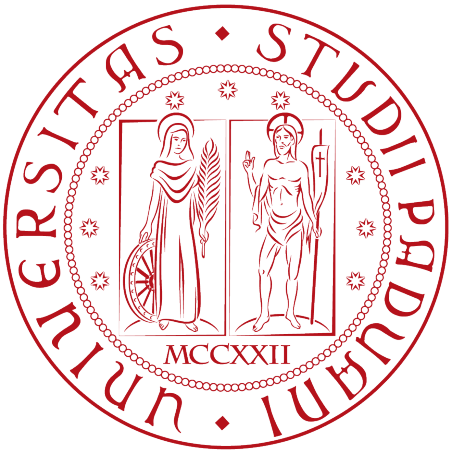
Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea in Informatica



Un futuro senza password: FIDO e gli autenticatori

Tesi di laurea

Relatore

Dott. Alessandro Brighente

Laureando

Stefano Meneguzzo

Anno Accademico 2022-2023

Sommario

Il documento qui presente fa riferimento al periodo di stage trascorso dallo studente Stefano Meneguzzo presso l'azienda Athesys s.r.l., per un totale complessivo di trecento ore. Nello specifico lo stage prevedeva lo studio del problema legato alle fragilità delle password comunemente usate per accedere alle aree personali nel web, e si proponeva di studiare una possibile soluzione per combattere i problemi di phishing e ottenere una maggiore efficienza a livello aziendale e personale per il mondo del lavoro e non. Di questo aspetto si è occupato FIDO Alliance negli ultimi anni con il progetto FIDO Authentication provando a creare una sua soluzione tramite l'utilizzo di passkeys e autenticatori. Le prime, se il progetto andrà a buon fine, finiranno con il prendere il posto delle password vantando una maggiore sicurezza e facilità di utilizzo. Per quanto riguarda gli autenticatori invece, ognuno presenta caratteristiche che lo distinguono dagli altri e per le aziende che decidono di implementare questa tecnologia può risultare utile avere un'idea chiara su quali caratteristiche presenta un particolare autenticatore. Per riuscire ad ottenere queste informazioni è necessario esaminare i metadati. I metadati sono informazioni aggiuntive associate a un'entità o a un oggetto. Nel contesto di FIDO, i metadati sono utilizzati per migliorare la sicurezza dell'autenticazione attraverso due componenti principali:

- Metadata Service Provider (MSP): L'MSP è una parte centrale del sistema FIDO. Fornisce metadati aggiornati sugli aspetti critici dei servizi FIDO, come i tipi di autenticatori supportati, le politiche di autenticazione e le chiavi pubbliche dei server FIDO. Gli utenti possono accedere a questi metadati prima di effettuare un'operazione di autenticazione, il che consente loro di verificare l'affidabilità del servizio e dell'autenticatore. Questo impedisce agli utenti di cadere vittima di attacchi di phishing o di autenticatori contraffatti.

- Metadata Statement: Questo è un documento JSON che contiene i dettagli relativi a un'autorità di certificazione FIDO, a un'autenticatore o a un servizio. Questo documento contiene informazioni come l'ID dell'autenticatore, l'algoritmo crittografico utilizzato, le chiavi pubbliche associate e altre informazioni rilevanti. Gli utenti possono utilizzare queste dichiarazioni per convalidare l'autenticatore prima di affidarsi ad esso.

Proprio quest'ultimi saranno i soggetti principali dello stage. Il principale obbiettivo dello stagista a livello di produzione di codice sarà infatti quello di creare delle classi di metadati e successivamente creare dei convertitori per passare da una versione di metadata ad un'altra.

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine al Prof. Alessandro Brighente, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.

Desidero ringraziare con affetto i miei genitori per il sostegno, il grande aiuto e per essermi stati vicini in ogni momento durante gli anni di studio.

Ho desiderio di ringraziare poi i miei amici per tutti i bellissimi anni passati insieme e le mille avventure vissute.

Padova, Dicembre 2023 Stefano Meneguzzo

Indice

Elenco delle figure

Capitolo 1

Introduzione

1.1 Introduzione

In un mondo sempre più virtuale come quello nel quale viviamo oggi, quasi ogni aspetto delle nostre vite finisce con l'essere digitalizzato, risulta pertanto importante comprendere come questi processi avvengano e quali siano quegli elementi alla base di questo fenomeno che permettono ad esso di stare in piedi. Durante il corso di questo tirocinio lo studente si è occupato proprio di questi concetti, in particolar modo di quella componente che sta alla base di tutto questo e che prende il nome d'identità digitale. Essa rappresenta le informazioni più o meno dettagliate di ogni persona che desideri crearsi una propria rappresentazione virtuale permettendo loro di compiere diversi tipi di operazioni in base alle autorizzazioni e ai contesti presenti. Il più semplice utilizzo dell'identità digitale è rappresentato dall'impiego di credenziali come username e password per l'accesso ad aree riservate, proprio quest'ultime saranno il soggetto principe di questo documento, sono ormai anni infatti che il mondo del web utilizza le password come strumento base per garantire un certo livello di sicurezza online, lo studente si chiede quindi: sono poi realmente così sicure? O esiste qualche tipo di soluzione migliore sotto diversi punti di vista?
A queste domande ha provato in questi anni a trovare una risposta l'azienda FIDO Alliance, proponendo una valida alternativa denominata passkey, essa permette a chiunque volesse identificarsi in una determinata piattaforma virtuale, di riuscire a farlo in maniera più rapida, più sicura ed efficiente rispetto alle classiche password. L'intero percorso di studio e realizzazione delle attività di stage ha avuto luogo principalmente in modalità di smartworking presso l'azienda Athesys con sede a Padova. In totale lo studente ha impiegato circa trecento ore per svolgere l'intero progetto.

1.2 L'azienda

Lo stage è stato svolto presso Athesys, un'azienda italiana che nasce nel 2010 con l'idea di fornire servizi e soluzione in ambito IT e cybersecurity. L'azienda possiede una sede a Padova in Via Giacinto Andrea Longhin, 79, nella quale ha lavorato lo studente. Athesys può offrire un ottimo livello di consulenza in diversi ambiti, tra i servizi offerti possiamo trovare[1]:

[1] *Athesys.* URL: https://security.athesys.it/servizi/.

1. security: lo scopo dell'azienda qui è quello di salvaguardare la conservazione dei dati, gestire la loro esposizione, adeguare le authentication policy al General Data Protection Regulation (GDPR) per tutelare il business proprio e quello degli stakeholder, per le aziende che ne desiderano beneficiare. In particolare l'azienda si occupa di gestire situazioni legate a:
   - Access manager e quindi suppporto legato ai processi di autenticazione e autorizzazione per accedere a risorse in modo centralizzato.
   - Identity governance che comprende l'insieme degli strumenti che, se implementati correttamente, aiutano la tua azienda ad eseguire un controllo efficace sui rischi d'accesso.
   - Privileged managment grazie al quale verranno forniti gli strumenti necessari per effettuare una mappatura di utenti, gruppi, ruoli e privilegi sia interni che esterni all'azienda che decide di usufruire di questo servizio in modo che possa facilmente gestire la suddivisione dei privilegi in base al tipo di utente autenticato.
2. DBMS: l'azienda qui offre la possibilità di ricevere supporto per quanto riguarda la gestione dell'intero ciclo di vita del tuo database aiutandoti a renderlo scalabile e sicuro. Nello specifico è possibile ottenere un supporto per il proprio database a livello di design, sviluppo e prevenzione o gestione di eventuali problematiche.
3. Cloud: esiste la possibilità di ricevere supporto nel cercare di ottimizzare la propria infrastruttura IT e integrarla o sostituirla con il cloud, in modo da abbattere i costi e ottimizzare la sicurezza.
4. Software development: A seconda delle esigenze del cliente l'azienda offre su richiesta la possibilità di sviluppo software con diverse metodologie(Agile, Waterfall, ecc.).
5. Business intelligence: Athesys offre assistenza alle aziende per costruire ecosistemi orientati al business e a migliorare l'aspetto tecnologico della gestione e dell'utilizzo dei dati. Un team di esperti sarà pronto a dare supporto nella raccolta e strutturazione dei dati e alla reportistica aziendale in base ai più avanzati standard tecnologici.

1.2.1 Monokee

Tra le tecnologie delle quali l'azienda sfrutta i prodotti una menzione speciale la merita Monokee. Essa è una startup che da qualche anno ha posto il proprio focus nella ricerca di sviluppare tecnologie e soluzioni per quanto riguarda la gestione dell'identità online, e di tutto quello che ne consegue in termini di autenticazione e autorizzazione di chi desidera accedere ad una piattaforma virtuale. Athesys sfrutta questa collaborazione per fornire accesso intelligente a dipendenti e stakeholders per connettersi in totale sicurezza ad applicazioni o dispositivi, aiutando a gestire e proteggere i dati sensibili.

1.2.2 L'offerta di stage

Nel tentativo di trovare una proposta di stage che potesse essere interessante lo studente ha deciso di partecipare a Stage-it, un evento a cui da anni l'università di Padova aderisce per permettere appunto agli studenti di poter ascoltare le numerose aziende

presenti con lo scopo di approcciarsi al mondo del lavoro e riuscire a mettersi alla prova in vista di un futuro prossimo. Quest'anno all'evento erano presenti una sessantina di realtà aziendali, ciascuna delle quali si è presentata con la propria postazione per permettere a chi fosse interessato di farsi conoscere. Lo studente ha preso parte a diversi colloqui con lo scopo di cercare un'azienda che proponesse un progetto interessante, particolare e possibilmente in un qualche ambito della sicurezza informatica. Alla fine dell'evento lo studente ha analizzato attentamente le diverse proposte di tirocinio e la scelta è ricaduta per l'appunto sul progetto presentato in questo stesso documento, le ragioni di questa scelta ricadono nel fatto che quest'ultimo sembrava il più interessante per mettersi alla prova andando a mettere mano ad un progetto con grandi aspettative per il futuro, tutto questo sempre restando nell'ambito della sicurezza informatica come desiderato dallo studente.

1.3 Il problema delle password

Alla base di tutti i discorsi che andremo a fare nel corso di questo documenti, c'è il concetto d'identità digitale, che in poche parole potremmo descrivere come la rappresentazione digitale dell'identità di una persona con il suo nome, cognome, data di nascita, indirizzo, ecc. Ogni individuo può averne molteplici, in siti web diversi, in base alle proprie esigenze. Quello che ci garantisce che questa identità digitale e tutte le informazioni legate ad essa possano restare un qualcosa di personale e privato di un individuo è il fatto che, spesso nel poter accedere alle informazioni private di una persona è richiesto l'inserimento di apposite credenziali. Esse sono rappresentate solitamente da un ID, come uno username o un indirizzo e-mail(pubblico) e una password(privata), le quali sono il principale strumento per garantirci che il profilo di una persona sia di esclusiva priorità della persona stessa. Dal momento però che un individuo qualsiasi viene a conoscenza dello username o e-mail con la quale un utente si è registrato, riuscire ad ottenere anche la password per entrare nelle aree private del malcapitato può essere relativamente facile. Spesso le persone tendono infatti a usare sempre la stessa password per siti diversi, nonostante le numerose raccomandazioni di cambiarle spesso e diversificarle. All'atto pratico questo può però creare alcuni svantaggi alle persone, ricordare diverse password, le quali continuano ad essere modificate ogni mese magari, può sicuramente essere complicato. Spesso avviene infatti che le persone dimentichino le chiavi di sicurezza per accedere, e sia quindi necessario recuperarle o modificarle. Il processo di recupero password può essere però snervante e una perdita di tempo per le persone, soprattutto se stiamo parlando di credenziali utili ad effettuare l'accesso in un'account aziendale. Bisogna comunque ammettere che al giorno d'oggi sia possibile utilizzare un password manager, uno strumento in grado di memorizzare le nostre password e tenerle al sicuro in modo da non perderle. Purtroppo però le password hanno un altro svantaggio: sono spesso facili da rubare. Può capitare infatti che i server, che possiamo vedere come dei grossi archivi di dati, di alcune aziende vengano attaccati da qualcuno, o che le persone siano vittime di alcuni ingegnosi metodi che vengono messi in pratica con lo scopo di rubare o farsi dare le password dalle persone. A questo punto, una volta ottenute le password di alcuni utenti, è possibile effettuare l'accesso alle loro aree private senza troppe difficoltà.

1.3.1 Le passkeys

Nello studiare il problema sopra presentato, sono diverse le soluzione offerte nel corso degli anni, in questo documento verranno viste però quelle analizzate durante il periodo di stage, riguardanti le passkeys. L'azienda FIDO negli ultimi tempi ha infatti pensato ad un proprio modo per venire incontro al problema, per farlo si è pensato ad un protocollo che prende il nome di FIDO2, esso farà uso della cifratura a chiave pubblica. Al momento della registrazione ad ogni utente, oltre al proprio identificativo verrà accostata una coppia di chiavi pubblica e privata, generate attraverso un autenticatore, esso è un particolare strumento, la maggior parte delle volte integrato nei dispositivi quali PC o smartphone e utile appunto per la creazione di queste chiavi e per la fase di verifica dell'utente. Da questo punto l'utente avrà quindi una chiave pubblica che andrà ad essere salvata nel server del servizio online al quale ci siamo registrati e collegata al nostro ID ed una privata utile per dimostrare chi siamo. In parole povere quando proveremo ad autenticarci, il servizio online ci chiederà chi siamo, noi gli rispondiamo fornendogli l'ID, e per verificare se stiamo mentendo ci verrà quindi chiesto di usare la chiave privata, che essendo appunto privata, dimostrerà la nostra identità. L'insieme di chiave pubblica e privata prende il nome di passkey. È inoltre importante notare come questa chiave privata, resti memorizzata nel nostro dispositivo non andando quindi mai a finire nel server, in questo modo se un server venisse attaccato nessuna informazione personale andrebbe ad essere rilasciata (al contrario di ciò che avviene con l'utilizzo delle password)[2].

1.4 L'idea

Una delle realtà aziendali che si interessa di collaborare con FIDO Alliance e trovare una soluzione alla questione delle password è proprio Athesys, la quale si occupa nello specifico di soluzioni di Identity and Access Management(IAM). Nell'ambito dell'autenticazione multifattore, il riconoscimento biometrico gioca un ruolo fondamentale. Lo scopo dello stage sarà quindi quello di studiare ed analizzare le specifiche del protocollo WebAuthN per l'utilizzo di credenziali biometriche nella sua implementazione a cura della FIDO Alliance, l'organizzazione principale ideatrice delle passkey. Nello specifico, lo studente lavorerà sulle differenze di protocollo e di formato dati dalla versione V2 dei metadati relativi agli autenticatori, alla versione V3 andando a realizzare una libreria dedicata alla conversione tra questi dove possibile.

1.5 Pianificazione

Lo stage svoltosi aveva come principale obiettivo lo studio e la comprensione di cosa fossero e come funzionassero i metadati e soprattutto la codifica di una libreria in grado di passare dalla versione V2 a quella V3 e viceversa. Il carico di lavoro complessivo può essere diviso in due parti:

1. Formazione sulle tecnologie per l'identità digitale (1.5 settimane): verranno studiate le basi della tecnologia digitale e dei protocolli di Single Sign-On (SSO). In aggiunta, verranno studiate le tecnologie basate sulla biometria e sui meccanismi password-less ed input-less (WebAuthN e FIDO Alliance).

[2] *PasskeyFIDO*. URL: https://fidoalliance.org/passkeys/.

- Deliverable numero 1: documentazione tecnologie SSO e soluzioni password-less ed input-less. Indicazione dei protocolli, funzionalità, pro e contro.

2. 3 cicli di sviluppo con metodologia AGILE (2 settimane)
   - Studio e progettazione funzionalità: concepts indicati nello standard
   - Meeting interno e condivisione stato avanzamento lavoro: presentazione dei risultati e delle opportunità associate con la soluzione candidata in esame.
   - Implementazione: implementazione in typescript delle classi e delle logiche necessarie per riflettere il contenuto dello standard in modalità API REST.
   - Documentazione: integrazione della documentazione aziendale
   - Deliverable numero 2: manuale d'utilizzo per la libreria e le Application Program Interface (API) realizzate.

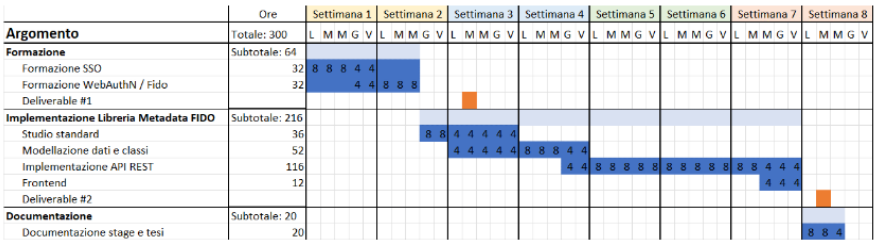


Figura 1.1: Diagramma di Gantt - Piano di lavoro

Come indicato nella figura precedente, il totale del lavoro pianificato ammonta a 300 ore. In figura si evidenzia la pianificazione generale delle attività di stage. Nello specifico, le attività sono suddivise per categorie (evidenziate in grassetto nei titoli di riga e in azzurro nelle colonne: Formazione, Implementazione, Documentazione) e per ognuna vi è indicato il numero di ore pianificate per ogni giornata di stage (in nero su campo blu). I deliverable per ogni gruppo sono indicati generalmente in consegna il martedì successivo al termine delle attività.

1.6 Obiettivi

Di seguito sono elencati gli obiettivi, i quali possono essere suddivisi in tre principali categorie. Nella prima, caratterizzata dalla marcatura <ob>, sono rappresentati i requisiti obbligatori, vincolati in quanto obiettivo primario richiesto dal committente. Nella seconda categoria, contraddistinta dalla marcatura <de>, sono invece presenti i vincoli desiderabili, non vincolati o strettamente necessari, ma dal riconoscibile valore aggiunto. Infine nella terza categoria, con la marcatura <op>, sono indicati i requisiti opzionali, rappresentante valore aggiunto non strettamente competitivo. A fianco a ad ogni marcatura sarà presente un numero progressivo per indicare il requisito in maniera univoca.

- ob01: comprensione del protocollo WebAuthn di FIDO 2.0
- ob02: comprensione delle tipologie di Metadata definiti nell'abstract `https://fidoalliance.org/specs/fido-v2.0-rd-20180702/fido-metadata-statement-v2.0-rd-20180702.html`

- ob03: implementazione di un modulo di parsing e validazione dei metadata (Statement V2)
- de01 : comprensione delle tipologie di Metadata definiti nell'abstract (https://fidoalliance.org/specs/mds/fido-metadata-statement-v3.0-ps-20210518.html)
- de02 : implementazione di un modulo di parsing e validazione dei metadata (Statement V3)

1.7 Modalità di svolgimento

Per quanto riguarda le modalità di svolgimento dello stage , lo studente potrà scegliere se svolgerlo totalmente da remoto, parzialmente o presentarsi in azienda negli orari lavorativi a seconda delle sua esigenze. Athesys mette infatti a disposizione le proprie postazioni in caso lo stagista preferisca recarsi in ufficio, può così decidere se recarsi a Padova in Via Longhin 79, dove l'azienda ha sede. L'orario lavorativo coinvolge tutti i giorni dal lunedì al venerdì in un arco temporale che va dalle 9.00 alle 18.00 con all'interno una pausa di 1h per poter mangiare. Allo studente è richiesto di prendere parte ogni settimana (solitamente di giovedì) ad una riunione con il tutor aziendale per fare il punto sulla situazione e avere nuove direttive su come procedere. In aggiunta, i referenti interni per il team di sviluppo rimarranno a disposizione per supporto e formazione secondo i loro calendari. Al candidato verrà riconosciuto un contributo pari ad 1 buono pasto al giorno.

1.8 Organizzazione del testo

Il secondo capitolo approfondisce tutti quegli argomenti teorici, la cui conoscenza risulterà di fondamentale importanza per riuscire a comprendere al meglio tutti quegli esperimenti descritti nel capitolo successivo e che rappresenteranno la parte effettiva di cosa è stato fatto dallo studente durante il corso del periodo di stage. I principali argomenti saranno riguardanti, i concetti d'identità digitale e tutte le sue caratteristiche, gli autenticatori cosa sono e che ruolo svolgono, il concetto di passkey, la realtà di FIDO Alliance, in che ambito opera e quali sono i progetti che hanno ideato.

Il terzo capitolo descrive gli esperimenti effettivamente svolti dallo studente. Si analizzerà qual è il problema già accennato nel capitolo introduttivo, andando ad aggiungere maggiori dettagli e parlandone in modo più tecnico, e si mostrerà a livello pratico cosa lo studente ha prodotto per fare fronte a questi problemi, con quale logica e perchè.

Nel quarto capitolo vengono tirate le conclusione guardando il lavoro svolto in relazione agli obiettivi prefissati, verrà valutata l'esperienza personale da un punto di vista dello studente e verranno fatti dei cenni su possibili problematiche relative all'utilizzo delle passkey e di quale si pensi possa essere la direzione nella quale le cose andranno nel prossimo futuro.

Capitolo 2

Background

Nel seguente paragrafo sono elencate tutte quelle informazioni o tecnologie la cui conoscenza può risultare utile per capire meglio come lo stage è stato svolto e con quali motivazioni sono state fatte determinate scelte rispetto ad altre. Verranno quindi introdotti i concetti base ed essenziali per comprendere correttamente il resto del documento, e permettere al lettore di riuscire a farsi una propria idea sull'operato dello studente e a trarne le proprie conclusioni.

2.1 L'identità digitale

In un mondo sempre più virtuale come il nostro, la possibilità di avere uno strumento per riuscire a farsi riconoscere e riconoscere gli altri nel web diventa di fondamentale importanza. Possiamo quindi introdurre il concetto d'identità digitale, la quale viene definita come "l'insieme delle risorse digitali associate in maniera univoca ad una persona fisica che la identifica, rappresentandone la volontà, durante le sue attività digitali"[1]. In altre parole sono quell'insieme di informazioni presenti online, in grado di rappresentare un soggetto in maniera unica. Le informazioni che essa rappresenta, sono tanto più complete, maggiore è la complessità della transazione nella quale è coinvolta. Dove con transazione si intende una serie di operazioni le quali creano una variazione in una base dati. Due sono le componenti principali che caratterizzano l'identità digitale:

- Chi sei
- le tua credenziali

Quest'ultime possono essere di vario tipo, ed anche molto complesse, ma il più classico degli esempi è rappresentato da un ID(username) e una parola segreta(password).

2.1.1 Identificazione

Il primo step che incontriamo quando vogliamo effettuare l'accesso in un'area del web dove è richiesta l'identità digitale è la fase d'identificazione, questo processo coincide con la capacità d'identificare in modo univoco un utente, un dispositivo o un'applicazione all'interno di una rete sulla base dei suoi attributi, per esempio lo username, l'ID

[1] *L'identità digitale.* URL: https://it.wikipedia.org/wiki/Identit%C3%A0_digitale.

o l'e-mail. È anche possibile utilizzare metodi d'identificazione più sofisticati, un esempio può essere l'utilizzo del riconoscimento biometrico, quest'ultimo comprende il riconoscimento del viso, dell'iride o più comunemente l'utilizzo di un sensore per riconoscere l'impronta digitale, quello che ormai è frequente usare per sbloccare lo schermo del proprio smartphone.

2.1.2 Autenticazione

Il processo di autenticazione è uno dei motivi per il quale l'identità digitale ha una così grande rilevanza nel mondo del web. Esso corrisponde a quella fase in cui in una transazione si verifica che l'identità digitale presentata sia effettivamente della persona o dell'entità corretta. Questo spesso avviene attraverso l'inserimento da parte dell'utente di una password, e il sistema verifica che essa corrisponda a quella associata all'ID. La forma più base di autenticazione è quella descritta in precedenza, in cui sono coinvolti semplicemente qualcosa che ti identifica, solitamente uno username, e qualcosa di segreto che solo chi autorizzato dovrebbe conoscere, di solito la password. Esistono però altri metodi di autenticazione, più sofisticati, che magari richiedono ulteriori passaggi per ottenere l'autorizzazione, spesso questo avviene poichè stiamo cercando di accedere a uno spazio virtuale con informazioni maggiormente sensibili. Vediamo alcuni esempi:

2FA: Questo metodo di autenticazione prevede l'aggiunta di un ulteriore strato di verifica prima di permettere all'utente di effettuare l'accesso alla propria area privata, oltre alla password, il sistema richiede infatti un altro tipo di informazione. Questa informazione aggiuntiva può essere di tre tipi:

1. Una cosa che conosci: può comprendere una password aggiuntiva o un codice PIN
2. Una cosa che hai: per esempio un telefono, un token USB(strumento in grado di generare un codice casuale), o un documento
3. Una cosa che sei: tipicamente una credenziale biometrica, come il riconoscimento dell'iride, del viso o dell'impronta

MFA: L'autenticazione a più fattori, o più brevemente MFA, è un 2FA a cui sono stati aggiunti uno o più strati di verifica per una maggiore sicurezza.

Il motivo per il quale un utente o un'azienda possa scegliere di adottare uno di questi due metodi di autenticazione rispetto al semplice utilizzo di password è che quest'ultime risultano piuttosto facili da rubare per un utente malintenzionato, solo nel 2022 sono infatti state rubate più di **24 miliardi** di password in tutto il mondo[2], aggiungere un ulteriore fattore di riconoscimento può quindi in parte aiutare a proteggere la privacy degli utenti.

2.1.3 Autorizzazione

Questo step segue quello dell'autenticazione, e in questa fase vengono concessi diversi livelli di privilegio in base al tipo di utente autenticato. Per fare un esempio molto semplice, in un sito di una scuola, un preside può avere accesso ad aree o impostazioni

[2] *Password Statistics*. URL: https://us.norton.com/blog/privacy/password-statistics#:~:text=In%202022%2C%20over%2024%20billion,%2C%20weak%2C%20or%20reused%20passwords..

differenti da quelle alle quali può avere accesso un professore, piuttosto che uno studente. Il professore potrebbe ad esempio avere tra le sue opzioni, la possibilità di modificare i voti degli studenti, mentre gli studenti avranno solamente la possibilità di visualizzarli.

2.1.4 Ulteriori proprietà

Sono inoltre da considerare altre proprietà che consentono all'identità digitale di acquisire l'importanza che oggi ha[3]:

- **Confidenzialità**: è la capacità del sistema d'impedire che terze parti intercettino una transazione senza il consenso dei soggetti interessati, spesso per riuscire ad ottenere questo fine è necessario ricorrere alla crittografia.
- **Integrità**: questa proprietà garantisce che ciò che si riceve sia uguale al contenuto di ciò che è stato inviato, per avere la certezza che nel tragitto, l'oggetto d'interesse non sia stato manomesso o danneggiato. Per riuscire a garantire questo, spesso si ricorre all'utilizzo della firma digitale e altre tecniche di crittografia che sfruttano l'utilizzo di chiavi pubbliche e private.
- **Autenticità**: quando si riceve un messaggio bisogna riuscire ad avere la certezza di chi sia il mittente, purtroppo falsificare questa informazione può risultare relativamente facile, ma è possibile cercare di ovviare a questo problema tramite l'utilizzo di tecniche crittografiche.
- **Non ripudio**: serve a garantire l'avvenuta spedizione o l'avvenuta ricezione di una certa transazione. Abbiamo quindi nel primo caso il **non ripudio della sorgente**, che serve a provare chi è il mittente dei dati in una transazione, nel secondo caso abbiamo invece il **non ripudio della destinazione**, il quale prova che i dati siano arrivati ad un certo destinatario.

2.1.5 Leggi sulla privacy

A causa della crescente diffusione d'internet tra la popolazione piano piano anche le leggi e le regolamentazioni a livello statale e internazionale hanno cercato in tutti i modi di stargli dietro, ciò è avvenuto anche nel caso dell'identità digitale, con una serie di leggi che vanno a tutela dei cittadini e dei loro dati presenti nel web.
Il 25 maggio del 2018, negli stati dell'Unione Europea è entrato in vigore il GDPR, una serie di regolamentazioni racchiuse in 99 articoli, che hanno il preciso scopo di dare delle linee guida riguardo il modo in cui le imprese possono raccogliere e gestire i dati dei propri clienti. In caso di non conformità da parte di qualche azienda, le sanzioni potrebbero risultare parecchio pesanti. Oltre ad un grande danno d'immagine al quale sicuramente l'azienda dovrebbe far fronte, ulteriori provvedimenti economici saranno addebitati alla stessa. Sono infatti previste multe pari al 4% del fatturato annuo dell'azienda o 2 milioni di euro a seconda di quale sia l'importo maggiore, nel caso l'impresa venga colta a violare gli articoli del regolamento[4]. È comunque possibile consultare l'intero documento dal sito ufficiale(`https://gdpr-info.eu/`).

[3] Darran Rolls Morey J.Haber. *Identity Attack Vectors: Implementing an Effective Identity and Access Management Solution.* Apress, 2020.
[4] *Athesys - GDPR.* URL: `https://security.athesys.it/gdpr/`.

2.2 Crittografia a chiave pubblica

Nel mondo della crittografia l'obbiettivo è principalmente quello di nascondere un certo messaggio in maniera tale che solo il destinatario possa vedere cosa c'è scritto, questo processo può essere effettuato con l'ausilio di una o più chiavi crittografiche. Per dare un'idea di come il processo venga svolto a grandi linee possiamo vedere un esempio di cifratura simmetrica(che usa una sola chiave). Supponiamo di avere un testo in chiaro(plaintext), visibile a tutti, con scritto "hello", il quale ora viene criptato con un'apposita chiave, come può essere "2jd8932kd8", una volta utilizzata la nostra chiave per nascondere il testo, esso diventa "X5xJCSycg14="(ciphertext). All'apparenza questa nuova stringa può sembrare priva di significato, ma riutilizzando la stessa chiave è possibile riottenere il testo in chiaro.

Listing 2.1: Esempio di cifratura del testo con una sola chiave

```
Plaintext + key = ciphertext:

hello + 2jd8932kd8 = X5xJCSycg14=
Ciphertext + key = plaintext:

X5xJCSycg14= + 2jd8932kd8 = hello
```

Nella crittografia a chiave pubblica, conosciuta anche come crittografia asimmetrica, è caratterizzata dalla presenza di due chiavi: una pubblica e una privata. Come è possibile intuire, la chiave pubblica di un individuo può essere diffusa al pubblico senza problema, mentre quella privata va tenuta segreta. Utilizzando una delle due chiavi possiamo criptare il nostro messaggio e successivamente decriptarlo utilizzando l'altra e viceversa. Grazie a questa tecnica è possibile per esempio per un giornalista o qualcun altro pubblicare la propria chiave pubblica in modo che chi volesse mandare un messaggio con lo scopo di farlo leggerlo solo allo stesso giornalista potesse usare la sua chiave pubblica per cifrare il testo, e a questo punto solo il giornalista stesso, il quale chiaramente possiederà la propria chiave privata sarà in grado di decriptare il testo per poterlo leggere, questo garantisce un buon livello di privacy nello scambio di messaggi. È anche possibile però, usare la propria chiave privata come firma per garantire l'autenticità di un certo messaggio, una persona che riceve un messaggio criptato da un altro individuo può verificare l'autenticità del mittente provando a decriptare il messaggio con la chiave pubblica dello stesso, se il risultato sarà un messaggio in chiaro allora avremo la certezza che il mittente sarà anche colui che possiede la chiave privata con la quale il messaggio era stato criptato.

2.3 Autenticatori

Prima di proseguire con la spiegazione del tirocinio, è fondamentale capire di cosa stiamo parlando quando ci riferiamo ad un autenticatore. Nel progetto FIDO (con il protocollo FIDO2) si è pensato di aggiungere al processo di autenticazione un nuovo attore: l'autenticatore appunto. Esso può essere pensato come un'entità astratta che può trovarsi all'interno dei dispositivi degli utenti (platform authenticators) come smartphone, tablet, PC, o può essere un componente esterno, come una chiavetta (roaming authenticators) e si occupa della creazione della coppia di chiavi crittografiche durante il processo di registrazione. Quindi procede con il tenere la chiave privata appena creata e condivide invece la corrispondente chiave pubblica con il server del

servizio online. Quando l'utente proverà quindi ad autenticarsi, verrà eseguita la fase di user verification (svolta dall'autenticatore stesso) nella quale sarà richiesta qualche tipo di interazione per confermare che l'utente sia il soggetto atteso. A questo punto, confermata l'identità dell'utente viene usata la chiave privata per firmare una challenge generata dal server, questa viene reinviata allo stesso, che ne verifica la correttezza utilizzando la corrispettiva chiave pubblica che è stata precedentemente accostata all'account dell'utente.

2.4 Passkey

Concetto altrettanto importante è quello delle passkey, pensate per essere il sostituto all'utilizzo delle password, essendo più veloci, facili da usare e sicure rispetto a quest'ultime. Quando un utente decide di registrarsi ad un servizio online, egli potrà poi scegliere di generare una passkey, la quale farà uso della crittografia a chiave pubblica e verranno di conseguenza generate una coppia di chiavi pubblica e privata, sbloccabili tramite un qualche tipo d'interazione con il dispositvo, spesso vengono utilizzati dei metodi di identificazione biometrica, come il riconoscimento dell'impronta digitale o quello del viso. Successivamente, vengono appunto generate questa coppia di chiavi, quella privata viene mantenuta all'interno del dispositivo ed accessibile solo attraverso il riconoscimento biometrico prima citato, in questo modo essa non sarà mai presente in copia dentro a nessun server, eliminando tutti quei possibili attacchi di tipo malevolo ai server che nel caso delle password permetterebbero al malintenzionato di rubare l'identità digitale a chiunque abbia effettuato la registrazione al servizio che fa uso del suddetto server. Esse permettono inoltre di effettuare l'accesso ad un determinato servizio indipendentemente da quale sia il dispositivo nel quale la passkey è stata salvata. Per verificare questa casistica è sufficiente recarsi su di un sito che supporti l'utilizzo di passkey dal nostro smartphone, effettuare il login tramite password e generare la passkey tramite l'apposita funzionalità. Successivamente, possiamo recarci sul medesimo sito tramite il nostro PC, selezionare l'opzione per l'accesso tramite passkey, connettere il dispositivo al nostro smartphone, il quale ci chiederà di sbloccare la passkey tramite l'utilizzo di credenziali biometriche per esempio, ed automaticamente sarà possibile navigare dentro la nostra area personale direttamente dal PC, da quel momento l'utente può scegliere di generare una passkey propria per il PC o continuare a usare il telefono.

2.5 FIDO Alliance

FIDO Alliance(Fast IDentity Online) è una "Open industry association", cioè un'associazione in cui diverse persone collaborano per sviluppare degli open standard. In particolare si pongono l'obbiettivo di sviluppare degli standard per l'autenticazione nel web in grado di ridurre la dipendenza di quest'ultimo dall'uso delle password. FIDO Alliance sta cercando di raggiungere i suoi obbiettivi con i seguenti metodi:

- Sviluppare specifiche tecniche che definiscono un insieme di meccanismi aperti, scalabili e interoperabili che aiutino a ridurre l'uso di password per permettere agli utenti di autenticarsi. Al momento FIDO Alliance ha rilasciato tre tipi diversi di specifiche per migliorare l'autenticazione:
  - FIDO Universal Second Factor (FIDO U2F)

  - FIDO Universal Authentication Framework (FIDO UAF)
  - FIDO2, che include il Web Authentication (WebAuthn) specification di W3C e il FIDO Client to Authenticator Protocol (CTAP)

  Tutte queste specifiche sono disponibili a chiunque voglia scegliere di adottarle.

- Creare programmi di certificazione in modo da garantire l'autenticità dei propri prodotti per dimostrare che rispettino le specifiche imposte dagli standard.

- Cercare dei buoni pattern che possano supportare il porgetto, nella sua crescita e diffuzione, tra i principali possiamo trovare Amazon, Apple, Google, PayPal ecc.

2.5.1 FIDO U2F

FIDO Universal second factor (U2F) è uno standard che supporta la tecnologia di autenticazione a due fattori. U2F permette a chi ne fa uso di mantenere l'infrastruttursa base di autenticazione tramite password, e ad essa aggiunge un ulteriore livello di autenticazione come secondo fattore. Quest'ultimo può essere l'utilizzo di un PIN o per esempio la pressione di un pulsante su un'apposita USB (token), in grado di generare un apposito codice di autenticazione.

2.5.2 FIDO UAF

FIDO Universal Autentication Framwork (UAF) permette ai servizi online di offrire un buon livello di sicurezza senza l'utilizzo di password e usando autenticazioni multi-fattore. La principale caratteristica che lo distingue rispetto al suo successore FIDO2, è la mancanza di standardizzazione e di supporto da parte di tutti i principali sistemi operativi e browser.

2.5.3 FIDO2

FIDO2 nasce da un'idea di FIDO Alliance, e si propone come nuovo protocollo di autenticazione pensato per sostituire l'utilizzo di password in maniera progressiva, con l'obbiettivo ultimo di andare a sostituirle in modo definitivo. Tutto questo tramite l'utilizzo di crittografia a chiave pubblica. L'elemento principe su cui si basa questo standard è l'utilizzo di passkeys. Esse sono un modo più sicuro, veloce e versatile di effettuare l'accesso ad applicazioni o siti web nei quali è necessario autenticarsi per accedere a determinate funzionalità. In aggiunta a questo le passkeys presentano la caratteristica di essere resistenti al fenomeno del phishing, quindi di non poter essere rubate da interfacce di siti fasulle progettate ad hoc. Questo perchè come spiegato in precedenza le passkey sono composte da una chiave pubblica e una privata, la quale non lascia mai il dispositivo al contrario delle password, che vengono memorizzate all'interno di appositi server. Di seguito andiamo a vedere quali sono i principali benefici che l'adozione degli standard introdotti con l'avvento di FIDO2 comporterebbero:

- **Sicurezza**: Le credenziali utilizzate da FIDO2 per effettuare il login sono uniche in tutti i siti web, non abbandonano mai il device dell'utente e non sono mai inviate e immagazzinate all'interno di un server. Con questo modello, viene eliminato il rischio di phishing ed ogni altro tipo di tentativo di furto di password, anche intercettando lo scambio d'informazioni tra il servizio web e l'utente, il malintenzionato non otterrebbe informazioni sensibili.

- **Comodità**: L'utente può scegliere il metodo con cui sbloccare le proprie credenziali che ritiene più comodo o adeguato, e potrà farlo con un metodo immediato, come l'utilizzo d'impronte digitali, riconoscimento del viso o l'utilizzo di un PIN.
- **Privacy**: Visto che ogni coppia di chiave crittografica è unica per ogni sito web, il loro utilizzo non può essere usato per tener traccia degli stessi utenti su siti differenti.
- **Scalabilità**: I siti web che scelgono d'implementare questa serie di standard, possono utilizzare FIDO2 tramite una semplice chiamata ad un API JavaScript che viene supportata dai principali browser e piattaforme presenti nel web.

2.5.4 Come funziona FIDO

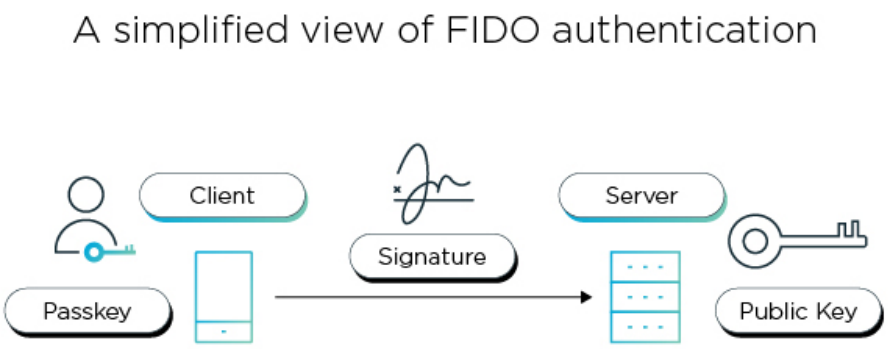


Figura 2.1: Schema funzionamento FIDO Authentication

Il protocollo di FIDO fa uso di tecniche di cifratura a chiave pubblica per riuscire a migliorare la fase di autenticazione. La fase di registrazione segue questi step:

- L'utente raggiunte il servizio online nel quale vuole registrarsi, e seleziona il tipo di autenticatore FIDO che preferisce a patto che sia in linea con le policy del servizio.
- L'utente sblocca l'autenticatore FIDO usando uno dei diversi metodi disponibili(lettura dell'impronta, un secondo device come secondo fattore di autenticazione, un PIN ecc.)
- Il device crea una nuova coppia di chiavi pubblica/privata unica per il device stesso, il servizio online e l'account dell'utente.
- La chiave pubblica viene inviata al servizio online e associata all'account dell'utente. La chiave privata invece resta localmente salvata nel device, senza mai lasciarlo.

La fase di login invece procede come segue:

- Il servizio online richiede all'utente di effettuare il login con un device già registrato e che rispetti le policy del servizio.
- L'utente sblocca l'accesso all'autenticatore FIDO usando lo stesso metodo usando al momento della registrazione.
- Il device usa l'identificatore dell'account dell'utente fornitogli dal servizio per selezionare la chiave corretta e firmare una challenge generata dal servizio stesso con la chiave privata.
- Il device invia questa challenge firmata al servizio, il quale la verifica con la chiave pubblica salvata in precedenza, e nel caso di successo permette all'utente di effettuare correttamente il login.

2.5.5 Struttura FIDO2

La specifica di FIDO2 è divisa in due componenti principali:

- **CTAP**: Client to Autenticator Protocol (CTAP) che è un protocollo di comunicazione utile per stabilire come il nostro autenticatore comunica con il browser. La prima versione creata è stata CTAP1, che si può pensare simile al concetto di U2F, era infatti pensato per essere usato in aggiunta alle password o un altro fattore. Con la creazione di CTAP2, che invece fa parte di FIDO2, l'autenticatore può essere usato sia come primo che come secondo fattore nel processo di autenticazione.
- **WebAuthN**: Web Autentication è un'API che ha lo scopo di permettere l'implementazione degli standard di FIDO da parte dei browser per un determinato sito web, in modo che gli utenti possano registrarsi ed autenticarsi ad un servizio online utilizzando la crittografia a chiave pubblica(passkey) invece delle password.

2.6 Certificazioni

I programmi di certificazione FIDO sono un elemento fondamentale per fare in modo che chiunque volesse adottare questi standard possa farlo in maniera corretta. I certificati servono a garantire che determinati prodotti rispettino gli standard richiesti, in più sono presenti dei programmi per delineare il livello di sicurezza degli autenticatori FIDO e allo stesso modo per testare e validare l'efficienza di determinati componenti biometriche. La possibilità di ottenere una certificazione porta vantaggi alle aziende produttrici, che possono quindi dimostrare l'autenticità dei loro prodotti nei confronti dei clienti distinguendosi quindi dalla concorrenza, ma anche le organizzazioni che decidono di aderire a questo programma possono beneficiarne, poichè tramite il certificato possono scegliere l'autenticatore che meglio risponde alle loro esigenze. In maniera del tutto simile anche i consumatori finali possono beneficiare degli stessi vantaggi. I prodotti che è possibile certificare FIDO sono molteplici, inclusi i server, il cui focus è più rivolto alla funzionalità e all'interoperabilità degli stessi. Di seguito però andremo a coprire la certificazione che riguarda gli autenticatori e le componenti biometriche.

2.6.1 Certificazione autenticatore

La certificazione per autenticatori è suddivisa in diversi livelli, dal primo chiamato L1, fino a quello che ad oggi è il massivo livello di certificaizone che è possibile ottenere,

cioè L3+. Per ottenere qualsiasi livello di certificaizone, è necessario innanzitutto che l'implementazione dell'autenticatore che desideriamo certificare rispetti tutti i requisiti funzionali. Ci si assicura quindi che le specifiche siano implementate in maniera corretta e verranno inoltre eseguiti alcuni test per verificare che quella specifica implementazione funzioni con differenti tipologie di server. Superata questa fase iniziale, è possibile scegliere per quale livello di certficazione si vuole fare richiesta. Esistono diversi livelli di certificazione, distinti in base ai requisiti e alle richieste minimi da adempiere. Per quanto riguarda le metodologie di valutazione, ogni livello chiaramente richiede dei requisiti differenti:

- L1: Per il livello L1 la revisione viene svolta dal personale di FIDO il cui scopo sarà quello di controllare l'implementazione dell'autenticatore, questa verifica verrà svolta attraverso la lettura della documentazione di riferimento. Oltre a questo viene svolta una validazione superficiale per controllare che non siano presenti errori banali, ma non è necessario nessun tipo di controllo in profondità.
- L2: Il secondo livello di certificazione richiede che sia presente un laboratorio certificato da FIDO nel quale sia possibile eseguire dei test. Il controllo della documentazione e quelli dell'implementazione dell'autenticatore qui richiede una revisione invece più approfondita e nel dettaglio.
- L3: L'ultimo livello di sicurezza richiede sempre la presenza del laboratorio certificato da FIDO, ma qui i test saranno più approfonditi, per esempio potrebbe essere richiesto del penetration testing del device, una revisione del codice e anche qui una revisione approfondita dell'implementazione.

Esiste poi la certificazione Delta, la quale può essere utile nel caso un determinato autenticatore venga aggiornato. Essa permette ai produttori di non dover rifare tutto il processo di certificazione da capo. Questa certificazione è possibile ottenerla se l'aggiornamento non include grossi cambiamenti a livello di comportamento o di sicurezza dell'autenticatore. è poi presente la derivative certification, questo tipo di certificazione è utile nel caso siano rilasciate nuove versioni di un autenticatore già certificato, che però non ha subito nessuna modifica. Un esempio può essere il rilascio di un nuovo modello di telefono il quale utilizza un autenticatore già esistente e certificato.

2.6.2 Certificazione biometrica

Questo tipo di certificazione è chiaramente rivolta alle componenti biometriche, e questo passaggio è tipicamente svolto prima dell'integrazione con un autenticatore FIDO. La certificazione della componente biometrica potrebbe anche essere svolta successivamente all'integrazione con l'autenticatore, ma è stata preferita questa politica anche perchè questo processo di certificazione richiede tempo e un certo livello di attenzione. Per svolgere correttamente questa certificazione, la componente biometrica deve essere inviata ad un apposito laboratorio nel quale verranno eseguiti specifici test. Dal momento che questa componente non è ancora stata integrata ad un autenticatore, al rivenditore è richiesto di fornire un apposito documento nel quale vengono descritti i diversi tipi di cambiamenti necessari per riuscire a svolgere correttamente il test d'integrazione con diversi tipi di autenticatori. I parametri usati durante questi test sono

- False Accept Rate (FAR) che descrive la probabilità che due individui diversi vengano accettati da una componente biometrica quando non dovrebbero.

- False Reject Rate (FRR) che descrive la probabilità che il riconscimento di un individuo fallisca quando non dovrebbe.
- Impostor Attack Presentation Match Rate (IAPMR) questo serve a misurare quando un individuo crea delle false credenziali biometriche e le usa per cercare di autenticarsi con il profilo di un altro utente.

Per quanto riguarda il FAR e il FRR, questi parametri verranno testati in laboratorio facendo partecipare diverse persone, e creando dei test online e offline si verifica in maniera empirica qual è il valore di ogni componente da testare. Per superare il test il FAR deve essere inferiore a 1:10.000. Per il FRR è invece richiesto che sia inferiore a 3:100. Nel verificare quale sia il IAPMR, il laboratorio si occuperà di creare delle credenziali fake e proverà a verificare caso per caso se rispetta i requisiti che si vogliono raggiungere.
Una volta ottenuta la certificazione è possibile usare questa componente biometrica per qualsiasi livello di certificazione di un autenticatore, ma è richiesto specificatamente che sia certificata solo per i livelli L2+ e L3.

2.7 Metadata Service e Metadata Statement

I metadata Service di FIDO Alliance, è una repository centralizzata in cui i rivenditori di autenticatori, una volta ottenuta la certificazione, possono mettere i metadati dei loro autenticatori per permettere a chi volesse di farne il download. Tramite questo strumento è possibile verificare alcune importanti informazioni degli autenticatori, come il loro livello di sicurezza o se sono stati eseguiti aggiornamenti e come questi ne abbiano modificato il comportamento. Le aziende che più di tutte potrebbero trovare utile questo servizio sono le agenzie governative o quelle che si occupano di gestire dati sensibili. Per loro risulterà infatti molto importante cercare autenticatori che abbiano almeno un livello minimo di certificazione come requisito base, per ovvi motivi di sicurezza. Queste aziende potranno quindi visionare tutti i metadata statement ciascuno in riferimento ad un particolare modello di autenticatore, andando a descriverne tutte quelle caratteristiche rilevati come una descrizione, l'algoritmo di protezione delle chiavi crittografiche e il loro formato di rappresentazione o se l'autenticatore svolge il ruolo di primo o secondo fattore di autenticazione. È comunque possibile consultare la documentazione completa al seguente link: `https://fidoalliance.org/specs/fido-v2.0-rd-20180702/fido-metadata-statement-v2.0-rd-20180702.html`.

Capitolo 3

Descrizione dello stage

In questo paragrafo verrà descritto il lavoro svolto dallo studente durante il suo periodo di stage presso l'azienda Athesys. L'obbiettivo primario riguardava la creazione di un'adeguata rappresentazione dei metadati nelle versioni V2 e V3 con conseguente aggiunta di una funzionalità per convertire i metadati da una versione ad un'altra.

3.0.1 Gli strumenti

Gli strumenti principali che sono stati utilizzati durante lo svolgimento di questo stage sono:

- **Angular**[1]: Framework Typescript, completamente open source sviluppato da Google. Esso è il successore di AngularJS, il quale utilizzava Javascript ed era il primo framework Javascript a introdurre il patttern della dependency injection. Le differenze principali con il suo predecessore sono l'utilizzo di una gerarchia di componenti come caratteristica architetturale primaria, fare uso di una sintassi leggermente differente, il grande utilizzo dei moduli, compilazione asincrona dei template, supporto per eseguire applicazioni angular su server e dynamic loading. Angular è noto per la caratteristica di creare single page applications e la possibilità di costruire su un back-end composto da servizi REST. Esso offre un design pattern di tipo Model-View-ViewModel nativo. Nel contesto dello stage, Angular è stato utilizzato per creare le classi utili per poi andare a sviluppare le funzioni di conversione.

- **Typescript**: linguaggio di programmazione open source sviluppato da Microsoft. Esso nasce con l'idea di estendere la sintassi di JavaScript, in modo che qualunque programma scritto in JavaScript, sia in grado di funzionare anche con TypeScript senza bisogno di alcuna modifica. La principale funzionalità che questo linguaggio mette a disposizione in più rispetto a JavaScript è sicuramente la possibilità di tipizzare i dati, in modo da evitare alcuni possibili errori che in fase di compilazione potrebbero non essere riconosciuti. Un esempio potrebbe essere:

Listing 3.1: Esempio di errore visibile a compile time

```
let firstName: string = 'John';
let age: number = 30;
function add(a: number, b: number): number {
```

[1] *What is Angular*. URL: https://angular.io/guide/what-is-angular.

```
            return a + b;
        }
```

Se qualcuno dovesse provare a chiamare questa funzione dandole come parametri firstName e age, sarebbe visibile l'errore senza bisogno di eseguire il programma. In particolare lo studente ha utilizzato TypeScript essendo il linguaggio richiesto dall'azienda.

- **Visual Studio Code**: Editor di codice sorgente, sviluppato da Microsoft per sistemi operativi quali Windows, Linux, MacOS. Offre un ottimo livello di supporto per i programmatori e garantisce loro una certa facilità di utilizzo. Offre inoltre un buon sistema di controllo per l'integrazione con Git. VS Code è basato su Electron, un framework che offre la possibilità di sviluppare applicazioni. L'uso di questo editor è stato scelto per la sua praticità e facilità di utilizzo.
- **Git**: Software gratuito ed open source nato nel 2005, utilizzato per il controllo di versione distribuito. I comandi con i quali è possibile utilizzare le sue funzionalità sono accessibili tramite interfaccia a riga di comando. Essenziale durante il corso dello stage per tenere traccia di ogni modifica o aggiunta nel codice.
- **GitHub**: Servizio di hosting per progetti software. Nome che deriva dal fatto che questa piattaforma nasce con l'idea di implementare il sistema di controllo versione Git. Ad oggi è lo strumento più utilizzato dai programmatori per permettere loro di memorizzare e modificare il loro codice o quello di altri. Grazie ad esso lo studente ha potuto cooperare al meglio con l'azienda avendo entrambi accesso alla repository di lavoro.
- **Microsoft Teams**: Piattaforma per la comunicazione da remoto, rilasciata nel 2017 da Microsoft come parte della famiglia di prodotti del pacchetto Microsoft 365. Teams viene principalmente usato come strumento di messaggistica e videochiamate per motivi lavorativi. Durante il periodo della pandemia di COVID-19, questa come altre piattaforme usate per lo smartworking hanno visto la loro rete di utenti crescere in maniera sostanziale. L'utilizzo di questa applicazione è stata fondamentale per poter comunicare con l'azienda ogni settimana ed aggiornarsi sulla situazione.
- **Google Developers Codelabs**: Tipologia di laboratorio creata da Google, utile poichè permette a chi desiderasse di fare pratica nel prendere confidenza con programmi più o meno complessi mentre si viene guidati nella comprensione e svolgimento degli stessi. La maggior parte di questi laboratori permettono a chi vi partecipa di costruire una piccola applicazione o aggiungere nuove funzionalità ad una già esistente. Grazie ad esso, lo studente ha potuto iniziare a capire meglio la struttura e il funzionamento dell'API WebAuthN.

3.1 Studio del problema

Prima di tutto lo studente è stato posto davanti al problema. È stato quindi osservato come nonostante lo sviluppo dell'identità digitale nel mondo del web negli ultimi anni sia stato sempre più capillare, le metodologie con le quali si garantiva un certo livello di sicurezza per quanto riguarda la protezione dei dati non sempre si sono dimostrate all'altezza. Lo strumento che ad oggi risulta essere il più usato come

chiave di sicurezza nell'autorizzare un individuo ad accedere alla sua area privata, è la password. Quest'ultima presenta però una serie di problemi che possono essere più o meno gravi a seconda delle situazioni. Questo perchè in primo luogo è spesso richiesto agli utenti, per motivi di sicurezza, di utilizzare password differenti per diversi tipi di account, capita infatti che ogni tanto alcuni server vengano attaccati e le informazioni contenute in essi vengano diffuse, oltre a rendere pubbliche la coppia di username e password di una persona per quello specifico sito, se un individuo sceglie di non cambiare mai password, è probabile che quella coppia username-password appena rubata possa essere utilizzata da qualche malintenzionato per accedere ad altre area private di altri siti web, andando a compromettere la privacy di quella persona in toto. Risulta quindi essenziale modificare la propria password ogni qualvolta si desideri registrarsi ad un diverso servizio web. Questo però può portare un'altra serie di problemi, principalmente legati ad un fattore di praticità. Può essere infatti, spesso difficile ricordarsi tutte le diverse password usate per ogni singolo account. Può quindi capitare, che ci si dimentichi ogni tanto di una specifica password e sia necessario recuperarla o modificarla, questo spreco di energie e soprattutto tempo, può causare problemi soprattutto se ciò avviene in ambito aziendale. Il tempo richiesto ad effettuare un cambio o un recupero di una password è infatti tempo lavorativo perso, è stato calcolato che le aziende perdono ogni anno moltissimi soldi a causa di questa specifica problematica. In aggiunta a tutto questo, le password sono anche relativamente facili da rubare, attaccando un server, un utente malevole potrà prendere possesso delle password degli utenti, sarà magari necessario decriptarle, ma resta comunque relativamente facile riuscire a superare questa barriera. A tutto questo possiamo aggiungere un altro tipo di metodo con il quale è possibile prendere possesso delle informazioni private di un utente: il fenomeno del phishing. Questo è una pratica con la quale un utente malevolo decide, spesso clonando l'interfaccia di un particolare sito web, di cercare di convincere la vittima ad inserire i propri dati personali. Un esempio potrebbe essere quello in cui, il malintenzionato clona il sito di una banca, ed invia un'e-mail a qualcun altro dicendogli che è stato eseguito un trasferimento di soldi dal suo conto, riportando sotto il link al sito finto dicendo all'utente di premerci sopra per annullare la transazione nel caso non fosse stato lui ad eseguirla. A questo punto l'utente, che chiaramente si troverà davanti ad un'operazione non eseguita da lui, potrebbe premere sul link, e se avrà la sfortuna che l'interfaccia del sito della banca clonato corrisponderà alla sua reale banca, potrebbe essere convinto ad inserire le proprie credenziali per effettuare l'accesso. Il problema a questo punto è che una volta inseriti i propri dati negli appositi campi, non verranno inviati alla banca che dovrebbe verificarli per permettere l'accesso, ma verranno inviati da qualche altra parte in modo che possano facilmente essere accessibili all'utente malevolo. Per dare un'idea di quanto questo fenomeno sia diffuso e possa rappresentare una vulnerabilità, basti vedere che solo nel 2022 l'89% delle organizzazioni si è trovata a dover affrontare un attacco di phishing[2].

3.1.1 Approccio

La prima parte del tirocinio si è quindi svolta cercando di capire quale fosse il problema e analizzarne le principali cause, successivamente è stato richiesto di cercare di capire cosa sia la realtà FIDO Alliance e che relazione avesse rispetto a questo specifico problema. Lo studente ha quindi consultato il sito di riferimento dell'azienda per cercare di capire più informazioni possibili. FIDO Alliance è una realtà imprenditoriale

[2]HYPR. *The state of passwordless security 2022.* Cybersecurity Insiders, 2023.

nata nel luglio 2012 con lo scopo di sopperire alla mancanza di interoperabilità tra tecnologie di autenticazione forti, e risolvere i problemi sopra descritti riguardanti le password. Per fare tutto questo, FIDO Alliance si è occupata di ideare una serie di strumenti che permettessero di migliorare la sicurezza per quanto riguarda i metodi di autenticazione online. L'azienda si è quindi occupata nel corso degli ultimi anni di sviluppare un set di standard utilizzabili per cercare di implementare un certo livello di sicurezza online nell'effettuare l'accesso alle aree personali nei vari siti web o applicazioni. Si è arrivati dunque a produrre FIDO Authentication, questo insieme di regole utile per appunto implementare un metodo di sicurezza sempre migliore. Tra questi diversi set, nel 2018 venne ideato FIDO2, esso permette agli utenti, di autenticarsi ad un servizio online tramite l'ausilio di uno smartphone o un PC senza la necessità di utilizzare delle password. Le componenti principali di questo protocollo sono due: CTAP e WebAuthN. Il primo è un protocollo essenziale per riuscire a far comunicare tra di loro i diversi device, con lo scopo di rendere questa tecnologia sempre più interconnessa, il secondo è invece un'API, utile per implementare FIDO2 nel proprio servizio web.

3.2 WebAuthN

Questa componente in particolare è stata soggetto di un maggior studio durante il corso del tirocinio, si è quindi cercato di capire il suo funzionamento e la sua importanza, tramite lo svolgimento di un breve laboratorio(`https://developers.google.com/codelabs/webauthn-reauth?hl=it#0`).

3.2.1 Scopo

Lo scopo di questo codelab, è quello di provare ad implementare l'API Web Authentication (WebAuthN), la quale permette la creazione e l'utilizzo delle chiavi crittografiche, necessarie per l'autenticazione con questo protocollo. L'API supporta l'utilizzo d'identificatori BLE, NFC e USB-roaming U2F o FIDO2, chiamati anche token di sicurezza, e un autenticatore di piattaforma che consente agli utenti di autenticarsi con le loro impronte digitali. Questo codelab permmetterà di essere guidati nella creazione di un sito web che utilizzerà una semplice funzione di riautenticazione tramite l'ausilio di un sensore per impronte digitali.

3.2.2 Piano

Per la realizzazione di questo codelab si è suddiviso il lavoro in tre principali step:

1. Registrare una credenziale usando le impronte digitali:
   - Creare la funzione 'registerCredential()'
   - Ottenere la challenge e altre opzioni dall'endpoint del server: '/auth/registerRequest'
   - Creare una credenziale
   - Registrare la credenziale all'endpoint del server: '/auth/registerResponse'
2. Costruire la UI(User interface) per registrare, ottenere e rimuovere le credenziali:
   - Creare la funzione: 'removeCredential()'

3. Far autenticare l'utente con la propria impronta digitale:
    - Creare la funzione di autenticazione: 'authetnicate()'
    - Ottenere la challenge e altre opzioni dal server
    - Verificare localmente l'utente e ottenere una credenziale
    - Verificare la credenziale all'endpoint: '/auth/signinResponse'

3.2.3 Svolgimento: Registrare una credenziale usando le impronte digitali

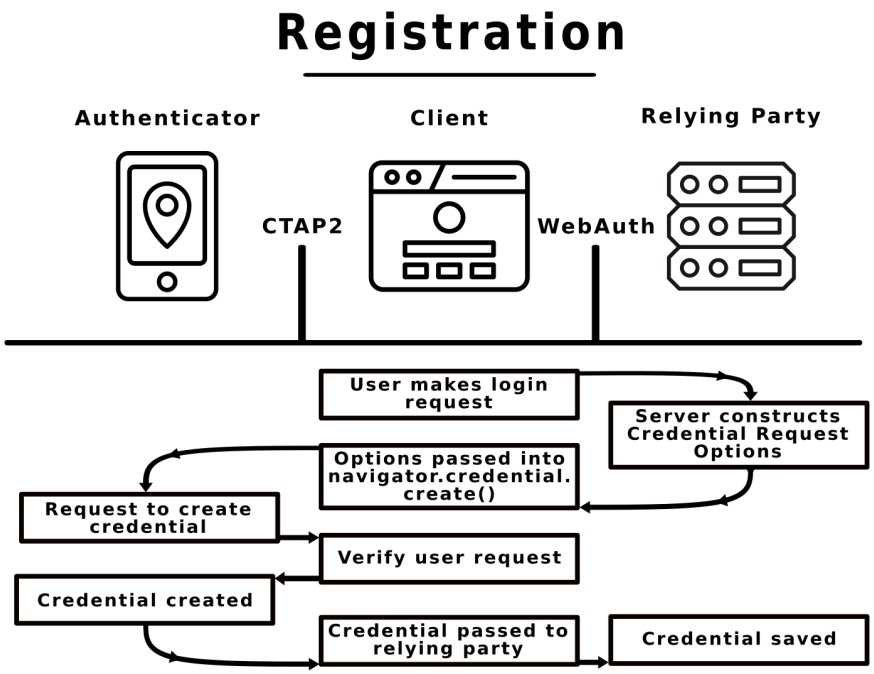


Figura 3.1: Schema registrazione credenziali FIDO

Per prima cosa si deve quindi permettere all'utente di registrare una credenziale utilizzando un UVPA, un tipo di autenticatore integrato nel dispositivo che permette la verifica dell'identità dell'utente, nel nostro caso tramite il riconoscimento dell'impronta digitale. Per riuscire a registrare una credenziale con l'impronta digitale si è prima di tutto richiesto, tramite una chiamata all'endpoint del server, una challenge con altre informazioni utili da passare a WebAuthN:

Listing 3.2: Richiesta challenge al server

```
const opts = {
  attestation: 'none',
  authenticatorSelection: {
    authenticatorAttachment: 'platform',
    userVerification: 'required',
    requireResidentKey: false
  }
```

```
};
const options = await _fetch('/auth/registerRequest', opts);
```

Poichè alcune informazioni potrebbero essere state criptate sarà necessario un passaggio extra per riuscire a visualizzarle correttamente. A questo punto è quindi possibile per il server chiamare il metodo

```
navigator.credentials.create()
```

per poter creare una nuova credenziale.
In seguito a questa chiamata, il browser interagirà con l'autenticatore cercando di verifciare l'identità dell'utente tramite l'UVPA. È importante a questo punto notare come durante l'utilizzo del sensore d'impronte digitali, che sia per creare una nuova credenziale o verificarne una già esistente, il server non vede mai questa informazione privata dell'utente, è infatti l'autenticatore che si occupa di verificare che l'impronta digitale dell'utente che sta effettuando l'interazione corrisponda a quella già presente nel server. L'informazione biometrica non lascia quindi mai il dispositivo utilizzato.
Una volta verificata l'identità dell'utente, verrà ricevuto un oggetto con dentro la credenziale che ci servirà.

Listing 3.3: Esempio di ciò che riceviamo con alcune informazioni criptate

```
{
  "id": "...",
  "rawId": "...",
  "type": "public-key",
  "response": {
    "clientDataJSON": "...",
    "attestationObject": "..."
  }
}
```

Dopo una decriptazione, possiamo quindi salvarci localmente l'ID necessario successivamente per effettuare l'autenticazione. Dopo aver inviato questo oggetto contenente la chiave pubblica al server, aspettiamo quindi che esso risponda con un codice 200, segnalando di fatto che la registrazione è avvenuta con successo.

```
return await _fetch('/auth/registerResponse' , credential);
```

3.2.4 Svolgimento: Costruire la UI(User interface) per registrare, ottenere e rimuovere le credenziali

A questo punto viene creata un interfaccia che permetta all'utente di visualizzare tutte le credenziali attualmente registrate tramite una chiamata ad un'apposita funzione

```
getCredentials()
```

che con una chiamata al'endpoint del server /auth/getKeys è in grado di ottenere le credenziali registrate e inoltre viene aggiunto un pulsante che permetta di aggiungerne altre. A seconda della disponibilità del device di sfruttare il riconoscimento delle impronte questo pulsante verrà visualizzato o sostituito con un messaggio di errore. Questa verifica consiste nel chiamare prima

```
window.PublicKeyCredential
```

per controllare che webAuthN sia disponibile e utilizzabile, successivamente viene invece chiamato

```
PublicKeyCredential.isUserVerifyingPlatformAuthenticatorAvailable()
```

per controllare che il device supporti correttamente UVPA per il riconscimento delle impronte, e nel caso una delle due restituisca un valore negativo, sarà quindi necessario mostrare a schermo un messaggio di errore per segnalarlo all'utente.
In questa interfaccia sarà inoltre disponibile un'ulteriore funzionalità, cioè quella di premere su un pulsante per rimuovere una credenziale che appartiene alla lista. Nel premere questo pulsante verrà quindi inviata una richiesta a /auth/removeKey passandogli l'ID della credenziale che si desidera rimuovere dal server e anche dalla memoria locale dell'autenticatore:

Listing 3.4: Chiamata all'endpoint per rimuovere una credenziale già registrata

```
  export const unregisterCredential = async (credId) => {
  localStorage.removeItem('credId');
  return _fetch(`/auth/removeKey?credId=${encodeURIComponent(
      credId)}`);
};
```

La funzione finale di rimozione avrà quindi la seguente forma:

Listing 3.5: Funzione invocata facendo click sul pulsante di rimozione della credenziale

```
  const removeCredential = async e => {
  try {
    await unregisterCredential(e.target.id);
    getCredentials();
  } catch (e) {
    alert(e);
  }
};
```

3.2.5 Svolgimento: Far autenticare l'utente con la propria impronta digitale

Ora è quindi presente nel server una nuova credenziale registrata e pronta ad essere eventualmente utilizzata. È quindi ora necessario dare la possibilità all'utente di riutilizzare questa credenziale, viene allora creato il tasto di autenticazione che permette all'utente appunto di autenticarsi nel sito attraverso le proprie credenziali biometriche precedentemente utilizzate nella fase di registrazione, qualora invece non venisse riconosciuta nel dispositivo la possibilità di utilizzare un lettore d'impronte o il supporto per WebAuthN, l'utente sarebbe reindirizzato nella sezione in cui è possibile effettuare il login tramite semplici username e password. Se tutto procede nel modo corretto per l'utente, sarà quindi possibile dalla pagina di login effettuare l'accesso attraverso le proprie impronte digitali, questo è possibile grazie alla chiamata che il click del pulsante di autenticazione provoca verso una funzione denominata authenticate(). Appena questa funzione viene chiamata, la prima cosa che fa è verificare che l'utente possieda nella memoria locale delle credenziali salvate, e nel caso fossero presenti l'ID di riferimento viene inviato all'endpoint del server in modo da ricevere una challenge.

Listing 3.6: Richiesta all'endpoint del server per avere una challenge

```
const opts = {};

let url = '/auth/signinRequest';
const credId = localStorage.getItem(`credId`);
if (credId) {
url += `?credId=${encodeURIComponent(credId)}`;
}

const options = await _fetch(url, opts);
```

A questo punto il server dovrebbe restituire una serie di informazioni utili:

Listing 3.7: Informazioni restituite dal server

```
    {
  "challenge": "...",
  "timeout": 1800000,
  "rpId": "webauthn-codelab.glitch.me",
  "userVerification": "required",
  "allowCredentials": [
    {
      "id": "...",
      "type": "public-key",
      "transports": [
        "internal"
      ]
    }
  ]
}
```

Tra tutte queste informazioni, la più importante è allowCredentials. Questo parametro può essere un singolo oggetto presente dentro un array o un array vuoto, in questo ultimo caso sapremo che non è stata trovata nessuna corrispondenza tra l'ID delle credenziali inviate al server e quelle effettivamente presenti nello stesso. A questo punto, nel caso sia stato trovato un match all'interno del server, sarà necessario passare alla fase di verifica, nella quale è richiesto all'utente di effettuare l'inserimento della propria impronta digitale tramite UVPA per verificare l'identità dell'utente firmando la challenge con la propria chiave privata.

Una volta verificata l'identità dell'utente, verrà recuperata la propria credenziale chiamando

```
navigator.credentials.get()
```

e verrà restituito qualcosa di simile a questo:

```
{
  "id": "...",
  "type": "public-key",
  "rawId": "...",
  "response": {
    "clientDataJSON": "...",
    "authenticatorData": "...",
    "signature": "...",
    "userHandle": ""
```

```
  }
}
```

da decodificare e inviare al server per completare la procedura.

```
return await _fetch(‘/auth/signinResponse‘, credential);
```

Se a questo punto il server risponde con un codice 200, sapremo che l'autenticazione è avvenuta con successo.[3]

3.2.6 Funzionamento

Seguiti gli step richiesti dal laboratorio, il risultato è una pagina web nella quale è possibile registrarsi inizialmente tramite l'utilizzo di username e password normalmente, e una volta fatto questo sarà possibile collegarsi e successivamente effettuare la registrazione delle proprie credenziali biometriche associate al proprio account. A questo punto è quindi disponibile una passkey che è possibile riutilizzare da questo momento in poi per riuscire ad autenticarsi con la stessa credenziale biometrica scelta precedentemente(in questo caso l'impronta digitale), senza la necessità di utilizzare la password. È stato quindi raggiunto lo scopo di questo laboratorio, dare all'utente la possibilità da prima di registrare una propria passkey tramite l'ausilio d'impronte digitali, e poi riutilizzare questa credenziale per autenticarsi senza il bisogno della password.

3.3 Autenticatori e Metadati

Successivamente a questo piccolo laboratorio, si è tornati a studiare alcuni concetti teorici molto utili per la comprensione del progetto di FIDO, in particolare delle soluzioni offerte con FIDO2. Risulta a questo punto molto importante capire cosa sono gli autenticatori e quali caratteristiche presentano. Gli autenticatori sono delle entità astratte che possono essere integrati dentro un dispositivo, come per esempio un il sensore d'impronte digitali che quasi ogni smartphone ormai adotta o un riconoscimento facciale in grado di distinguere i visi delle persone (alcuni esempi di autenticatori possono essere Touch ID, Face ID, and Windows Hello), o possono invece essere dei dispositivi esterni come per esempio delle chiavette. Ciascuno di questi autenticatori per poter essere riconosciuto ufficialmente da FIDO deve essere sottoposto ad una fase di controllo e di verifica delle funzionalità e livello di sicurezza che è in grado di garantire, in modo tale da riuscire ad ottenere uno dei livelli di certificazione offerti da FIDO(L1, L1+, L2, L2+ o L3). Queste certificazioni si distinguono per il livello di accuratezza con i quali vengono eseguiti i test, ma soprattutto per metodi con i quali garantiscono al cliente un certo livello di protezione delle proprie credenziali. In questo modo le aziende che hanno bisogno di una protezione dei dati maggiore, come può essere il sito di una banca o di un ente governativo, può richiedere il livello L2+ o L3 mentre un'azienda che non ha queste necessità può scegliere di affidarsi ad autenticatori con caratteristiche meno stringeni. Ma come possono alla fine, le aziende che scelgono di adottare i protocolli e le tecnologie FIDO a capire nel dettglie quali sono le caratteristiche di questi autenticatori? Qui entrano in gioco i Metadati. Essi sono di fatto un documento scaricabile e univoco per ogni tipologia di autenticatore, nel quale è possibile trovare diverse informazioni utili sullo stesso, come i metodi di cifratura

[3] *WebAuthN Guide*. URL: https://webauthn.guide/.

di alcuni dati, i metodi di autenticazione biometrica che supporta, il formato delle chiavi pubbliche utilizzate o più banalmente anche il codice identificativo piuttosto che una breve descrizione del prodotto. Tutte queste informazioni sono quindi consultabili dalle aziende e dai clienti per capire quale modello di autenticatore si adegua meglio alle loro esigenze.

3.3.1 Metadata

La prima richiesta pratica per lo studente è stata quindi la costruzione di apposite classi che potessero rappresentare in maniera coerente i metadati utilizzati da FIDO su Angular utilizzando il linguaggio di programmazione TypeScript. Per prima cosa è stato quindi necessario trascrivere i campi dati con il loro tipo corretto. Per alcuni campi è bastato usare i tipi predefiniti per altri è invece stato necessario definirne di nuovi. Un esempio può essere

```
    private upv: upvType[];
```

definito come segue:

```
type upvType = {
    major: number;
    minor: number;
}
```

Ad ogni campo dati è stato poi accostato un getter ed un setter, funzioni utili per permettere dall'esterno di ottenere l'informazione contenuta nel campo dati col il primo o d'impostarla con il secondo, essendo questi campi dati privati e quindi non accessibili all'infuori della classe.
Importate inoltre osservare la creazione di un'apposita funzione che permettesse d'inizializzare la classe per poterla poi andare a testare. La funzione d'inizializzazione ha la seguente struttura:

```
public static initialize<T extends MetadataV2>(data: T):
    MetadataV2 {
     const result = new MetadataV2();
     const dataJson = JSON.stringify(data);
     Object.assign(result, JSON.parse(dataJson));
     return result;
}
```

Qui possiamo notare l'utilizzo dei generics, in modo da non essere stringenti nella richiesta di un certo tipo come parametro[4]. Nel corpo della funzione quello che succede è che prima viene creata un'istanza vuota di MetadataV2, e nelle successive due righe viene preso in input il parametro e trasformato in una stringa, a questo punto viene eseguito il parsing dell'ogetto, cercando di riempire con i campi dati adeguati le voci di MetadataV2, assegnandole in maniera congrua. In fine questo nuovo oggetto di tipo MetadataV2 viene restituito come istanza della nostra classe.

3.3.2 Validazione Metadata

A questo punto possiamo osservare che ad ogni campo dati è stata inoltre associata un ulteriore funzione oltre al setter ed al getter, la funzione di validazio-

[4] *TypeScript Generics.* URL: https://www.typescriptlang.org/docs/handbook/2/generics.html.

ne. Queste funzioni vengono chiamate da un altra classe contenuta in un nuovo file(MetadataV2Validation.ts), in essa sono presenti una serie di if per verificare che ogni campo dati sia stato inizializzato in maniera corretta, qualora dovesse presentarsi un errore nella validazione di uno di questi campi dati, un messaggio di errore verrà inserito dentro un array di stringhe, il quale verrà stampato al completamento della validazione di tutti i campi. Ma cosa vuol dire validare un campo dati? Come suggerisce il nome, questa funzione ha lo scopo di assicurarsi che il valore associato al campo dati dopo l'inizializzazione sia un valore che rientra nei parametri imposti dalla documentazione. Alcune validazioni richiedono solo che una certa stringa sia corretta, altri che il formato corrisponda a quello che ci aspettiamo(per esempio quello di un URL), altre volte invece vogliamo che il valore sia contenuto in un set di valori prestabiliti, in questo caso si è spesso ricorsi all'utilizzo di enumerazioni. Un esempio che possiamo vedere è quello di authenticationAlgorithm, questo dato fa uso di un particolare enumeratore:

```
enum authAlgorithm {
    'secp256r1_ecdsa_sha256_raw' = 1,
    'secp256r1_ecdsa_sha256_der',
    'rsassa_pss_sha256_raw',
    'rsassa_pss_sha256_der',
    'secp256k1_ecdsa_sha256_raw',
    'secp256k1_ecdsa_sha256_der',
    'sm2_sm3_raw',
    'rsa_emsa_pkcs1_sha256_raw',
    'rsa_emsa_pkcs1_sha256_der',
    'rsassa_pss_sha384_raw',
    'rsassa_pss_sha512_raw',
    'rsassa_pkcsv15_sha256_raw',
    'rsassa_pkcsv15_sha384_raw',
    'rsassa_pkcsv15_sha512_raw',
    'rsassa_pkcsv15_sha1_raw',
    'secp384r1_ecdsa_sha384_raw',
    'secp512r1_ecdsa_sha256_raw',
    'ed25519_eddsa_sha512_raw'
};
```

In esso sono presenti tutti i valori che può assumere authenticationAlgorithm e ad ognuno di questi sono associati dai valori che partono da 1 e crescono in maniera progressiva. Come vedremo dopo questo avviene perchè mentre nella versione 2 dei Metadata questo valore è rappresentato da un intero che parte da 1 e cresce, nella versione 3, è invece necessario utilizzare una stringa, risulterà quindi più facile con questo tipo d'implementazione fare poi la conversione. A questo punto quindi nella funzione di validazione di authenticationAlgorithm sarà sufficiente restituire true se il valore inizializzato è contenuto nel set di valori presenti nell'enum, altrimenti false.

```
public validateAuthenticationAlgorithm(): boolean {
    return this.authenticationAlgorithm in authAlgorithm;
}
```

3.4 MetadataV2 vs MetadataV3

Di recente FIDO Alliance ha rilasciato una nuova versione dei Metadata FIDO, e non si è limitata ad aggiungere o togliere determinate voci, ma la modifica ha impattato anche il modo con cui alcuni campi dati vengono rappresentati. Oltre a questo è stato aggiornato anche il Metadata Service in modo da renderlo più facilmente fruibile a chi desiderasse consultarlo. Per prima cosa infatti, l'accesso ai Service non richiede più l'utilizzo di un token. Altra grossa novità è che ora per accedere ai Metadati non è più necessario scaricare ogni file singolarmente finendo inevitabilmente con trovarsi pieni di file differenti, ora infatti è possibile scaricare un unico file(BLOB) il quale contiene tutti i Metadata. È inoltre stata aggiornata l'interfaccia dalla quale è possibile inviare Metadati, richiedere certificazioni o inviare notifiche di sicurezza.
Per quanto riguarda i Metadata Statement veri e propri, i campi dati nei quali è richiesto un valore numerico a rappresentare un certo dato ora sono stati sostituiti con delle stringhe, più immediate nella comprensione.

```
WAS: "userVerificationDetails": [[{ "userVerification": 2 }]]
NOW: "userVerificationDetails": [[{ "userVerificationMethod": "
    fingerprint_internal" }]]
```

In particolar questo si applica ai seguenti campi dati: authenticationAlgorithms, publicKeyAlgAndEncodings, attestationTypes, userVerificationDetails, keyProtection, matcherProtection, attachmentHint, tcDisplay[5].

3.4.1 Conversione

Dopo aver capito quali sono le strutture generali dei Metadata nelle versioni 2 e 3, e quali sono stati i cambiamenti effettuati tra una versione e l'altra bisogna capire nella pratica come implementare delle funzioni in grado di effettuare queste conversioni di formato senza possibilmente perdere informazioni.
Sono quindi state create le funzioni ConverterFromV3toV2 (dentro la classe MetadataV2) e ConverterFromV2toV3 (dentro la classe MetadataV3) per passare rispettivamente dalla versione 3 alla 2 e viceversa. Queste funzioni sono strutturate in maniera del tutto simile, prendono in input il file contenente i Metadata da convertire, e campo per campo verifica che sia stato inserito un valore, e se c'è, dove necessario, viene modificato in modo da aderire allo standard dell'altra versione. Per riuscire ad effettuare queste conversioni in maniera adeguata si è dovuto prima di tutto osservare la documentazione e comprenderla in maniera corretta, a questo punto lo studente ha prodotto delle tabelle di conversione per i campi dati interessati da qualche tipo di cambiamento, in modo tale da riuscire poi a creare gli algoritmi di conversione in maniera più logica ed efficace possibile. Eccone un esempio:

protocolFamily(V2)	protocolFamily(V3)
"uaf" / ...	"uaf"
"u2f"	"u2f"
"fido2"	"fido2"

Alcune conversioni hanno semplicemente richiesto l'aggiunta o la rimozione di un dato, possiamo per esempio prendere schema, un campo dati aggiunto nella versione 3 che serve a indicare appunto la versione di Metadati della quale fa parte, l'unico

[5] *From MDS2 to MDS3.* URL: https://medium.com/webauthnworks/webauthn-fido2-whats-new-in-mds3-migrating-from-mds2-to-mds3-a271d82cb774.

valore attualmente accettabile è di conseguenza 3, non essendo presente nelle versioni precedenti. Può essere invece che capire come inserire correttamente un dato non sia così facile, un esempio è assertionScheme, che viene tolto nella versione 3, per capire che valore mettere nella 2 è quindi necessario osservare altri campi dati come protocolFamily

```
switch(metadataV3.getProtocolFamily()) {
    case 'uaf': {
        metadataV2.setAssertionScheme('UAFV1TLV');
        break;
    }
    case 'u2f': {
        metadataV2.setAssertionScheme('U2FV1BIN');
        break;
    }
    case 'fido2': {
        metadataV2.setAssertionScheme('FIDOV2');
        break;
    }
}
```

Altre volte invece può essere che un campo prima obbligatorio non lo sia più, o viceversa, possiamo osservare protocolFamily che da tipo DOMString diventa required DOMString nella versione 3. In questo caso passare da 3 a 2 non presenta alcun problema, mentre per passare da 2 a 3 nel caso nella versione 2 non sia presente nessun valore, sarà possibile ricavarlo da altri campi, come per esempio da assertionScheme, il quale ha un valore corrispondente e univoco per ogni valore di protocolFamily.

```
if(metadataV2.getProtocolFamily() === undefined) {
    switch(metadataV2.getAssertionScheme()) {
        case 'UAFV1TLV': {
            metadataV3.setProtocolFamily('uaf');
            break;
        }
        case 'U2FV1BIN': {
            metadataV3.setProtocolFamily('u2f');
            break;
        }
        case 'FIDOV2': {
            metadataV3.setProtocolFamily('fido2');
            break;
        }
    }
} else {
    metadataV3.setProtocolFamily(metadataV2.getProtocolFamily()!)
}
```

In questo frammento di codice possiamo infatti vedere come prima venga controllato se in protocolFamily della versione 2 sia presente un valore, in questo caso basterà copiarlo anche nello stesso campo della successiva versione, se invece il dato non è presente, bisognerà ricavarlo guardando assertionScheme e a partire da esso prendere il valore corrispondente.
Ci sono poi quei casi in cui è stato cambiato il formato di rappresentazione, e da valori numerici nella versione 2, si è deciso di passare a stringhe, più chiare e immediate

nella lettura e nella comprensione nella versione 3. Per effettuare la conversione qui si è sfruttata la struttura degli enumeratori, che come già detto ad ogni valore di tipo stringa hanno associato un valore di tipo numerico.

```
if(metadataV2.getAttestationTypes() !== undefined) {
    let auxAttestationTypes: string[] = [];
    metadataV2.getAttestationTypes().forEach(element => {
        auxAttestationTypes.push(attestations[element]);
    });
    metadataV3.setAttestationTypes(auxAttestationTypes);
}
```

In questo caso abbiamo attestationTypes che è un'array di valori numerici, noi li scorriamo in un ciclo e per ogni elemento, lo convertiamo grazie alla struttura del enumeratore andando a prendere il corrispettivo valore di tipo stringa e lo inseriamo in un nuovo array. Finita l'operazione prendiamo il nostro array con tute le stringhe dentro, e sfruttando il setter del corrispettivo campo dati della versione 3, inseriamo queste stringhe nel metadata appena convertito. Per passare invece dalla versione 3 alla 2, quello che faremo sarà esattamente la stessa cosa ma a parti invertite.

```
if(metadataV3.getAttestationTypes() !== undefined) {
    let auxAttestationTypes: number[] = [];
    for(const ele of metadataV3.getAttestationTypes()) {
        let index: number = attestations[ele as keyof typeof
            attestations];
        auxAttestationTypes.push(index);
    }
    if(auxAttestationTypes.length != 0) {
        metadataV2.setAttestationTypes(auxAttestationTypes);
    }
}
```

3.5 Compilazione e verifica

Una volta completata tutta la codifica dei file, è stato necessario richiamarli nel file principale (index.ts) per poi poter compilare il tutto in maniera corretta. In questo file è presente un try e catch nel quale vengono presi in input dei file di prova contenenti metadati in versione 2 e 3, successivamente verranno usate le rispettive funzioni d'inizializzazione per costruire la loro rappresentazione con le classi di metadata appena descritte. Successivamente verrà chiamata la funzione di validazione per entrambe le classi e qualora nella creazione delle classi o nella validazione si dovesse presentare un qualche tipo di errore, esso verrà catturato dal catch e stampato a schermo. Inoltre sono state chiamate le funzioni di conversione per stampare il risultato degli output prodotti in seguito alla loro esecuzione. Per l'esecuzione dell'intero progetto è possibile aprire una finestra del terminale, entrare nell'apposita cartella ed eseguire i seguenti comandi:

```
tsc index.ts --resolveJsonModule
node index.js --experimental-modules
```

Per verificare che il codice prodotto rispondesse alle nostre esigenze in maniera corretta, si è preso alcuni metadati i quali avevano una rappresentazione sia nella versione 2 che

in quella 3, è stato testato su di loro il codice, andando a verificare la correttezza delle funzioni di validazione e di conversione andando a confrontare il file prodotto con la conversione con quello che sarebbe dovuto essere il suo risultato atteso.

Capitolo 4

Conclusioni

4.1 Bilancio formativo

In conclusione a questo stage lo studente si trova quindi ad aver prodotto delle rappresentazioni dei metadati in apposite classi, relativi a diversi autenticatori, offrendo inoltre la possibilità di convertirli passando dalla versione 2 alla più recente versione 3 o viceversa. Inoltre lo studente dispone ora di una più vasta conoscenza dell'utilizzo dell'identità digitale, di cosa voglia dire usare le password, quali siano le fragilità che le caratterizzano e quali sono invece le possibili soluzioni a questi problemi provate nel corso del tempo e con quali risultati. In particolare lo stage ha offerto un'ottima conoscenza delle passkey di FIDO come soluzione al problema delle password, lo studente ha quindi potuto capire quale sia l'idea che sta dietro l'impiego di esse, e perchè possono rappresentare il futuro dell'autenticazione online.

4.2 Raggiungimento degli obiettivi

Per quanto riguarda gli obbiettivi prefissati con l'azienda Athesys all'inizio dello stage, possiamo prima di tutto cercare di valutare se i requisiti obbligatori siano stati raggiunti o meno.

- ob01: comprensione del protocollo WebAuthn di FIDO 2.0. Per quanto riguarda il primo obiettivo, lo studente ha effettivamente studiato e compreso il protocollo di WebAuthN tramite lo studio di cosa esso sia e inoltre, come descritto nel capitolo precedente, lo stagista ha svolto un laboratorio offerto da Google proprio con lo scopo di capire meglio il funzionamento del protocollo e le sue funzionalità.
- ob02: comprensione delle tipologie di Metadata definiti nell'abstract `https://fidoalliance.org/specs/fido-v2.0-rd-20180702/fido-metadata-statement-v2.0-rd-20180702.html`. La riuscita di questo requisito è stata essenziale per riuscire a rappresentare i metadati, validarli e convertirli in modo corretto.
- ob03: implementazione di un modulo di parsing e validazione dei metadata (Statement V2). Come discusso nel capitolo precedente, lo studente ha prodotto delle apposite funzioni in grado di eseguire il parsing e validare ogni singolo campo dati in modo da garantire la correttezza dei metadati rappresentati.

Passiamo quindi ora invece a verificare che anche i requisiti desiderabili siano stati completati con successo.

- de01 : comprensione delle tipologie di Metadata definiti nell'abstract https://fidoalliance.org/specs/mds/fido-metadata-statement-v3.0-ps-20210518.html. Questo requisito in maniera del tutto simile a ob02, può dirsi pienamente soddisfatto.
- de02 : implementazione di un modulo di parsing e validazione dei metadata(Statement V3). In maniera del tutto simile, anche questo requisito è stato soddisfatto allo stesso modo di ob03.

4.3 FIDO: il presente e le sfide del futuro

Per quanto riguarda FIDO e i suoi protocolli dedicati alla sicurezza online, ad oggi sono numerose le aziende che implementano questa tecnologia, tra le più famose possiamo citare Apple, Microsoft e Google. Proprio quest'ultima di recente ha deciso d'implementare l'utilizzo delle passkey come principale metodo per accedere al proprio account privato di Google.
Ad oggi l'utilizzo delle passkey sembra infatti la nuova frontiera sia per quanto riguarda la comodità di utilizzo, sia per il fattore sicurezza. Ogni cosa però può presentare i suoi problemi, e anche l'utilizzo di questa nuova tecnologia può avere alcune problematiche. Le passkey sono infatti strettamente legate al poprio dispositivo, può quindi capitare che se il device utilizzato per generare le passkey venga smarrito o rubato chiunque fosse in grado di accedervi, sbloccandolo, potrebbe potenzialmente accedere alle passkey del proprietario, questo scenario resta comunque poco probabile in quanto per sbloccare una passkey, soprattutto se accessibile tramite autenticazione biometrica, sarebbe necessario riuscire a riprodurre un'impronta digitale piuttosto che l'immagine del viso del proprietario. Bisogna inoltre considerare con gli autenticatori esterni, come possono essere le chiavette, ad oggi sono ancora abbastanza costose, potrebbe quindi non essere un grosso problema scegliere di comprarle per un'azienda, ma per quanto riguarda i privati potrebbero scegliere di virare su altri tipi di metodi di autenticazione. In aggiunta a questo le passkey sono una tecnologia relativamente recente, di conseguenza è possibile che ancora non tutti i servizi web o browser siano in grado di supportarle. È comunque possibile verificare se il proprio browser supporta l'API di webAuthN sfruttando la console del proprio browser inserendo il comando

```
window.PublicKeyCredential
```

se viene restituito undefined allora avremo un risultato negativo, altrimenti potremo proseguire con la seconda verifica nella quale controlliamo se c'è il supporto per un platform authenticator (un autenticatore integrato nel dispositivo) in questo modo:

```
    PublicKeyCredential.
        isUserVerifyingPlatformAuthenticatorAvailable().then((
        available) => {
    if(available) {
        console.log("available");
    } else {
        console.log("not available");
    }
})
```

Come si può notare, in caso di esito positivo vedremo stampata a schermo la scritta "available", nel caso contrario invece comparirà la scritta: "not available".

Difficile immaginare da qui ai prossimi anni come cambierà il mondo del web, quasi sicuramente se le cose dovessero proseguire come ci si aspetta, la digitalizzazione di sempre più aspetti del nostro mondo sarà inevitabile, risulta quindi essenziale studiare e capire quali siano i metodi migliori per proteggere la propria identità e i propri dati. FIDO offre quindi una soluzione che merita sicuramente di essere considerata e valutata per la gestione della propria identità digitale.

4.4 Valutazione personale

Lo studente in conclusione può dirsi soddisfatto delle conoscenze acquisite per quanto riguarda tutto il mondo di FIDO Alliance, con i loro protocolli e le loro soluzioni alternative all'uso delle password per poter avere delle metodologie di autenticazione sul web sempre più sicure e pratiche con il fine di migliorare l'esperienza online degli utenti. Inoltre lo stagista può dichiararsi contento del codice prodotto, in grado di convertire i metadati dalla versione V2 a V3 e viceversa. Lo stage nel complesso viene considerato come un'esperienza ampiamente positiva, che ha permesso al tirocinante di arricchire il suo bagaglio culturale sulle tematiche delle passowrd e più in generale sulla sicurezza nel web, si è avvicinato in maniera più o meno diretta con il mondo della sicurezza informatica, ha prodotto codice funzionante acquisendo quindi una maggiore familiarità con l'uso di tecnologie come Angular o TypeScript, molto utili per il futuro lavorativo e inoltre questa esperienza è stata formativa per quanto riguarda l'approccio al mondo del lavoro, imparando come lavora una realtà aziendale, con quali tempi e dinamiche e ha migliorato le capacità di cooperare con altri programmatori per riuscire a raggiungere un fine comune.

Acronimi e abbreviazioni

API Application Program Interface. 5, 20, 36

GDPR General Data Protection Regulation. 2, 9, 36

Glossario

API in informatica con il termine *Application Programming Interface API* (ing. interfaccia di programmazione di un'applicazione) si indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma. La finalità è ottenere un'astrazione, di solito tra l'hardware e il programmatore o tra software a basso e quello ad alto livello semplificando così il lavoro di programmazione. 35

cloud In informatica, con il termine cloud si intende una grande rete virtuale di server remoti, in giro per il mondo collegati insieme e pensati per funzionare come un unico grande econsistema. 2

committente Con questo termine, ci si riferisce generalmente a colui che ordina un certo lavoro con l'intento di andare ad acquistare il prodotto finale. 5

COVID-19 virus, la cui diffusione ha portato nel marzo 2020 allo scoppio di una pandemia a livello globale con conseguenti obblighi di quarantena in diversi stati. 18

cybersecurity Con il termine cybersecurity si intende quell'insieme di processi e misure di protezione con l'idea di protegge aziende e persone da eventuali attacchi informatici. 1

firma digitale Solitamente, in riferimento alla crittografia, la firma digitale serve a provare l'autenticità di un messaggio o documento. Il mittente partendo da un documento genera una stringa di dimensione fissa e univoca per quel documento, usando una funzione di hash, a questo punto firma il documento utilizzando la propria chiave privata, criptando di fatto il messaggio, a questo punto il destinatario può utilizzare la chiave pubblica del mittente per decifrare il messaggio avendo quindi la conferma di chi sia il mittente. 9

GDPR *GDPR, General Data Protection Regulation* Importante insieme di regolamentazioni sulla protezione e gestione della privacy online all'interno dei paesi dell'Unione Europea. 35

generics Essi sono un modo con cui è possibile creare pezzi di codice riutilizzabile con diversi tipi di dato, senza essere troppo stringenti. Il vantaggio è quello di poter scrivere quindi un pezzo di codice che può essere usato con tipi differenti, senza la necessità di duplicarlo. 26

id In informatica, l'id è l'abbreviazione di "identificativo" ed è un insieme di numeri o simboli che stabiliscono in maniera univoca l'identità di una persona su internet. 3, 4

metadati In informatica con metadati si fa riferimento ad un insieme di informazioni sui dati. 4, 16, 17, 32

open standard Con questo termine comunemente ci si riferisce a degli standard che sono stati resi disponibili al pubblico e che sono modificabili da chiunque voglia cercare di migliorare la versione esistente con l'obbiettivo di collaborare allo sviluppo del progetto. 11

phishing Questo fenomeno è tipicamente usato per ingannare le persone cercando di convincerle, tipicamente tramite e-mail fasulle, ad accedere a determinati siti nei quali inserire le proprie credenziali private con il chiaro scopo di andare a rubare informazioni personali ai malcapitati. 12, 19

riconoscimento biometrico In ambito informatico, con riconoscimento biometrico si intende un sistema in grado d'identificare una persona sulla base di sue caratteristiche fisiologiche o comportamentali. 4, 8

stakeholder Può essere una persona, piuttosto che un gruppo o un'organizzazione con un grande interesse nelle decisioni che vengono prese in un progetto, nonchè nella sua realizzazione. 2

token USB Essi sono delle particolari chiavette, le quali contengono dei chip in grado di generare dei particolari codici identificativi. 8

Bibliografia

Riferimenti bibliografici

HYPR. *The state of passwordless security 2022*. Cybersecurity Insiders, 2023 (cit. a p. 19).

Morey J.Haber, Darran Rolls. *Identity Attack Vectors: Implementing an Effective Identity and Access Management Solution*. Apress, 2020 (cit. a p. 9).

Siti web consultati

Athesys. URL: https://security.athesys.it/servizi/ (cit. a p. 1).

Athesys - GDPR. URL: https://security.athesys.it/gdpr/ (cit. a p. 9).

From MDS2 to MDS3. URL: https://medium.com/webauthnworks/webauthn-fido2-whats-new-in-mds3-migrating-from-mds2-to-mds3-a271d82cb774 (cit. a p. 28).

L'identità digitale. URL: https://it.wikipedia.org/wiki/Identit%C3%A0_digitale (cit. a p. 7).

PasskeyFIDO. URL: https://fidoalliance.org/passkeys/ (cit. a p. 4).

Password Statistics. URL: https://us.norton.com/blog/privacy/password-statistics#:~:text=In%202022%2C%20over%2024%20billion,%2C%20weak%2C%20or%20reused%20passwords. (cit. a p. 8).

TypeScript Generics. URL: https://www.typescriptlang.org/docs/handbook/2/generics.html (cit. a p. 26).

WebAuthN Guide. URL: https://webauthn.guide/ (cit. a p. 25).

What is Angular. URL: https://angular.io/guide/what-is-angular (cit. a p. 17).