiche e metodi diversi. Specifica due diversi livelli di meccanismi per far rispettare il livello di sicurezza fornito, come mostrato nella Figura 5.

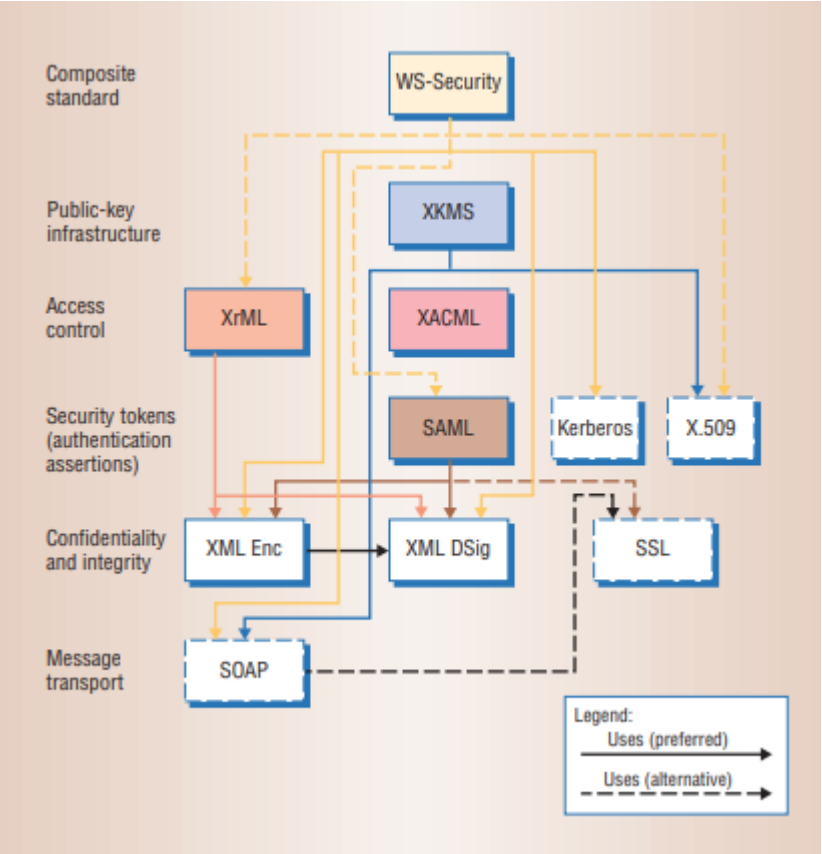

Figura 5 XML e Web services security standard e le loro dipendenze. [13]

Il primo è implementato a livello di messaggio, definendo un'intestazione SOAP che realizza estensioni per la sicurezza. Il secondo è realizzato a livello di servizio per eseguire meccanismi di sicurezza di livello superiore, come il controllo degli accessi o l'autenticazione.

In particolare, a livello di messaggio possiamo trovare due principali tecniche di sicurezza XML standard che possono essere introdotte nelle estensioni dell'intestazione SOAP: XML Signature (XML DSig nella figura) e XML Encryption (XML Enc nella figura). La prima mira a far firmare digitalmente una piccola porzione del contenuto XML (tale elemento è chiamato digest) in modo da fornire integrità e non ripudio per l'intero contenuto XML. Quest'ultima, invece, ha lo scopo di criptare una parte dell'intero contenuto XML utilizzando una determinata chiave, che può essere pubblica o privata a seconda della strategia di cifratura scelta. Nel caso di WS-Security, l'intestazione SOAP ha un determinato campo, chiamato DigestValue, per contenere il digest con le indicazioni del metodo di firma adottato. Se si utilizza XML Enc, l'intestazione SOAP deve

contenere la chiave di crittografia adottata, che viene a sua volta crittografata utilizzando un'apposita chiave pubblica.
Oltre a questi due importanti metodi a livello di messaggio, ne esiste un altro: Secure Socket Layer (SSL), che realizza una forma sicura del protocollo di trasporto TCP, offrendo meccanismi per l'accordo di chiave, la crittografia e l'autenticazione degli endpoint di una comunicazione orientata alla connessione.
A questi meccanismi a livello di messaggio si aggiungono quelli a livello di servizio. Il Security Assertion Markup Language (SAML) è un framework per lo scambio di informazioni di autenticazione e autorizzazione, modellate come i cosiddetti token di sicurezza, in modalità richiesta/risposta quando i partecipanti alla comunicazione non condividono la stessa piattaforma o non appartengono allo stesso sistema.
Il cuore di questo framework è l'asserzione, espressa come costrutto XML, contenente l'identità del richiedente e le decisioni di autorizzazione o le credenziali.
Pertanto, SAML trasmette il risultato di un processo di autenticazione, con l'asserzione che si presume essere un token di sicurezza per accedere a servizi protetti, concessi solo elaborando le informazioni presenti nell'asserzione, che si trovano nell'intestazione SOAP. SAML si occupa solo di formalizzare i token di sicurezza ed è abbinato ai meccanismi a livello di messaggio sopra citati per ottenere anche la riservatezza e l'integrità.
XACML viene utilizzato in modo che i ruoli e le politiche di accesso della prima organizzazione possano essere diffusi alla seconda. Così, quando la seconda organizzazione riceve un token di sicurezza emesso dalla prima, è in grado di riconoscerlo e di prendere la giusta decisione di accesso.
Infine, troviamo altre due specifiche: Extensible Rights Markup Language (XrML) e XML Key Management Specification (XKMS). Il primo è utilizzato per esprimere diritti e condizioni relative al controllo dell'accesso (come i tempi di scadenza); mentre il secondo definisce le interfacce per la distribuzione delle chiavi utilizzate in XML DSig e XML Enc.

Il WSS comprende diverse specifiche, ognuna delle quali si occupa di uno specifico aspetto avanzato della sicurezza, ed è illustrato in Figura 6

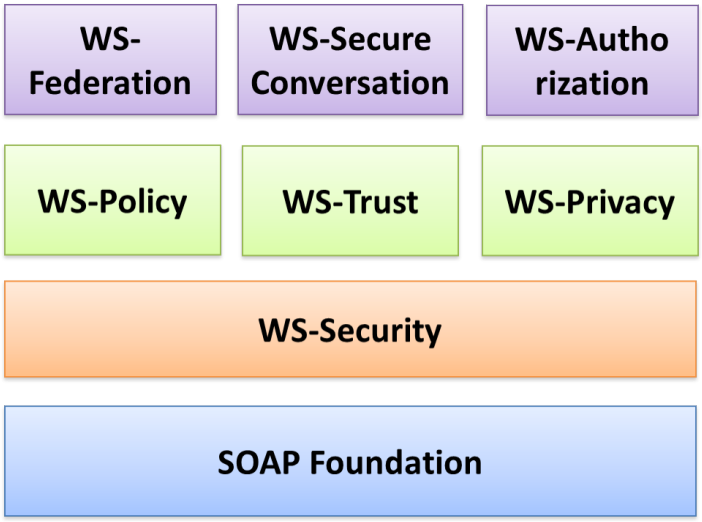

Figura 6 Web Service Security Framework

Lo standard Web Services Policy Framework (WS-Policy) specifica come esprimere i diversi tipi di politiche che un dato servizio web adotta per mezzo di un'asserzione di politica, che indica un requisito, o una capacità, di un soggetto, e si prevede che siano definite in specifiche separate, specifiche del dominio. Per quanto riguarda la sicurezza, le asserzioni da utilizzare sono definite nello standard Web Services Security Policy Language (WS-SecurityPolicy). La specifica WS-Trust ha il compito di estendere WS-Security in modo da gestire l'emissione, il rinnovo e la convalida dei token di sicurezza, nonché di gestire le relazioni di fiducia dirette e mediate tra i diversi domini di sicurezza attraverso la creazione di servizi di emissione di token di sicurezza. Infine, WS-Privacy fornisce un modello per la trasmissione delle preferenze e delle pratiche organizzative in materia di privacy.

Oltre a questi standard, altre specifiche compongono la parte superiore del framework: (i) il Web Services Federation Language (WS-Federation) estende WS-Trust per fornire un'architettura flessibile di identità federata; (ii) il Web Services Secure Conversation Language (WS-SecureConversation) estende la sicurezza del singolo messaggio fornita da WS-Security a una conversazione di più messaggi correlati; e (iii) WS-Authentication indica come esprimere le politiche di autorizzazione e gestire i dati di autorizzazione. Di questi standard costruiti sulla base di WS-Security, non tutti hanno ottenuto la stessa considerazione e lo stesso successo, e alcuni sono stati abbandonati: XACML ha

sostituito WS-Authentication, mentre WS-Privacy non è ancora stato pubblicato da OASIS.

Standard di sicurezza web per controllo accessi

WebAccessControl (WAC)

WebAccessControl (WAC) è un sistema decentralizzato per consentire a diversi utenti e gruppi varie forme di accesso alle risorse, in questo sistema gli utenti e i gruppi sono identificati da URI HTTP.

Il sistema è simile al sistema di controllo degli accessi utilizzato in molti file system, tranne per il fatto che i documenti controllati, gli utenti e i gruppi sono tutti identificati da URI. In particolare, gli utenti sono identificati da WebID (per identificare in modo univoco una persona, un'azienda, un'organizzazione o un altro agente utilizzando un URI), mentre i gruppi di utenti sono identificati dall'URI di una classe di utenti che, se la si cerca, restituisce un elenco di quelli della classe. Ciò significa che una persona ospitata da qualsiasi sito può essere membro di un gruppo ospitato da qualsiasi altro sito.

Tramite il WAC si concede l'accesso a un documento su un sito a utenti e gruppi ospitati da altri siti. Gli utenti non hanno bisogno di avere un profilo sul sito per avere accesso ai documenti su di esso.

Per poter condividere un codice comune viene proposta un'ontologia comune che fornisce i termini necessari per la memorizzazione delle liste di controllo accessi.

Ogni richiesta per una risorsa Web restituisce un documento HTTP contenente un'intestazione Link a una risorsa ACL che descrive l'accesso alla risorsa data e potenzialmente ad altre, come mostrato da questo diagramma in Figura 7.

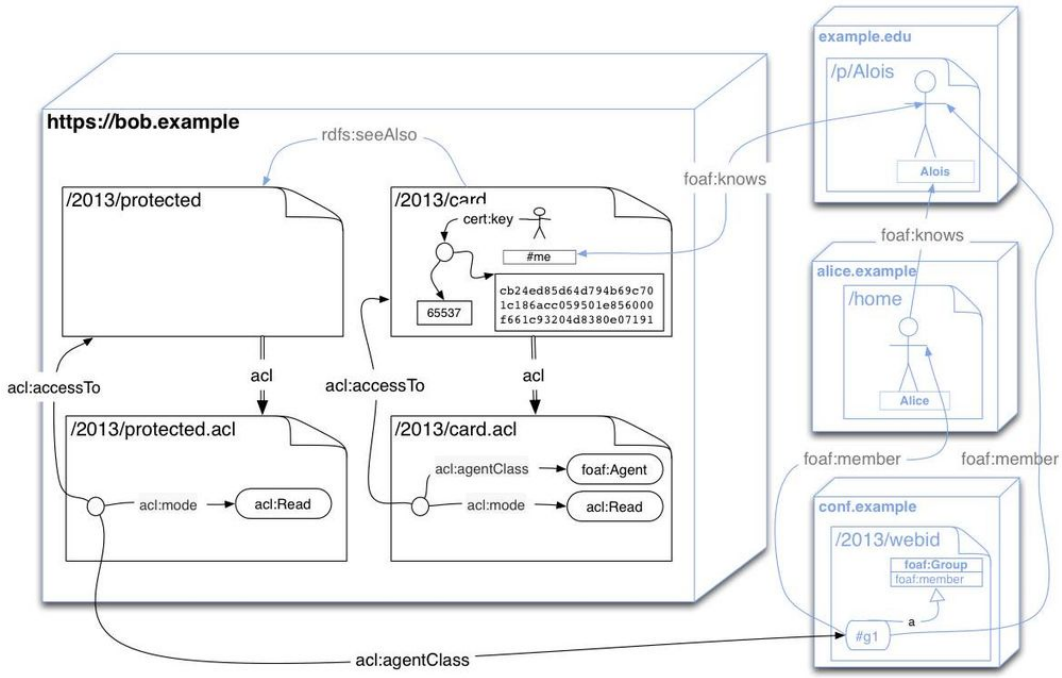

Figura 7 Diagramma WAC

User-Managed Access (UMA)

I protocollo User-Managed Access (UMA) è progettato per fornire a un utente web un punto di controllo unificato per autorizzare chi e cosa può ottenere l'accesso ai propri dati personali online (come gli attributi di identità), ai contenuti (come le foto) e ai servizi (come la visualizzazione e la creazione di aggiornamenti di stato), indipendentemente da dove si trovano tutte queste cose sul Web. Inoltre, UMA consente all'utente di presentare richieste al fine di verificarne l'idoneità e ricevere l'autorizzazione.

UMA è una nuova soluzione di gestione degli accessi che consiste in un'architettura e un protocollo di delega del controllo degli accessi basato su OAuth 2.0.

Di seguito sono riportati alcuni requisiti chiave dell'UMA:

1. **servizio di relazione di accesso dedicato**: il controllo di accesso dovrebbe essere esternalizzato dalle applicazioni Web e fornito come servizio online dedicato. Tale servizio dovrebbe consentire a un utente di controllare la condivisione dei dati e le relazioni di accesso ai servizi tra l'hosting dei servizi online e l'accesso ai dati. Tutti questi servizi dovrebbero poter risiedere in domini distinti e stabilire relazioni tra loro in modo dinamico,
2. **politiche guidate dall'utente**: un individuo dovrebbe essere in grado di configurare le proprie politiche richieste per il servizio di relazione di accesso per prendere decisioni di controllo dell'accesso. Pertanto, un utente dovrebbe essere in grado di applicare la stessa politica su risorse Web distribuite, utilizzando un'esperienza di interazione coerente,
3. **supporto per il controllo degli accessi basato sui reclami**: per adattarsi bene all'ambiente web altamente dinamico e aperto, il controllo degli accessi non dovrebbe basarsi esclusivamente sull'identificazione e l'autenticazione preliminari dei richiedenti; i server potrebbero non conoscere le possibili identità dei client che accedono ai dati in anticipo. I criteri dovrebbero consentire a un utente di definire le proprietà che i client devono possedere prima che l'autorizzazione possa essere concessa. Inoltre, un utente dovrebbe essere in grado di imporre condizioni contrattuali che regolano i diritti di accesso, nonché l'archiviazione dei dati, l'ulteriore utilizzo e l'ulteriore condivisione da parte di chi richiede i servizi,
4. **gestione utente del controllo accessi**: un individuo dovrebbe essere in grado di modificare le condizioni di accesso e terminare facilmente i rapporti. Dovrebbe inoltre essere possibile controllare e monitorare vari aspetti di tali relazioni in qualsiasi momento. Se lo desidera, l'utente non dovrebbe essere coinvolto direttamente nelle interazioni tra i servizi che accedono ai dati e i servizi che ospitano i dati. Piuttosto, dovrebbe essere possibile guidare tali interazioni applicando solo le politiche definite dall'utente.

Per soddisfare questi e altri principi e requisiti di progettazione documentati, UMA ha l'architettura illustrata di seguito in Figura 8.

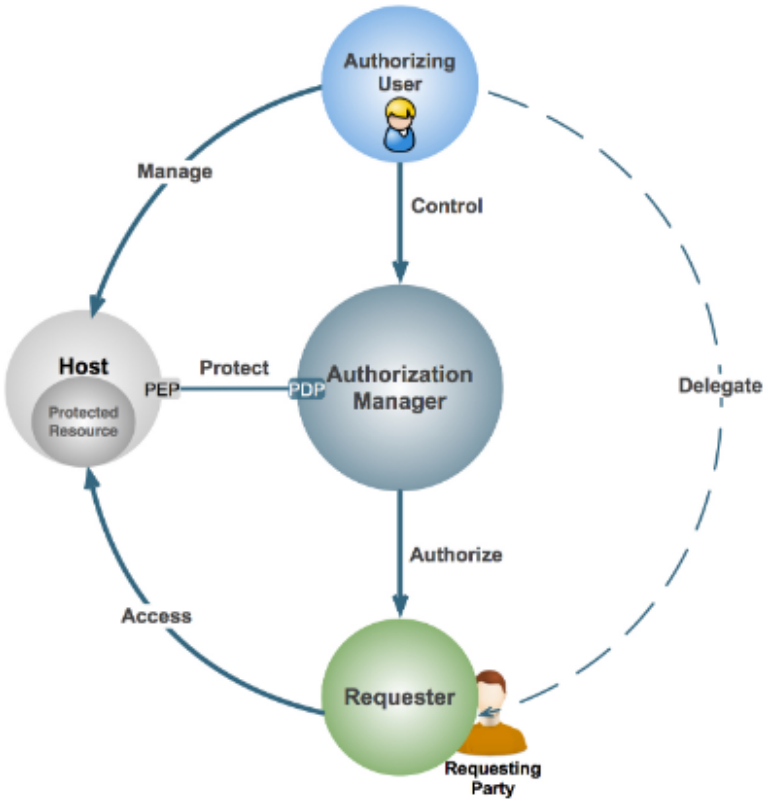

Figura 8 Architettura UMA

Un utente autorizzatore provvede a esternalizzare la protezione delle risorse dall'insieme scelto di host di risorse (punti di applicazione delle politiche) a un gestore delle autorizzazioni o AM (punto di decisione e amministrazione delle politiche), configurando quest'ultimo con politiche che controllano il modo in cui prende le decisioni sulla delega dell'autorizzazione di accesso quando un richiedente tenta di accedere a una risorsa protetta.

Json Web Token JWT

JSON Web Token (JWT) è uno standard aperto [RFC 7519] che definisce un modo compatto e autonomo per la trasmissione sicura di informazioni tra le parti come un oggetto JSON. Queste informazioni possono essere verificate e affidabili perché sono firmate digitalmente. I JWT possono essere firmati utilizzando un segreto (con l'algoritmo HMAC) o una coppia di chiavi pubblica/privata utilizzando RSA o ECDSA.
Le caratteristiche principali di un token JWT sono la self-contained e la compattezza.

Quest'ultima proprietà ci dà la possibilità di inviarlo tramite un URL, un parametro POST o all'interno di un'intestazione HTTP. Inoltre, la sua trasmissione è veloce. Inoltre, è self-contained poiché il payload contiene tutte le informazioni richieste sull'utente, per evitare di interrogare il database più di una volta.
I token firmati possono verificare l'integrità delle attestazioni in esso contenute, mentre i token crittografati nascondono tali attestazioni garantendo la confidenzialità. Quando i token vengono firmati utilizzando coppie di chiavi pubblica/privata, la firma certifica anche che solo la parte che detiene la chiave privata è quella che l'ha firmata.
I token JWT possono essere utili per l'autorizzazione e sono un buon modo per trasmettere informazioni in modo sicuro tra le parti.

Struttura di un JSON Web Token

Un JSON Web Token è costituito da tre parti separate da punti (.) che sono:

1. header,
2. payload,
3. signature.

Pertanto, un JWT in genere sarà simle al seguente: (Header).(Payload).(Signature)
Nel dettaglio verranno descritte le tre componenti.

Header

L'header o intestazione è in genere composta da due parti: il tipo di token, che è JWT, e l'algoritmo di firma utilizzato, come HMAC SHA256 o RSA.
Per esempio, utilizzando il formato JSON codificato con Base64Url verrà costruita la prima parte del JWT:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Payload

La seconda componente del token JWT che segue l'header è il **payload**, che conterrà i claims. I claims sono le dichiarazioni su un'entità (in genere l'utente) e dati aggiuntivi.
Esistono tre tipi di claims:

1. **Redistered claims**, si tratta di un insieme di claims predefiniti per fornire un insieme di attestazioni utili e interoperabili. Alcuni di essi sono:
    - iss(issuer)
    - exp(expiration time)
    - sub(subject)
    - aud(audience)
    - ecc...
2. **Public claims**, possono essere definiti a piacimento di coloro che utilizzano il JWT. Ma per evitare collisioni, dovrebbero essere definiti nel registro dei token Web JSON di IANA o essere definiti come un URI che contiene uno spazio dei nomi resistente a collisioni.
3. **Privete claims**, sono dei claims personalizzati creati per condividere informazioni tra le parti che concordano sull'utilizzo delle stesse e tali informazioni non sono claims public o registered.

Une esempio di payload codificato in Base64Url per formare la seconda parte del token Web JSON:

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

Signature

La **firma** (o signature) viene utilizzata per verificare che il messaggio non sia stato modificato lungo il percorso e, nel caso di token firmati con una chiave privata, è possibile anche verificare che il mittente del JWT sia chi dice di essere. Per creare la parte della signature bisogna prendere l'intestazione codificata, il payload codificato, un segreto e l'algoritmo specificato nell'header che firmerà la signature.
Ad esempio, se si desidera utilizzare l'algoritmo HMAC SHA256, la firma verrà creata nel modo seguente:

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret)
```

L'output è costituito da tre stringhe URL Base64 separate da punti che possono essere facilmente passate in ambienti HTML e HTTP, pur essendo più compatte rispetto agli standard basati su XML come SAML.
Di seguito è mostrato un JWT con l'intestazione e il payload precedenti codificati ed è firmato tramite un secret.

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4
gRG9lIiwiaXNTb2NpYWwiOnRydWV9.
4pcPyMD09olPSyXnrXCjTwXyr4BsezdI1AVTmud2fU4
```

Figura 9 Esempio Token JWT

Funzionamento dei JSON Web Token

Quando un utente accede a un server, il server autentica l'utente e invia un token con la risposta all'utente. Il server non memorizza sul server alcuna informazione relativa all'utente. Il client invia il token in ogni richiesta poiché HTTP è stateless. Ciò rende il server completamente stateless, il che aiuta nella scalabilità. Ogni volta che l'applicazione utente deve accedere ad altri domini o applicazioni di terze parti, il server verifica se il token è scaduto o meno. In caso contrario, l'applicazione può continuare ad accedere alle risorse, altrimenti l'applicazione dovrà autenticarsi nuovamente con il server e quindi accedere alle risorse.

Il diagramma seguente (Figura 10) mostra come ottenere e utilizzare un JWT per accedere ad API o a risorse:

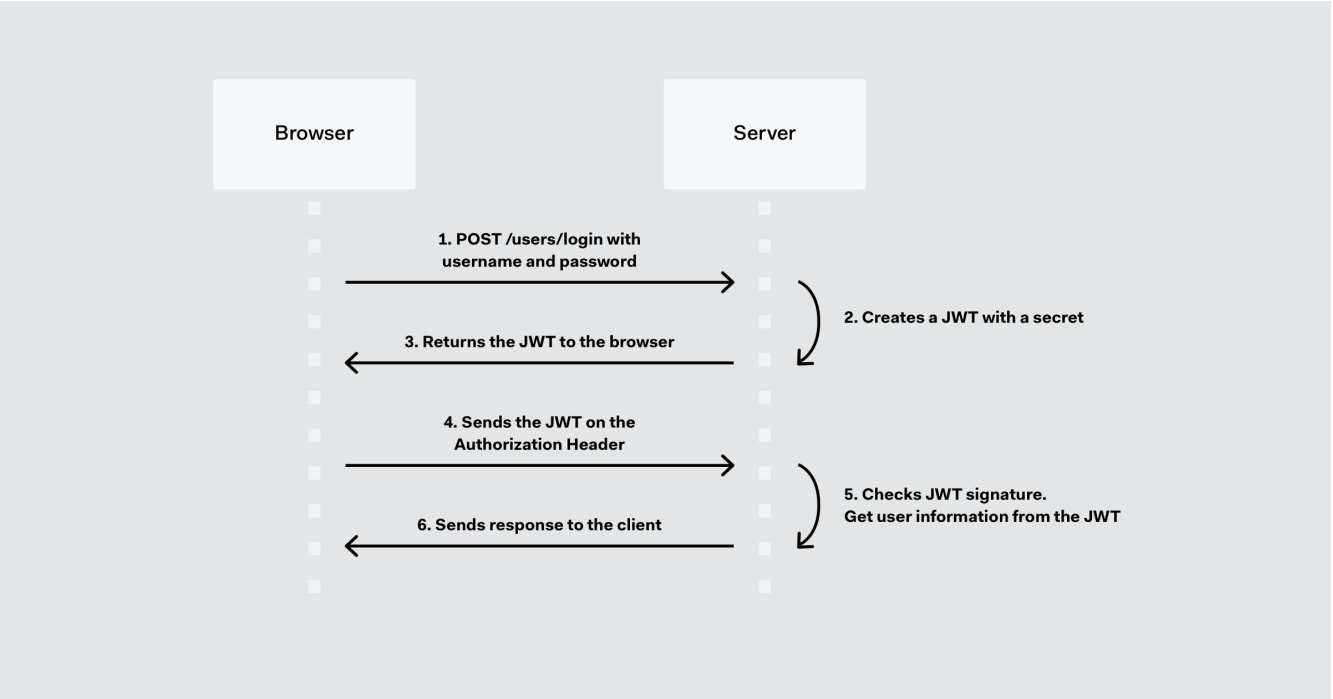

Figura 10 richiesta risorse con utilizzo JWT

1. la client request richiede l'autorizzazione al authorization server,
2. quando l'autorizzazione viene concessa, l'authorization server restituisce il token JWT di accesso per l'applicazione al client che l'ha richiesta,
3. il client utilizza il token di accesso per accedere a una risorsa protetta,
4. il server a questo punto controlla se il JWT è valido, risolve il token con il segreto e ottiene le informazioni utente da esso,
5. il server invia la risposta al client permettendolo di accedere alla risorsa.

Sistemi di Autorizzazione

XML Security

Gli standard di sicurezza XML definiscono i vocabolari XML e le regole di elaborazione per soddisfare i requisiti di sicurezza.
Questi standard utilizzano tecnologie crittografiche e di sicurezza legacy, nonché tecnologie XML emergenti, per fornire una soluzione flessibile, estensibile e pratica al fine di soddisfare i requisiti di sicurezza. Gli standard di sicurezza XML includono la firma

digitale XML per l'integrità, la crittografia XML per la riservatezza, la gestione delle chiavi XML (XKMS) per la registrazione, l'ubicazione e la convalida della chiave pubblica, lo standard security assertion markup language (SAML) per trasmettere l'autenticazione, l'autorizzazione e le asserzioni degli attributi.
La firma XML è la sintassi XML specifica che fornisce l'integrità del messaggio, l'autenticazione del messaggio e l'autenticazione del firmatario.
Tramite essa è possibile firmare sia parti di documenti che l'intero documento. Inoltre, utilizzando una sola firma XML è possibile firmare diverse risorse come dati di testo, dati binari e così via. La firma XML può essere applicata a tutti i campi XML. Il contesto più utilizzato è quello del e-commerce e della trasmissione di dati, dove è necessario proteggere i documenti e i dati digitali XML.
La firma XML risolve diversi problemi di sicurezza e migliora l'interoperabilità tra i sistemi di firma. Allo stesso modo, la crittografia e l'autenticazione dei dati sono una parte importante dei servizi web, per i quali possiamo risolvere il problema della sicurezza in modo efficiente applicando la firma XML.

Il linguaggio di markup per il controllo degli accessi (eXtensible Access Control Markup Language) (XACML) è un linguaggio di descrizione delle politiche di controllo degli accessi proposto dai comitati OASIS (Organization for the Advancement of Structured Information Standards) ed è uno schema XML standard per rappresentare le politiche di controllo degli accessi relative alle politiche di autenticazione e autorizzazione. XACML rappresenta le regole che specificano chi, cosa, quando e come è possibile accedere alle informazioni.
Oltre a definire un linguaggio per le politiche, XACML fornisce anche un formato di richiesta e di risposta per interrogare il sistema di policy. Ciò consente al processore XACML di conoscere il risultato dell'esecuzione dopo aver valutato l’accesso in base alla politica di autenticazione. XACML può essere applicato in tutti i campi in cui è necessario affrontare il problema della sicurezza controllo degli accessi.

SAML

SAML è uno standard open source basato su XML e definito da OASIS che consente di condividere informazioni relative a identità tra domini di sicurezza. La specifica SAML,

pur essendo principalmente mirata a fornire il single sign-on (SSO) tra domini web, è stata anche progettata per essere modulare ed estensibile al fine di facilitare l'uso in altri contesti.
SAML 2.0, la versione attuale, è stata pubblicata come standard OASIS nel 2005.

SAML definisce tre categorie di entità:

1. utenti finali. Un utente finale è una persona che deve essere autenticata prima di poter utilizzare un'applicazione,
2. service Provider o fornitore di servizi. Un fornitore di servizi è qualsiasi sistema che fornisce servizi, in genere i servizi per i quali gli utenti richiedono l'autenticazione, comprese le applicazioni Web o aziendali.
3. identity Provider o fornitore di identità. Un Identity Provider amministra le informazioni relative all'identità.

Lo scopo principale di SAML è definire il linguaggio di markup utilizzato per standardizzare la codifica dei dati di autenticazione per lo scambio tra sistemi. Include anche tutti i protocolli e i binding associati che utilizzano messaggi conformi a SAML per scambiare asserzioni di sicurezza tra utenti finali, fornitori di servizi e fornitori di identità.

SAML incorpora quattro diversi tipi di componenti:

- le asserzioni SAML sono dichiarazioni di identità, autenticazione e autorizzazione. Sono formattate utilizzando i tag XML specificati in SAML,
- i protocolli SAML definiscono il modo in cui le diverse entità richiedono e rispondono alle richieste di informazioni sulla sicurezza. Come le asserzioni SAML, questi protocolli sono codificati con tag XML specificati in SAML,
- i binding SAML sono i formati specificati per i messaggi del protocollo SAML da incorporare e trasportare su diversi meccanismi di trasmissione,
- i profili SAML determinano il modo in cui le asserzioni, i protocolli e i binding SAML vengono utilizzati insieme per l'interoperabilità in determinate applicazioni.

Secondo la specifica del protocollo di base SAML, un'asserzione SAML è un'unità di informazioni che fornisce zero o più dichiarazioni fatte da un'autorità SAML. Le autorità

SAML sono qualsiasi sistema che genera asserzioni di autenticazione SAML. I fornitori di identità SAML sono esempi di queste autorità.

SAML specifica tre tipi di asserzioni:

- un'asserzione di autenticazione indica che l'oggetto dell'asserzione è stato autenticato. Include l'ora, il metodo di autenticazione e il soggetto da autenticare,
- un'asserzione di attributo associa il soggetto dell'asserzione con gli attributi specificati. Un attributo SAML specificato è un attributo che fa riferimento a un'informazione definita relativa all'oggetto dell'autenticazione,
- un'asserzione di decisione di autorizzazione indica se la richiesta di un soggetto di accedere a una risorsa è stata approvata o rifiutata.

SAML definisce i propri protocolli generalizzati per le interazioni di richiesta/risposta tra i sistemi e le entità che possono essere autenticate, ovvero i committenti o i soggetti. I protocolli SAML 2.0 includono i seguenti protocolli:

- il protocollo di richiesta di autenticazione definisce le richieste di asserzioni di autenticazione e le risposte valide a tali richieste. Questo protocollo viene utilizzato quando una richiesta inviata da un utente a un fornitore di servizi deve essere reindirizzata a un fornitore di identità,
- il Single Logout Protocol definisce una tecnica che consente di terminare quasi contemporaneamente tutte le sessioni attive di un utente. Questa funzionalità è importante per le implementazioni SSO che richiedono la terminazione delle sessioni con più risorse quando l'utente si disconnette,
- Assertion Query and Request Protocol definisce le richieste di asserzioni di autenticazione nuove ed esistenti,
- Artifact Resolution Protocol definisce come richiedere e trasmettere i messaggi del protocollo SAML utilizzando un valore identificativo o un artefatto. Questo approccio semplifica lo scambio di messaggi di protocollo specifici,
- Name Identifier Management Protocol definisce un meccanismo che consente a un fornitore di identità di gestire il proprio nome cambiando l'identificatore del nome

e il suo formato o di notificare ad altre entità SAML che un identificatore del nome è stato terminato,

- Name Identifier Mapping Protocol definisce un meccanismo per la mappatura di un identificatore utente tra diversi fornitori di servizi.

Questi protocolli di request/response sono definiti come parte di SAML per consentire ai sistemi di richiedere l'autenticazione, rispondere alle richieste di autenticazione e scambiare asserzioni SAML. Questi protocolli sono indipendenti dai protocolli di rete a cui i messaggi SAML sono legati per il trasporto in rete.

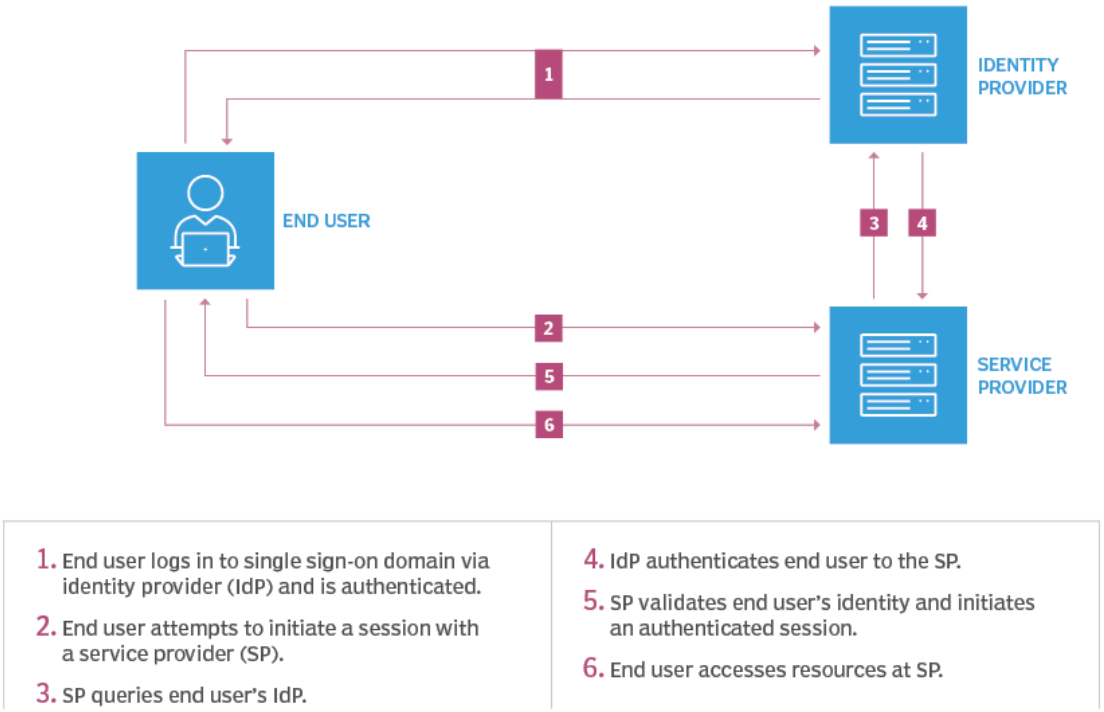

Figura 11 Protocolli SAML generalizzati per le interazioni tra i sistemi e le entità.

OAuth

OAuth (Open Authorization) è un framework di autorizzazione standard open source per l'autorizzazione basata su token nel contesto Internet. OAuth consente alle informazioni sull'account di un utente finale di essere utilizzate da servizi di terze parti, come Facebook e Google, senza esporre le credenziali dell'account dell'utente a terze parti. Agisce come intermediario per conto dell'utente finale, fornendo al servizio di terze parti un token di

accesso che autorizza la condivisione di informazioni specifiche sull'account. Il processo per ottenere il token è chiamato flusso di autorizzazione.

OAuth 1.0 è stato rilasciato per la prima volta nel 2007 come metodo di autorizzazione per l'API (Application Program Interface) di Twitter. Nel 2010, l'IETF OAuth Working Group ha pubblicato la prima bozza del protocollo OAuth 2.0. Come l'originale OAuth, OAuth 2.0 offre agli utenti la possibilità di concedere l'accesso di applicazioni di terze parti alle risorse Web senza condividere una password. Tuttavia, è un protocollo completamente nuovo e non è retrocompatibile con OAuth 1.0. Le funzionalità aggiornate includono un nuovo flusso di codice di autorizzazione per ospitare applicazioni mobili, firme semplificate e token di breve durata con autorizzazioni di lunga durata.
L'obiettivo di OATH è quello di realizzare un sistema di autenticazione aperto e senza diritti d'autore utilizzando, dove possibile, standard già esistenti, attraverso una serie di componenti hardware alla portata di tutti gli utenti. Gli scenari d'utilizzo di un sistema OATH sono molto vasti e spaziano dai siti online delle banche, a dispositivi desktop, fino ad arrivare alle VPN. Inoltre, il sistema permette la coesione di framework diversi, impedendo in questo modo di essere vincolati ad un solo venditore.
In Figura 12 è rappresentata l'architettura di OAuth.

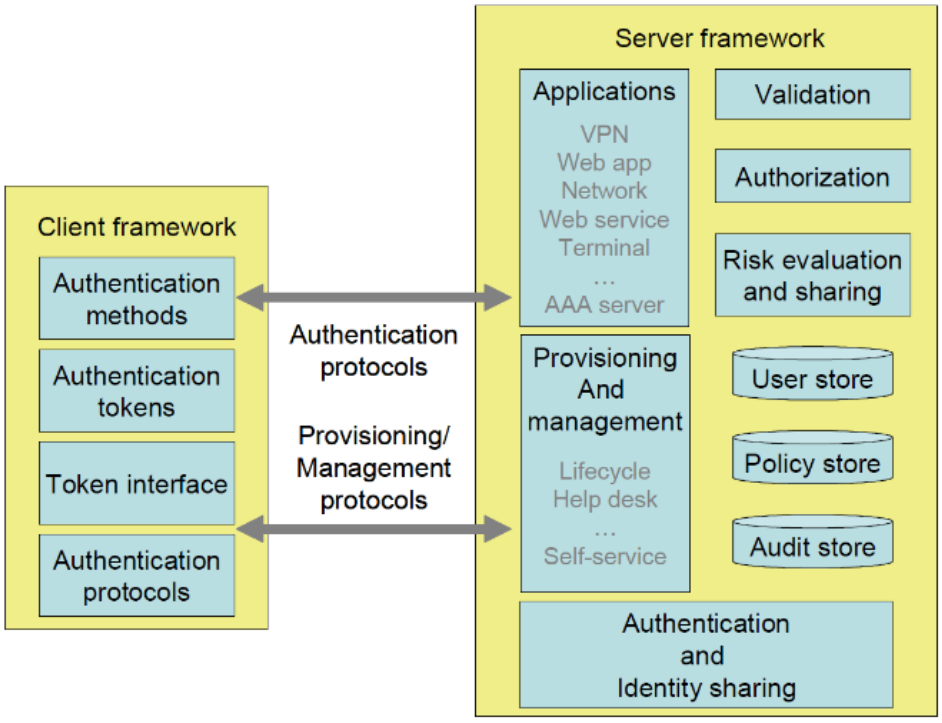

Figura 12 Modello di autenticazione tramite OpenID

1. Il client framework rappresenta l'insieme di tutti i metodi, i tokens, le interfacce ed i protocolli implementabili su un'architettura OATH.
2. Il provisioning e management framework, si occupa di gestire il ciclo di vita del software e delle credenziali di sicurezza per un sistema di autenticazione.
3. il validation framework, permette di verificare la validità delle credenziali in base al tipo di sistema implementato,
1. l'applications, fa da intermediario tra il client che vuole autenticarsi e il protocollo che viene implementato,
2. l'authorization, decide l'insieme delle risorse a cui il client può accedere dopo essersi autenticato,
3. l'user store, è un DB all'interno del quale sono contenute le informazioni inerenti al client,
4. il policy store, è un unico DB all'interno del quale sono memorizzate le politiche di tutti i componenti dell'architettura (esiste anche un'architettura in cui viene definita una policy store dedicata per i singoli componenti),
5. l'audit store, è un DB che contiene un repository con i controlli per i vari componenti (utilizzando strutture simili a quelle del policy store),
6. l'authentication and identity sharing `e la componente che implementa le tecnologie e i modelli necessari ad abilitare la condivisione del sistema di autenticazione,

7. risk evaluation and sharing, permette di valutare il rischio associato ad una particolare transazione.

OpenID

OpenID è una specifica open source per l'autenticazione single sign-on (SSO). Rilasciato per la prima volta nel 2005, consente ai siti web e ai servizi di autenticazione di scambiare informazioni sulla sicurezza in modo standardizzato. Nel febbraio 2014, la OpenID Foundation ha lanciato una nuova versione del protocollo chiamata OpenID Connect. OpenID Connect si basa sul framework di autenticazione OAuth 2.0 per migliorare la gestione delle identità, l'interoperabilità e il supporto allo sviluppo di applicazioni mobili.
L'obiettivo di OpenID Connect è quello di consentire all'utente finale di effettuare il login una sola volta e di accedere a più risorse diverse sul Web e fuori dal Web. La peculiarità dei meccanismi chiamati “Single Sign-On”, è quella di poter effettuare il login su diversi siti utilizzando solo un “User Identifier”, definito dal sistema in fase di registrazione. Tutta l’autenticazione viene quindi spostata su un provider OpenID, richiedendo quindi all’utente di autenticarsi una sola volta per sessione. Questo permette di rendere il processo di autenticazione più sicuro, in quanto non si dovrà più andare a contattare più server gestiti da diverse entità. Questa scelta implementativa è un esempio di come un miglioramento relativo alla sicurezza provochi anche un miglioramento sull’usabilità dell’utente finale, al quale basterà ricordarsi un’unica combinazione di ID e password, per potersi autenticare su siti differenti. OpenID può anche implementare meccanismi di autenticazione aggiuntivi come OTP o certificati SSL.
L’architettura su cui si basa OpenID è formata da (figura 13):

A. User Agent, il browser web dal quale l’utente può utilizzare l’OpenID identifier per autenticarsi al sito (relying party), ottenuto dal fornitore OpenID di fiducia.
B. OpenID Provider, un server che fornisce i vari OpenID dentifier degli utenti definiti in fase di registrazione. Inoltre, convalida le credenziali dell'utente per conto della relying party.
C. Relying party, il sito web a cui l’utente vuole collegarsi. Questo identificativo sarà convalidato dal provider OpenID dell'utente.

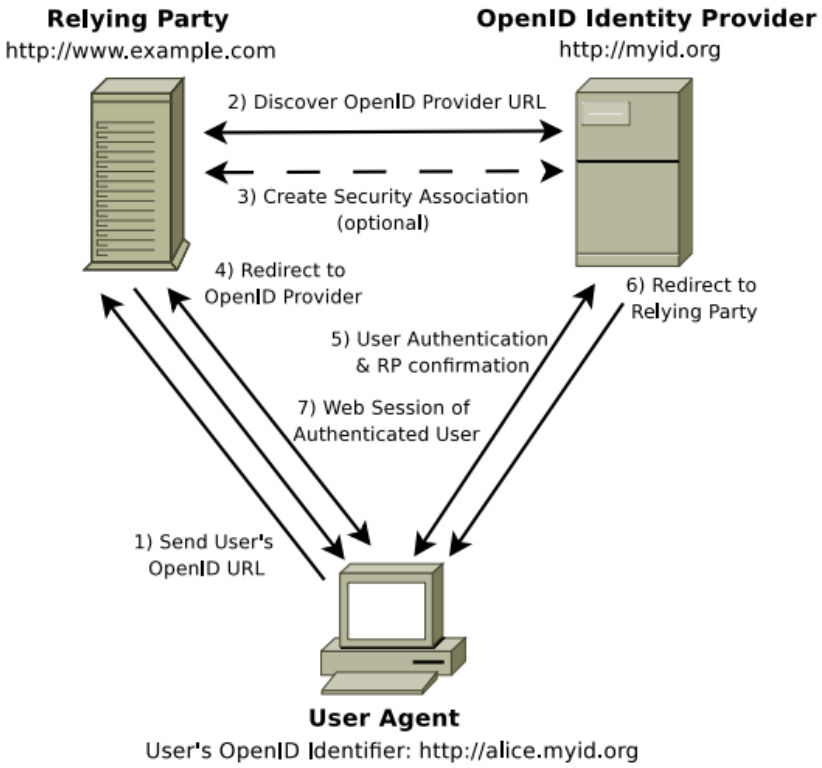

Figura 13 Architettura OpenID

Kerberos

Il servizio di autenticazione Kerberos, sviluppato presso il MIT (Massachusetts Institute of Technology), fornisce un'autenticazione di terze parti affidabile per verificare l'identità degli utenti.
Kerberos si basa sul protocollo di Needham-Schroeder. Utilizza una terza parte affidabile per centralizzare la distribuzione delle chiavi detta Key Distribution Center (KDC) utilizzando la crittografia a chiave simmetrica; 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.

Kerberos fornisce un mezzo per verificare le identità degli utenti su una rete aperta (non protetta). Ciò si ottiene senza fare affidamento sull'autenticazione da parte del sistema operativo host, senza basare la fiducia sull'indirizzo host, senza richiedere la sicurezza fisica di tutti gli host sulla rete e presumendo che i pacchetti viaggino lungo la rete possono essere letti, modificati e inseriti a piacimento.
Kerberos esegue l'autenticazione in queste condizioni come servizio di autenticazione di terze parti fidate, utilizzando la crittografia convenzionale. È fidato nel senso che ciascuno dei suoi client ritiene che il giudizio di Kerberos sull'identità di ciascuno degli altri client sia accurato.
Kerberos esegue l'autenticazione in queste condizioni come servizio di autenticazione di terze parti fidate, utilizzando la crittografia convenzionale. È affidabile nel senso che ognuno dei suoi client crede che il giudizio di Kerberos sull'identità di ogni altro client sia accurato.

Il problema che Kerberos affronta è il seguente: un sistema distribuito in cui gli utenti delle postazioni di lavoro desiderano accedere ai servizi dei server distribuiti sulla rete. Vorremmo che i server fossero in grado di limitare l'accesso agli utenti autorizzati e che siano in grado di autenticare le richieste di servizio. In questo sistema esistono le seguenti tre minacce:

1. un utente può accedere a una particolare postazione di lavoro e fingere di essere un altro utente che opera da quella postazione di lavoro,
2. un utente può alterare l'indirizzo di rete di una stazione di lavoro in modo che le richieste inviate dalla stazione di lavoro alterata sembrino provenire dalla stazione di lavoro impersonata,
3. un utente può “origliare” gli scambi e utilizzare un replay attack per entrare in un server o per interrompere le operazioni.

In ognuno di questi casi, un utente non autorizzato può accedere a servizi e dati a cui non è autorizzato ad accedere. Kerberos fornisce un server di autenticazione centralizzato la cui funzione è quella di autenticare gli utenti ai server e i server agli utenti.

Kerberos prevede i seguenti requisiti:

1. Sicurezza: un intercettatore di rete non deve essere in grado di ottenere le informazioni necessarie per impersonare un utente.
2. Affidabile: Kerberos deve essere altamente affidabile e deve impiegare un'architettura di server distribuita, con un sistema in grado di un sistema in grado di eseguire il backup di un altro.
3. Trasparente: l'utente non deve essere a conoscenza del fatto che l'autenticazione sta avvenendo, al di là dell'obbligo di inserire una password.
4. Scalabile: il sistema deve essere in grado di supportare un gran numero di client e server.

FIDO: WebAuthn

WebAuthn è uno standard Web pubblicato dal World Wide Web Consortium (W3C). WebAuthn è un componente fondamentale del progetto FIDO2 sotto la guida di FIDO Alliance. L'obiettivo del progetto è standardizzare un'interfaccia per l'autenticazione degli utenti ad applicazioni e servizi basati sul Web utilizzando la crittografia a chiave pubblica. L'agenzia FIDO (Fast Identity Online) si pose l'obiettivo di creare una serie di protocolli standard aperti di autenticazione forte per eliminare le password, cambiare la situazione in cui la maggior parte delle attuali strategie di autenticazione utilizza ancora le password come credenziali ed eliminare o ridurre la dipendenza degli utenti dalle password. Inizialmente, hanno proposto il protocollo FIDO *UAF (Universal Authentication Framework Protocol*) per consentire l'offerta locale di metodi di autenticazione diversi dalle password sui dispositivi utilizzati, ad esempio uno scanner di impronte digitali. Accanto a FIDO UAF, hanno introdotto l'*Universal 2nd Factor Protocol* (FIDO *U2F)* per consentire ai servizi di offrire un secondo fattore per aumentare la sicurezza degli accessi basati su password. Nel 2019, successore di FIDO *FIDO2* è stato completato e migliorato sui diversi aspetti della versione FIDO originale per consentire più casi d'uso come l'interazione tra dispositivi esterni come autenticatori e il browser web. Di conseguenza è costituito da *CTAP2* (Client to Authenticator Protocol,

versione 2) e *WebAuthn (Web Authentication API)* sviluppati principalmente dal *World Wide Web Consortium (W3C)*.

Nel complesso, ci sono tre attori chiave che prendono parte a qualsiasi flusso FIDO che sia per la *registrazione* o *l'autenticazione*:

- Il FIDO *Relying Party*, che memorizza i dati e le credenziali pubbliche degli utenti.
- *L' Autenticatore* FIDO, che può confermare la presenza di un utente e in alcuni casi anche verificarne l'identità prima di firmare e quindi autenticare una contestazione inviata dal *Relying Party*.
- *Il client* FIDO, che funge da intermediario tra *Relying Party* e *Authenticator* , visualizzando anche un'interfaccia utente grafica per offrire opzioni che l'utente può configurare.

In FIDO2, la specifica WebAuthn gestisce la comunicazione tra la relying *party* e *il client*, essendo quindi indipendente dai processi di *autenticazione* e rimanendo compatibile con la precedente specifica FIDO *U2F*.

Il *protocollo Client to Authenticator* descrive come un *autenticatore* comunica con un *client* sul dispositivo dell'utente. Si compone di tre parti, ovvero l'API *Authenticator, le codifiche dei messaggi* e le *associazioni specifiche per il trasporto:*

- **API Authenticator:** l'API *Authenticator* definisce diverse operazioni sull'autenticatore che possono essere eseguite indipendentemente l'una dall'altra. Questi includono l'ottenimento di informazioni sulle capacità dell'autenticatore *(authenticatorGetInfo)* , le operazioni necessarie per la registrazione *(authenticatorMake Credential)* e le procedure di autenticazione *(authenticatorGetAssertion)* , ad esempio. A questo livello di astrazione, il trasporto sottostante non è rilevante.
- **Codifica dei messaggi**: la codifica dei messaggi definisce le regole per la codifica delle richieste e delle risposte dall'API Authenticator. Poiché alcuni trasporti sono limitati dalla larghezza di banda e gli autenticatori potrebbero avere capacità di elaborazione molto limitate o poca memoria disponibile, si basa sulla rappresentazione concisa di oggetti binari (CBOR) .
- **Collegamenti specifici per il trasporto**: questa parte è suddivisa in tre sottosezioni, che descrivono i collegamenti dei messaggi specifici per ciascuna tecnologia di trasporto. I tre trasporti attualmente supportati sono USB, NFC e BLE.

IDENTITY PROVIDER (IDP)

Un'identità digitale consiste nella rappresentazione digitale di un insieme di informazioni note per uno specifico individuo o organizzazione, ad esempio, e-mail e password sono informazioni che vengono utilizzate per identificare utenti di determinati servizi online. L'identità di una persona comprende un'enorme quantità di dati personali. Tutti i sottoinsiemi dell'identità rappresentano la persona o alcuni componenti della persona Figura 14 (identità parziali). Alcune di queste componenti identificano in modo univoco la persona, altre no. A seconda della situazione e del contesto, la persona può essere rappresentata da diverse identità parziali.

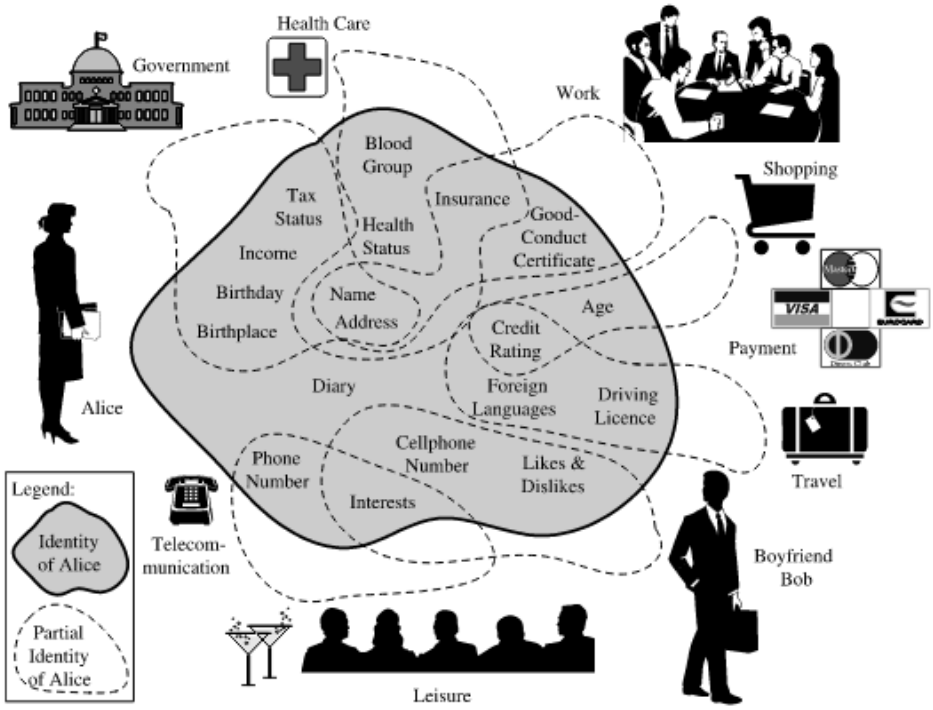

Figura 14 identità e identità parziali [14]

Un sistema di gestione dell'identità (identity management system) fornisce gli strumenti per gestire queste identità parziali nel mondo digitale. Ad esempio, una persona può utilizzare una o più identità parziali per lavoro, per attività ricreative, o per interagire con aziende come una banca o come un qualsiasi provider di servizi online. Pertanto, la gestione dell'identità nel mondo digitale è strettamente relata al comportamento delle persone nelle attività quotidiane. Ciascuno decide cosa dire all'altro di sé, dopo aver

considerato il contesto e il ruolo degli attori coinvolti nell'interazione. A volte soprannomi o pseudonimi possono essere legati all'identità, In alcuni casi si potrebbe rimanere anonimi, ad esempio se si vuole acquistare bevande alcoliche è necessario dimostrare di avere la maggiore età ma non è necessario rilasciare altre informazioni quali nome, cognome, nazionalità, luogo di nascita ecc. In altri casi è necessario rivelare dati personali identificativi, ad esempio quando viene richiesto da un rappresentante governativo di esibire la carta d'identità. Spesso l'anonimato non è accettabile, ma sono necessari solo alcuni dati personali o credenziali. Le scelte dell'utente di esporre solo determinate informazioni ad applicazioni digitali sono supportate dagli identity management system. Le organizzazioni che attraverso gli identity management system controllano e gestiscono le identità digitali sono chiamate **identity provider (IdP)** e possono essere pubbliche (ad esempio lo Stato) oppure private (ad esempio aziende, Social ecc). Gli Identity management sono una componente chiave per queste organizzazioni che devono gestire gli account dei propri clienti (identità) e proteggere sia gli accessi alle risorse, sia i servizi e le informazioni personali dei clienti stessi (attributi). L'IdP può autenticare direttamente l'utente o fornire servizi di autenticazione a provider di servizi di terze parti (app, siti web o altri servizi digitali) a seconda del modello di gestione dell'identità. Qualsiasi sito web che richiede un accesso, per esempio, utilizza un IdP per autenticare gli utenti. Dal punto di vista dell'IdP, un utente è noto come richiedente. Il richiedente può essere un umano o una macchina. Un IdP può autenticare qualsiasi entità, inclusi i dispositivi. Lo scopo di un IdP è tracciare queste entità e sapere dove e come recuperare le identità principali che determinano se una persona o un dispositivo può accedere ai dati sensibili.

Un **provider di servizi o service provider (SP)** è l'entità che fornisce il servizio a cui accedere, mentre un IdP è l'entità che crea, archivia e gestisce le identità, inoltre ha la capacità di autenticare un utente. In determinati modelli di gestione dell'identità queste entità coincidono, infatti inizialmente, le organizzazioni hanno affrontato la gestione dell'identità da una prospettiva puramente interna. La loro preoccupazione principale era (ed è tuttora per alcuni di loro) fornire ai propri clienti un accesso sicuro e senza interruzioni ai propri servizi.

Abbiamo tre modelli principali di modelli di gestione dell'identità digitale:

1. modello centralizzato,
2. modello federato,
3. modello decentralizzato user-centrico.

Modello Centralizzato

In questo modello, noto anche come **Modello a Silos**, le informazioni personali e l'autenticazione degli utenti sono limitate ai confini dell'organizzazione, l'identità è detenuta in modo centralizzato dalle organizzazioni citate in precedenza che in questo specifico caso coincidono con i provider del servizio stesso; dunque, identity provider e provider del servizio sono la stessa entità. Fanno parte di questo modello tutti i servizi dove è necessario creare un account presso l'organizzazione per poter accedere al servizio gestito dall'organizzazione stessa (ad esempio Facebook). In altre parole, le organizzazioni permettono agli utenti di accedere a determinati servizi richiedendo dei *segreti* che solo l'utente può conoscere (Password, Pin ecc.). Tutti i dati personali dell'utente sono conservati all'interno di database gestiti dall'organizzazione, questi database sono detti anche *Silos* da cui il nome Modello a Silos. Tutto ciò porta a una situazione in cui gli utenti hanno diverse identificazioni indipendenti su Internet che impediscono loro di beneficiare di registrazioni e autenticazioni già eseguite presso altre organizzazioni. Nascono quindi problematiche relative all'esperienza utente poiché ogni utente deve creare diverse password per ogni servizio al quale vuole accedere. Consentire alle organizzazioni di condividere le autenticazioni e gli attributi degli utenti è fondamentale sia per gli utenti per beneficiare di un accesso continuo ai servizi di varie organizzazioni, sia per le organizzazioni per realizzare nuove opportunità di business. Tuttavia, la condivisione di autenticazioni e attributi tra le organizzazioni solleva una serie di problemi chiave (inclusi problemi tecnici, di interoperabilità e privacy). Pertanto, alcune organizzazioni hanno compiuto molti sforzi per implementare sistemi Single Sign-On (SSO) consentendo ai propri servizi di condividere le autenticazioni e gli attributi degli utenti e impedendo, in questo modo, ai propri clienti di effettuare più registrazioni e

autenticazioni durante l'accesso ai propri servizi. Tuttavia, la maggior parte delle tecnologie e dei modelli di gestione delle identità esistenti che le organizzazioni hanno utilizzato e implementato internamente sono inappropriati in un contesto cross-domain. Non sono solamente necessari degli standard per consentire l'interoperabilità di sistemi di gestione delle identità, ma serve qualcosa che è più della semplice standardizzazione di tecnologie e modelli esistenti. Sono necessari nuovi modelli e protocolli standard di gestione dell'identità. Altre problematiche nascono da rischi di sicurezza dovuti a possibili *data leak*.

Pseudo SSO

I sistemi pseudo-SSO implicano l'uso di un componente SSO che raccoglie e memorizza le credenziali degli utenti (usemame e password, chiavi private, certificati ecc.), e le utilizza per ottenere l'autenticazione degli utenti. All'inizio di una sessione, l'utente deve solo accedere una volta al componente SSO. Quando l'utente accede a un SP, il componente SSO utilizza le credenziali appropriate per autenticarsi automaticamente, esattamente come farebbe normalmente l'utente (ad es. compilando il nome utente e la password nel modulo di autenticazione che il SP presenta all'utente). Nella figura 15 viene schematizzato l'architettura descritta.

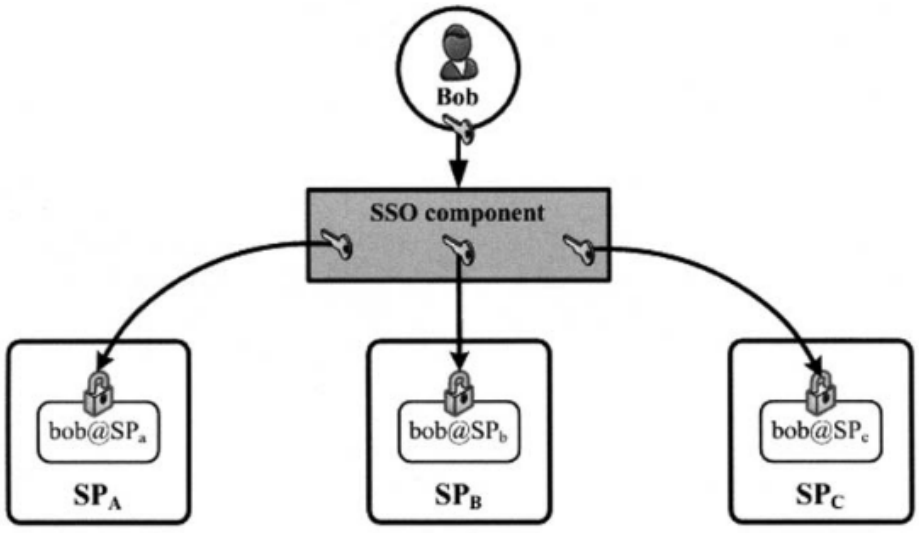

Figura 15: Approccio Pseudo SSO

Le soluzioni pseudo-SSO rientrano in due categorie a seconda della posizione del componente SSO:

1. pseudo-SSO locale, in cui il componente SSO si trova all'interno della macchina dell'utente. Le credenziali degli utenti possono essere archiviate localmente sul disco rigido,
2. pseudo-SSO remoto, in cui l'utente ha la possibilità di salvare le credenziali su database remoti consentendo agli utenti di beneficiare delle funzionalità SSO da qualsiasi luogo. In un'architettura pseudo-SSO basata su proxy, il componente SSO risiede su un server proxy. In questo caso, la funzionalità SSO è offerta come servizio dall'organizzazione che ospita il proxy.

Il componente SSO è un componente centrale delle architetture pseudo-SSO ed è di particolare importanza per la privacy.
Questo componente esegue le autenticazioni per conto degli utenti e necessita dell'accesso alle credenziali in chiaro. Nelle architetture pseudo-SSO basate su proxy, una singola organizzazione ospita il componente SSO e può potenzialmente ottenere le credenziali degli utenti, utilizzarle per impersonare gli utenti all'interno dei diversi SP, ottenere l'accesso alle informazioni personali degli utenti e correlare le loro identità parziali.

Microsoft .NET Passport

Uno dei primi esempi di un modello centralizzato, che realizza la Single Sign-On, Per la gestione di identità digitali è **Microsoft .NET Passport** che fu sviluppato alla fine degli anni 90 e commercializzato nei primi anni del 2000. Passport fu un ambizioso servizio di identificazione e autenticazione online. Gli utenti hanno bisogno di un solo nome utente di accesso (un indirizzo e-mail o un numero di telefono) e una password archiviata in un luogo univoco per tutte i SP. Oltre al nome utente e alla password, gli utenti possono opzionalmente memorizzare informazioni personali (nome, cognome, sesso, professione, data di nascita, indirizzo, ecc.) in modo tale che Passport disponga di un

server centralizzato che contenga tutte le informazioni dell'utente. A seconda delle loro scelte, gli utenti possono condividere parte di queste informazioni con i SP. Inoltre, gli utenti Passport possono anche archiviare carte di credito e indirizzi di spedizione nel proprio portafoglio Passport ed effettuare acquisti online utilizzando il servizio di acquisto rapido .Net Passport.

Durante la creazione dell'account online Passport, all'identità dell'utente viene assegnato un PUID ovvero Passport Unique Identifier. Il PUID è un valore numerico a 64 bit che è lo stesso per tutti i SP. Questo PUID è usato nella comunicazione di token di sicurezza e degli attributi dell'utente ai SP. Nella figura 16 viene schematizzato l'architettura descritta.

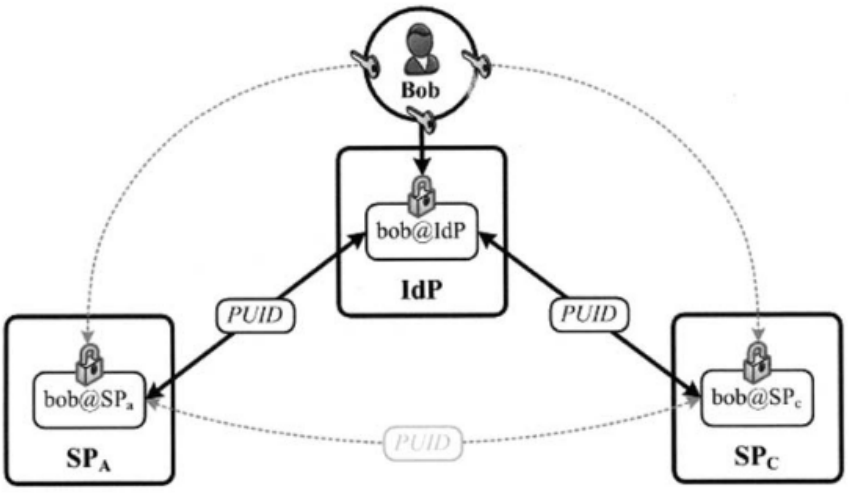

Figura 16: Approccio globale Centralizzato

Modello Federato

Per ovviare alle problematiche del Modello Centralizzato nasce il modello federato in cui l'identity provider fa da intermediario tra l'utente e il servizio a cui l'utente sta accedendo.L'IdP gestisce le identità e i profili dei suoi utenti locali, mentre gli SP gestiscono in modo autonomo i servizi di cui l'utente usufruisce. L'IdP, dunque, è l'effettivo possessore dei dati associati all'utente.

La gestione delle identità federate (Federated identity management - FIM) è una tecnologia efficace per condividere risorse tra più organizzazioni. Consente agli utenti di un'organizzazione di accedere ai servizi di altre organizzazioni federate in modo sicuro e senza interruzioni. Un esempio può essere **SPID**, l'utente attraverso SPID è in grado di accedere a diversi servizi, altro esempio di gestione federata dell'identità digitale è rappresentato da Google: un utente Google oggi può accedere in maniera federata a diversi servizi attraverso il suo account Google.
Questo modello di gestione dell'identità digitale permette di risolvere alcune problematiche legate all'esperienza utente; infatti, tramite una sola identità è possibile godere di un'esperienza di Single Sign On. Quando un utente accede a un SP, il service provider può scambiare automaticamente le informazioni di autenticazione con l'IdP dell'utente per identificarlo.

Ci sono diverse soluzioni FIM esistenti come:

- Security Assertion Mark Language (SAML) che di fatto è lo standard FIM,
- Liberty Identity Federation (ID-FF),
- WS-Federation.

Modello Self-Sovereign Identity (SSI)

Gli esistenti modelli di gestione dell'identità digitale che si basano su repository di dati centralizzati e identity prvider hanno portato a un numero crescente di violazioni dei dati, causando una perdita significativa di dati personali e un costo enorme per tutte le parti interessate e in particolare gli utenti. Un nuovo e rivoluzionario modello di gestione dell'identità digitale è rappresentato dal paradigma della **Self-Sovereign Identity (SSI)**. Il concetto di SSI è completamente *user-centrico*, ovvero, in contrasto con i modelli precedenti dove il SP era al centro del modello, l'utente è l'unico possessore della propria identità digitale e di tutti i dati ad essa associati. Tutto questo è possibile grazie all'utilizzo di tecnologie come la **Blockchain** che consente la realizzazione di *sistemi decentralizzati*. Inoltre, grazie alle caratteristiche di una Blockchain, quali immutabilità il

modello Self-Sovereign Identity vanta una maggiore sicurezza verso tutti gli attori coinvolti.

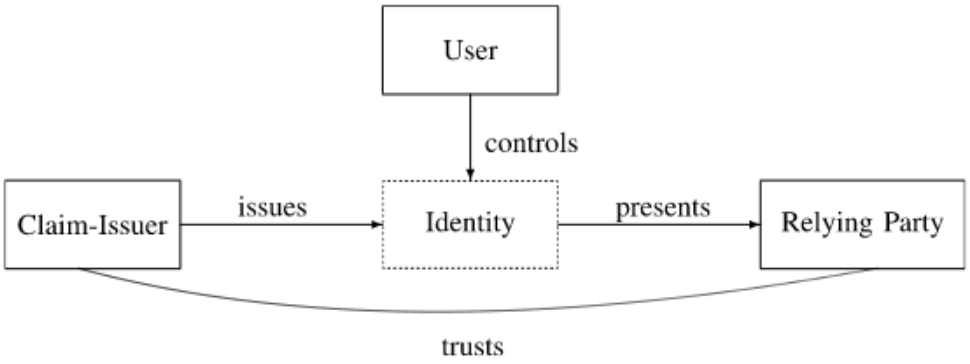

Figura 17: Self-Sovereign Identity Actors. [15]

In Figura 17 si osserva la relazione tra i diversi attori del sistema. L'autorità di certificazione rilascia l'identità attestando determinati attributi dell'utente. Questa identità è controllata dall'utente stesso. A tutte le parti che hanno bisogno di identificare l'utente (ad esempio servizi web) verranno presentate i claim dell'identità gestita dall'utente. Per accettare l'identità, il verificatore deve avere un rapporto di fiducia con l'emittente del claim (claim issuer). La base di questo nuovo tipo di architettura è il distributed ledger della blockchain.

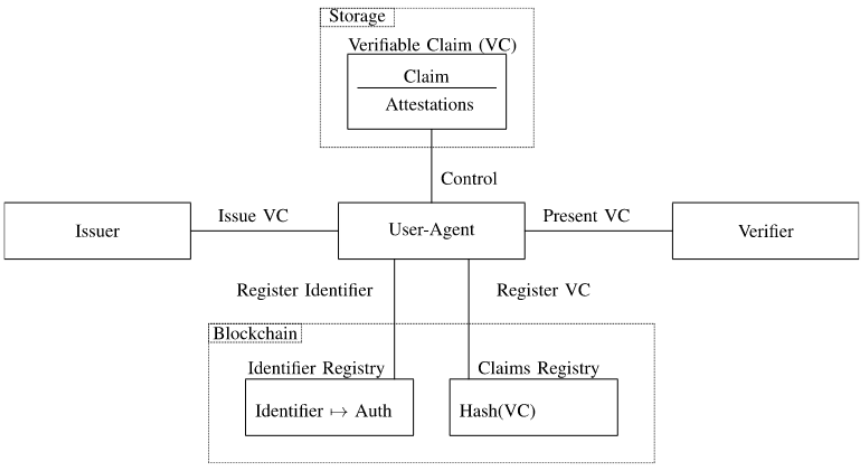

Figura 18: Self-Sovereign Identity Architecture [16]

In Figura 18 sono illustrate le relazioni tra le diverse componenti di una tipica architettura SSI. La blockchain funge da sostituto dell'autorità di registrazione nei classici sistemi di

gestione delle identità. L'identificatore e le credenziali verificabili sono gestiti direttamente dall'utente. L'identificatore è legato all'utente specifico mediante l'uso di un metodo di autenticazione come la crittografia asimmetrica. Stabilendo un abbinamento di identificatore e chiave pubblica sulla blockchain, l'identificatore può essere verificato da chiunque legga la blockchain ponendo una challenge all'utente stesso o a un delegato dell'utente. L'effettiva attestazione di identità viene conservata in un archivio controllato dall'utente, in genere fuori catena per motivi di privacy. La parte che fa affidamento, chiamata anche claim-verifier, può quindi confrontare l'identificatore pubblicamente disponibile con l'identificatore passato nell'attestazione dall'utente. Dopo aver autenticato l'utente con il metodo di autenticazione presentato nella blockchain pubblica, il claim stesso può essere verificato e accettato o rifiutato dal verificatore (relaying party). Un modello concorrente molto popolare può essere descritto come Claim Registry Model. In quel modello la blockchain non funziona solo come registro per gli identificatori di un'identità, ma anche per contenere le impronte crittografiche di tutte le attività associate di un'identità. Questo modello può essere visto come un'estensione dell'Identifier Registry Model. In questo processo nessuna informazione sull'utente deve essere memorizzata né presso l'emittente né il verificatore. Solo la fiducia tra l'emittente e il verificatore deve essere stabilita in anticipo. L'architettura SSI si basa sulla mappatura di un identificatore su uno specifico metodo di autenticazione registrato sulla blockchain.

Decentralized Identifiers DID

Sviluppati dal World Wide Web Consortium (W3C), i **Decentralized Identifiers (DID)** sono una componente chiave del modello SSI. La specifica DID definisce uno schema di identificatore crittografico univoco globale simile allo schema di identificatore univoco universale (UUID) con una serie di differenze. In primo luogo, la specifica DID non si basa su un'autorità centralizzata per gestire gli identificatori (per questo chiamati Decentralized Identifiers), ma piuttosto possono essere gestiti utilizzando un'infrastruttura decentralizzata come Distributed Ledger (DLT). In secondo luogo, gli indirizzi DID hanno proprietà crittografiche. gli indirizzi DID sono generati in base a coppie di chiavi crittografiche. la proprietà di un DID può essere dimostrata utilizzando prove crittografiche

come le firme digitali. Un indirizzo DID è composto da tre parti. Ogni DID ha il formato mostrato in Fig.13

Figura 19: Formato DID

la prima parte di ogni DID è lo schema DID, seguito dal metodo DID. la terza parte consiste in un identificatore nel contesto di un metodo DID. Ogni indirizzo DID si risolve in un Descriptor Object DID o DDO, noto anche come DID Document. DDO è un documento JSON-LD machine-readable con varie informazioni riguardo il DID subject.
Questi includono chiavi pubbliche crittografiche, service endpoint, parametri di autenticazione, timestamp e altri metadati. I service endpoint forniscono un indirizzo pubblicamente disponibile per stabilire la connessione con il DID subject.
Un DID subject è un'entità (persona, gruppo, organizzazione, ecc.) identificata da un DID e descritta da un DDO. Un record DID è costituito da una coppia chiave-valore di un DID e un DDO. La Fig.14 illustra un documento DDO di esempio. Un'entità può avere più coppie di chiavi e DID e può utilizzare un DID diverso per ogni interazione.

Figura 20: Esempio di decentralized identifier e DID document

il metodo DID definisce i metodi specifici che uno schema DID con cui può essere implementato su un particolare DLT o reti. ciò include le operazioni CRUD di creazione, lettura, aggiornamento ed eliminazione dei record DID. Ad oggi, ci sono 90 metodi DID registrati. questi metodi includono Bitcoin, Ethereum, Sovrin, Interplanetary File System (IPFS) e Veres Ones. La specifica DID distingue tre tipi di DID.

Anywise DID o *Public DID* è un DID destinato all'uso con un numero sconosciuto di parti. Il *Pairwise DID* è destinato alle interazioni in cui il DID dovrebbe essere conosciuto solo dal suo subject e esattamente da un'altra entità come un fornitore di servizi (SP). Infine, il DID *N-wise* deve essere conosciuto esattamente da un numero N di entità compreso il suo subject. Un *universal resolver* è un sistema di risoluzione DID che supporta più sistemi di identificazione decentralizzati. Affinché un sistema DID supporti il resolver universale, deve implementare un adattatore DID per interfacciare il resolver universale con i metodi DID specifici del sistema.

Il protocollo *DID Auth* consente a un proprietario di identità di utilizzare la propria applicazione client, come il proprio dispositivo mobile o browser, per dimostrare a un

fornitore di servizi di avere il controllo di un DID. Il protocollo DID Auth si basa su un ciclo challenge-response personalizzato a seconda della situazione. (Questo protocollo può sostituire l'uso di nome utente e password come forma di autenticazione e consente un canale di comunicazione autenticato tra un proprietario di identità e un fornitore di servizi.

DID Comm è un protocollo basato su DID mediante il quale due o più entità SSI possono comunicare in modo privato e sicuro in modo peer-to-peer. Il protocollo si basa su DID e supporta l'autenticazione reciproca tra le parti.

Verificable Credential VC

Le credenziali verificabili (VC) sono una specifica sviluppata dal W3C Verifiable Credentials Working Group. Una credenziale verificabile può rappresentare tutte le stesse informazioni rappresentate da una credenziale fisica. L'aggiunta di tecnologie, come le firme digitali, rende le credenziali verificabili più affidabili rispetto alle loro controparti fisiche. La Fig.15 mostra i ruoli chiave all'interno dell'ecosistema delle credenziali verificabili.

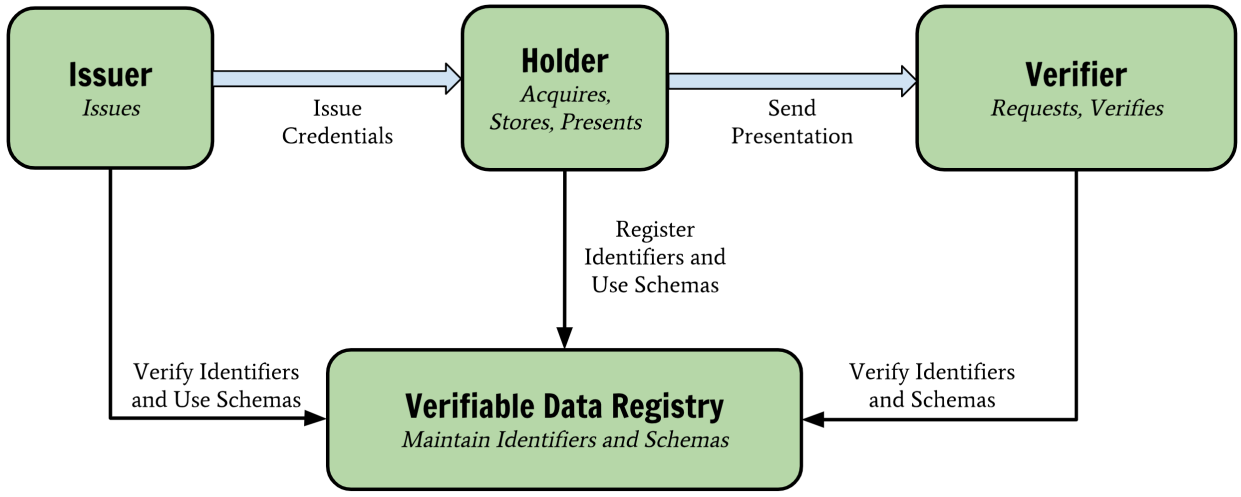

Figura 21: Attori coinvolti nelle Verificable Credential [15]

Un *Holder* è un'entità che controlla una o più credenziali verificabili. Un *Issuer* è un'entità che crea nuove credenziali verificabili. Una banca e un'agenzia governativa sono esempi di emittenti di credenziali. Un *Verifier* è un'entità che ottiene un'altra credenziale

verificabile da verificare. Un sito di e-commerce che si aspetta credenziali dai propri clienti è un esempio di verificatore. Un *verifiable data registry* è responsabile della mediazione della creazione e verifica di identificatori, chiavi, schemi di credenziali verificabili e altri dati rilevanti necessari per utilizzare credenziali verificabili.
Una credenziale verificabile è composta da vari elementi. (includono l'URI del soggetto, l'URI dell'emittente delle attestazioni e gli URI che identificano in modo univoco la credenziale. Inoltre, un VC include le condizioni di scadenza delle attestazioni e la firma crittografica.
Un URI può essere un DID. il W3C Verifiable Credentials Working Group ha anche definito il concetto di *Verifiable Presentations (VP)*. Le presentazioni verificabili definiscono le modalità con cui le VC sono firmate e presentate dal titolare. Un VC o un VP può essere descritto con JSON-LD, JSON o JSON Web Token.

I titolari di credenziali verificabili possono generare presentazioni verificabili e quindi condividere queste presentazioni verificabili con i verificatori per dimostrare di possedere credenziali verificabili con determinate caratteristiche. Sia le credenziali verificabili che le presentazioni verificabili possono essere trasmesse rapidamente, rendendole più convenienti rispetto alle loro controparti fisiche quando si cerca di stabilire la fiducia a distanza.
La parola "verificabile" nei termini credenziali verificabili e presentazione verificabile si riferisce alla caratteristica di una credenziale o presentazione di poter essere verificata da un verificatore. La verificabilità di una credenziale non implica che la verità delle affermazioni ivi codificate possa essere valutata; tuttavia, l'emittente può includere valori nella proprietà dell'evidenza per aiutare il verificatore ad applicare la propria logica aziendale per determinare se le affermazioni hanno una veridicità sufficiente per le proprie esigenze.
Un *Claim* è una dichiarazione su un soggetto. Un soggetto è una cosa su cui si possono fare affermazioni. Le affermazioni sono espresse utilizzando relazioni soggetto-proprietà-valore.
È importante riconoscere che esiste uno spettro di privacy che va da pseudonimo a fortemente identificato. A seconda dell'uso, le persone hanno diversi livelli di comfort su

quali informazioni sono disposte a fornire e quali informazioni possono essere derivate da ciò che viene fornito.

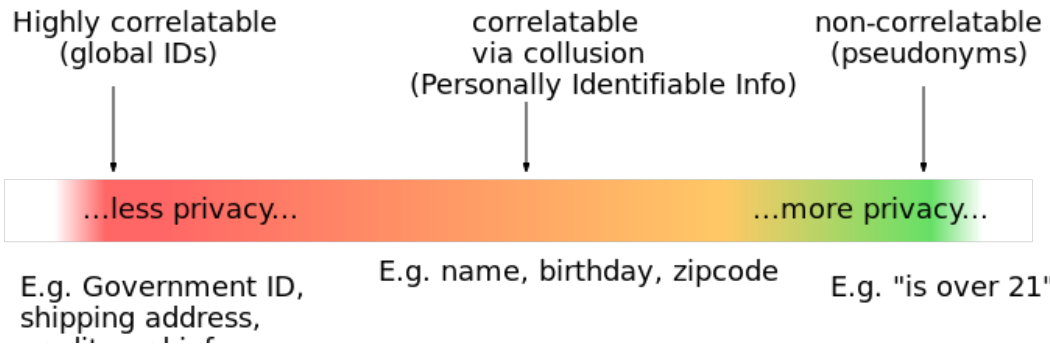

Figura 22: Spettro di privacy che va da pseudonimo a completamente identificato

Ad esempio, la maggior parte delle persone probabilmente desidera rimanere anonima quando acquista alcolici perché il controllo normativo richiesto si basa esclusivamente sul fatto che una persona abbia un'età superiore a un'età specifica. In alternativa, per le prescrizioni mediche scritte da un medico per un paziente, la farmacia che esegue la prescrizione è tenuta a identificare più chiaramente il medico e il paziente. Pertanto, non esiste un approccio alla privacy che funzioni per tutti i casi d'uso. Le soluzioni per la privacy sono specifiche del caso d'uso.

Il Modello dei dati delle credenziali verificabili si sforza di supportare l'intero spettro della privacy e non assume posizioni filosofiche sul corretto livello di anonimato per qualsiasi transazione specifica.

Ci sono una serie di considerazioni sulla sicurezza di cui gli emittenti, i detentori e i verificatori dovrebbero essere a conoscenza durante l'elaborazione dei dati descritti da questa specifica. Ignorare o non comprendere le implicazioni di questa sezione può causare vulnerabilità della sicurezza. Sebbene questa sezione tenti di evidenziare un'ampia serie di considerazioni sulla sicurezza, non è un elenco completo. Gli implementatori sono invitati a chiedere consiglio ai professionisti della sicurezza e della crittografia quando implementano sistemi mission-critical utilizzando la tecnologia delineata in questa specifica.

Considerazioni su i JSON Web Token

I vantaggi dell'utilizzo di JWT sono:

1. riduzione delle richieste al Database: ciò implica un minor numero di query DB, il che implica tempi di risposta più rapidi,
2. utilizzato tra i servizi: è possibile avere un server di autorizzazione che si occupa dell'accesso/registrazione e genera il token, tutte le richieste successive non dovranno andare al server di autorizzazione poiché l'unico server di autenticazione avrà la chiave privata, e il resto dei server avrà la chiave pubblica per verificare la firma. Questo è utile nel caso di sistemi aziendali in cui il server di autorizzazione si trova in un ambiente sicuro e non esiste alcuna connessione tra il server di autenticazione e il resto dei server oltre alla chiave pubblica predefinita,
3. sicurezza: poiché il cookie non viene inviato, impedirebbe gli attacchi CSRF (Cross-site request forgery). Inoltre, non ci sta nessuna informazione basata sulla sessione da manipolare. E la revoca del token consente di convalidare il token in base al tempo di scadenza,
4. estensibilità: fornisci l'autorizzazione selettiva alle app di terze parti. Inoltre permette di creare un'API propria e permette di distribuire un token di autorizzazione speciale che consente all'applicazione di accedere ai dati dell'utente,
5. piattaforme multiple: man mano che le applicazioni si espandono per gestire diversi dispositivi mobili e domini, deve essere gestito il CORS.
   Finché l'utente dispone di un token valido, è possibile mantenere l'interoperabilità

I svantaggi relativi al JWT sono:

1. compromissione della chiave: Poiché JWT è si basa su una sola chiave. Se la chiave viene svelata da uno sviluppatore/amministratore disattento è necessario generare una nuova chiave (coppia di chiavi) che verrà utilizzata su tutti i sistemi in questo momento invalidando tutti i token client esistenti,
2. impossibile inviare messaggi ai client (identificazione dei client dal server): Poiché non abbiamo alcuna registrazione sui client collegati all'estremità del database, non possiamo inviare messaggi a tutti i client,

3. gli algoritmi di cifratura utilizzati possono essere deprecati: poiché JWT si basa completamente sull'algoritmo di firma. Anche se non sia frequente, in passato molti algoritmi di crittografia/firma sono stati deprecati poiché possono presentarsi debolezze nell'algoritmo,
4. dati generali: la dimensione del token JWT sarà maggiore di quella di un normale token Session. Più dati vengono aggiunti nel token JWT, più cresce in modo lineare. Ogni richiesta necessita del token al suo interno per la verifica della richiesta. Ad esempio, un token JWT da 1 KB implica che ogni richiesta avrà 1 KB di caricamento, il che è davvero negativo in caso di connettività di rete a bassa velocità,
5. complicato da capire: il JWT utilizza algoritmi di firma crittografica per verificare i dati e ottenere l'ID utente dal token. La comprensione dell'algoritmo di firma in sé richiede le basi della crittografia. Quindi, nel caso in cui lo sviluppatore non sia completamente istruito, potrebbe introdurre falle di sicurezza nel sistema.

Decentralized public key infrastructur DPKI

L'infrastruttura a chiave pubblica è costituita da un insieme di servizi, strumenti, processi e tecnologie che facilitano l'esecuzione di operazioni crittografiche basate sulla crittografia a chiave pubblica. Il modello di certificato PKI più comunemente utilizzato è noto come PKI X.509 o PKIX. In questo modello, le *central certificate authorities (CA)* creano il certificato digitale X.509. Un certificato che associa una chiave pubblica a una particolare identità.

Il modello di governance utilizzato dalle suddette autorità centrali rischia di porre il controllo dei dati di identità e il processo decisionale nelle mani di un ristretto insieme di autorità centrali, con la possibilità che tali autorità si comportino male o diventino vittime di violazioni della sicurezza.

Una diffusa e nota implementazione di un modello di fiducia decentralizzato, prima dell'emergere della blockchain, è il protocollo Pretty Good Privacy (PGP).

Proposto da Phil Zimmermann e contrariamente ai modelli PKI tradizionali, il modello di fiducia PGP si basa su una rete di fiducia. La popolarità e la crescita di PGP sono state ostacolate da vari problemi di usabilità e gestione delle chiavi, oltre ai problemi di sicurezza associati all'uso di chiavi a lungo termine.

Proposta dal Rebooting of the Web of Trust (RWoT), l'infrastruttura a chiave pubblica decentralizzata (DPKI) fornisce un'architettura di fiducia decentralizzata in cui nessuna singola entità può compromettere la sicurezza e l'integrità del sistema nel suo insieme. A differenza della PKI tradizionale, la DPKI non dipende dalle autorità di certificazione centrali o dalle autorità di registrazione (RA). La dipendenza da queste entità centrali può essere eliminata facendo affidamento su una piattaforma decentralizzata come radice iniziale della fiducia. L'architettura DPKI può essere realizzata con l'aiuto di un archivio dati chiave-valore decentralizzato come blockchain. Facendo affidamento sull'immutabilità della blockchain per memorizzare le chiavi pubbliche, è possibile garantire che i dati non possano essere cancellati o modificati da nessuno una volta scritti. Le chiavi crittografiche possono essere trovate da chiunque abbia accesso alla blockchain.
Questo risolve parte delle preoccupazioni relative a privacy e sicurezza associate un punto centrale dell'autorità. A causa della natura trasparente e aperta delle blockchain pubbliche, qualsiasi modifica ai dati è verificabile da tutti i membri della rete. Sebbene un verificatore possa fare affidamento su DPKI per consentire a una particolare entità di autenticarsi e rivendicare il possesso di un particolare identificatore e chiave pubblica, ciò non si traduce nel fatto che il verificatore si fida dell'identità dell'entità. L'identificazione di un'entità avviene attraverso lo scambio di credenziali verificabili. Sebbene le credenziali verificabili contengano attestazioni di identità firmate digitalmente, in definitiva spetta al verificatore fidarsi e accettare o rifiutare tali attestazioni in base alle proprie politiche e al proprio modello di fiducia.
Il progetto Sidetree proposto attraverso Decentralized Identity Foundation (DIF) è un protocollo emergente per la creazione di reti DPKI scalabili che possono funzionare su qualsiasi sistema di ancoraggio decentralizzato esistente come Bitcoin, pur essendo aperto e pubblico come il sistema sottostante che utilizzano. Sivakumar e Singh hanno

sviluppato una DPKI basata su blockchain mediante la quale le operazioni di gestione delle chiavi vengono affrontate attraverso l'uso di smart contract.

È importante notare che non tutte le violazioni della sicurezza e le preoccupazioni relative alla PKI tradizionale sono dovute alla sua architettura fondamentale, ma piuttosto a un'applicazione inadeguata dei controlli di sicurezza in un deploy PKI o alla mancanza di un'adeguata gestione delle chiavi da parte degli utenti che interagiscono con il sistema.
Problemi simili possono essere riscontrati nei modelli DPKI che offrono un'elevata resilienza dei dati ma possono essere vulnerabili a vari attacchi. A causa del ruolo svolto dalla CA, le CA pubbliche popolari hanno adottato livelli elevati di sicurezza operativa e fanno di tutto per proteggere i dati sensibili. Anche i sistemi PKI decentralizzati devono seguire rigorose misure di sicurezza. Mentre alcuni aspetti di DPKI possono fare affidamento sulla blockchain, molte attività come le operazioni di firma crittografica che richiedono chiavi segrete devono essere eseguite in un ambiente sicuro, centrale e fuori dalla blockchain.
Inoltre, mentre i sistemi DPKI offrono un'architettura trasparente e aperta, le CA nel modello PKI tradizionale hanno compiuto passi verso una migliore trasparenza offrendo servizi come i registri di trasparenza dei certificati, attraverso i quali tutti i certificati emessi vengono pubblicati sul sito web della CA.
Molti sistemi aziendali PKI sono stati implementati in vari settori, tra cui quello bancario, sanitario e della difesa. A causa della rapida crescita dell'ecosistema PKI, l'implementazione e la gestione dei sistemi PKI sono diventate più semplici. Ad oggi ci sono molte opzioni PKI on-premise e basate su cloud disponibili nel settore.
Per gli ecosistemi privati in cui i dettagli sull'emissione del certificato e sulla ristrutturazione delle parti devono rimanere privati, il costo dell'implementazione di una DPKI privata con un numero elevato di nodi potrebbe non essere finanziariamente fattibile.

RIFERIMENTI

1. A. S. AlQahtani, Z. El-Awadi and M. Min, "A Survey on User Authentication Factors," 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2021, pp. 0323-0328, doi: 10.1109/IEMCON53756.2021.9623159.
2. C. Esposito and M. Ciampi, "On Security in Publish/Subscribe Services: A Survey," in IEEE Communications Surveys & Tutorials, vol. 17, no. 2, pp. 966-997, Secondquarter 2015, doi: 10.1109/COMST.2014.2364616.
3. Sharmila K, Janaki V, Nagaraju A. A Survey on User Authentication Techniques. Orient.J. Comp. Sci. and Technol;10(2)
4. Madwesh, Mahith and Nadimpalli, Sandeep Varma, Survey on Authentication Techniques for web applications (May 17, 2019). Proceedings of the Second International Conference on Emerging Trends in Science & Technologies For Engineering Systems (ICETSE-2019).
5. Aldosary, Maha, and Norah Alqahtani. "A Survey on Federated Identity Management Systems Limitation and Solutions." International Journal of Network Security & Its Applications (IJNSA) Vol 13 (2021).
6. Baldoni, Roberto. "Federated Identity Management systems in e-government: the case of Italy." Electronic Government 9.1 (2012): 64-84.
7. Delft, Bart van, and Martijn Oostdijk. "A security analysis of OpenID." IFIP Working Conference on Policies and Research in Identity Management. Springer, Berlin, Heidelberg, 2010.
8. Uruena, Manuel, and Christian Busquiel. "Analysis of a privacy vulnerability in the openid authentication protocol." IEEE Multimedia Communications, Services and Security (2010).
9. Malville, Eric, Jean-Michel Crom, and Gael Gourmelen. "A survey on identity federation solutions." Annales des télécommunications. Vol. 61. No. 3. Springer-Verlag, 2006.11:58
10. Clauß, Sebastian, and Marit Köhntopp. "Identity management and its support of multilateral security." Computer Networks 37.2 (2001): 205-219.11:58

11. Hunt, Ray. "PKI and digital certification infrastructure." Proceedings. Ninth IEEE International Conference on Networks, ICON 2001.. IEEE, 2001.12:00
12. Ometov, A.; Bezzateev, S.; Mäkitalo, N.; Andreev, S.; Mikkonen, T.; Koucheryavy, Y. Multi-Factor Authentication: A Survey. Cryptography 2018, 2, 1. https://doi.org/10.3390/cryptography2010001
13. M. Naedele, "Standards for XML and Web services security," in Computer, vol. 36, no. 4, pp. 96-98, April 2003, doi: 10.1109/MC.2003.1193234.
14. Agbinya, Johnson & Islam, Rumana & Kwok, Chandra. (2008). Development of Digital Environment IdentiTY (DEITY) System for Online Access. 1 - 8. 10.1109/BROADCOM.2008.52.
15. Kang, Meng, and Victoria Lemieux. "A decentralized identity-based blockchain solution for privacy-preserving licensing of individual-controlled data to prevent unauthorized secondary data usage." Ledger 6 (2021).
16. Mühle, Alexander & Grüner, Andreas & Gayvoronskaya, Tatiana & Meinel, Christoph. (2018). A Survey on Essential Components of a Self-Sovereign Identity.