Università degli studi di Modena e Reggio Emilia

Dipartimento di Ingegneria "Enzo Ferrari"

Corso di Laurea in Ingegneria Informatica - Sede di Mantova

Titolo: prima riga
Seconda riga
Terza riga
Quarta riga

Relatore:
Prof. Luca Ferretti

Candidato:
Giulio Barabino

Correlatore:
Ing. Federico Magnanini

Anno Accademico 2021/2022

Indice

Capitolo 1

Introduzione

In questo capitolo si propongono degli esempi per gli oggetti utilizzati più di frequente in latex: la Sezione 1.1 descrive come scrivere citazioni, la Sezione 1.2 propone degli esempi di oggetti float, la Sezione 1.3 descrive come compilare questo documento.

1.1 Citazioni

Inserisco qualche citazione per mostrare la bibliografia. Per gli articoli accademici è quasi sempre possibile reperire i blocchi da inserire nel file bib da scholar [2], come ad esempio [1]. Scholar in questo caso è una risorsa/sito online e per questo. Precediamo le citazione da uno spazio indivisibile tramite il carattere ~.

1.2 Oggetti float

Nella Sezione 1.2.1 si propone un esempio di figura float, mentre nella Sezione 1.2.2 si propone un esempio di tabella float.

1.2.1 Figure

La Figura 3.1 è un esempio di figura float.

Figura 1.1: Esempio di figura float in latex.

1.2.2 Tabelle

La Tabella 1.1 è un esempio di tabella.

allineamento centrale	allineamento a sinistra	allineamento a destra
centrale	sinistra	destra

Tabella 1.1: Esempio di tabella float in latex.

1.3 Compilazione

Di seguito il codice da utilizzare per generare il pdf:

```
$ pdflatex main.tex
$ bibtex main.aux
$ pdflatex main.tex
$ pdflatex main.tex
```

Capitolo 2

Conoscenze di Base

In questa sezione verranno introdotti i concetti necessari per proseguire con la lettura dei capitoli successivi.

2.1 Passwordless

L'autenticazione passwordless è un metodo di autenticazione che permette ad un utente di effettuare il login ad un servizio senza la necessità di conoscere una password o più genericamente una conoscenza considerata segreta. Tipicamente utilizza una coppia di chiavi crittografiche, una privata e una pubblica: la prima viene generata e immagazzinata sul dispositivo dell'utente mentre la seconda viene inviata al server per verificare l'autenticità dei messaggi inviati dall'utente. La chiave privata, o segreta, non lascia mai il dispositivo su cui è stata creata e per accedervi è necessaria una qualche sorta di autenticazione da parte dell'utente. L'autenticazione può riguardare:

- Qualcosa che l'utente **possiede** come un telefono cellulare, un token OTP, un autenticatore hardware
- Qualcosa che l'utente **è** come l'impronta digitale, la scansione retinica o il riconoscimento vocale [3]

La registrazione passwordless e, conseguentemente, l'autenticazione vengono svolte seguendo un meccanismo *challenge-response*: al pervenire di una richiesta di regis-

trazione il server invia una cosidetta *challenge*. L'utente che ha iniziato l'operazione avrà il compito di apporre, tramite propria chiave privata, una firma crittografica sulla challenge e di fornire in risposta al server la challenge firmata accompagnata dalla chiave pubblica. Così facendo il server verificherà l'autenticità della firma tramite la chiave appena ricevuta e in caso di esito positivo, la immagazzinerà. La fase di autenticazione verrà svolta in modo analogo con la differenza che il server è già in possesso della chiave pubblica e non sarà necessario inviarla.

Come si può vedere non viene scambiato alcun segreto e l'unica interazione richiesta all'utente è quella in fase di firma della challenge. Anche allora l'utilizzatore non dovrà inserire codici o password ma semplicemente autenticarsi al dispositivo contenente la chiave privata tramite uno dei fattori sopra elencati. Sfruttando l'autenticazione passwordless è possibile sopperire alle criticità tipiche dei segreti a bassa entropia come le password, quali phishing, brute forcing etc.

2.2 FIDO

FIDO Alliance è un'associazione nata nel 2013 con lo scopo di migliorare i sistemi di autenticazione. Sono gli autori di *FIDO*, un set di specifiche che racchiude gli standard CTAP e WebAuthn. Nel corso degli anni vi sono stati un susseguirsi di iterazioni degli standard, prima conosciuta come Universal Authentication Factor per poi diventare Universal 2nd Factor e poi giungere alla versione corrente FIDO 2.0. I due protocolli CTAP e WebAuthn si occupano, rispettivamente, di definire le specifiche tramite cui vengono programmati gli autenticatori crittografici per fare in modo che possano operare con un client e di definire le specifiche tramite cui standardizzare l'autenticazione a servizi web sfruttando le chiavi crittografiche.

Capitolo 3

Modellazione del sistema

In questo capitolo verranno inizialmente descritti gli attori che concorrono alle fasi di registrazione/autenticazione per poi definire il flusso delle operazioni stesse.

3.1 Attori

Lo schema di seguito rappresenta lo stato attuale dello standard FIDO in accordo all'implementazione descritta successivamente.

Figura 3.1: PLACEHOLDER.

3.1.1 Service Provider

Il Service Provider è un fornitore di un generico servizio a cui l'utente è interessato ad accedere. Può essere un qualunque servizio di streaming, banking, shopping etc. il quale fa uso di un intermediario per l'autenticazione dei propri utenti. Questo può avvenire per varie ragioni sia economiche che legate alla sicurezza.

Il Service Provider fa uso del **security level** per indicare il livello di *survivability* desiderato: esso può crescere all'aumentare della confidenzialità del servizio erogato.

3.1.2 Identity Provider

L'Identity Provider è un ente terzo che si occupa di fornire a un Service Provider il servizio di autenticazione. Fa parte del cosiddetto *Relying Party*, cioè quella parte che fornisce un accesso sicuro ad un servizio.

Nella variante *survivable*, invece che fornire un Identity Server unico con il quale il FIDO Client comunica, ne fornisce un numero **n** desiderato dal Service Provider, a seconda della sicurezza richiesta dal tal servizio. In questo modo durante la fase di creazione l'operazione dovrà essere replicata dal client su tutti gli n Identity Server. La parte di autenticazione invece verrà svolta solo su un sottoinsieme $\mathbf{k} \leq \mathbf{n}$ di questi. Compito dell'Identity Provider è anche quello di fornire il token di autenticazione al client da presentare al Service Provider per fare in modo che l'utente possa accedere al servizio scelto.

I vari Identity Server comunicano con il FIDO Client, tramite User Agent, per creare le credenziali degli utenti mantenendo in memoria le challenge generate e gli identificatori degli utenti registrati con le relative chiavi pubbliche ricevute.

Compito dell'Identity Server è anche quello di rilevare eventuali tentativi di duplicazione dell'autenticatore fisico. Per fare ciò lo standard FIDO prevede un contatore gestito dall'autenticatore, sia esso globale o multiplo, che viene aggiornato ad ogni operazione avvenuta con successo. Il contatore prende il nome di *signature counter*. Il server mantiene in memoria l'ultimo valore ricevuto e, all'interazione successiva, controlla che non vi siano discrepanze. In particolare se il valore ricevuto è minore di quello salvato in memoria dal server allora si può essere in presenza di un tentativo di clonazione. (L'accadere dell'inverso non è detto che costituisca un problema, motivo per il quale può essere utilizzato anche un contatore globale.) Ciò significa che un attaccante maligno ha duplicato l'autenticatore ad uno stato precedente mentre l'utilizzatore ha continuato ad autenticarsi incrementando il contatore.

Nella variante survivable è stato modellato il sistema adottando un contatore specifico per ogni livello di sicurezza. Ad ogni autenticazione, e quindi ad ogni livello di sicurezza desiderato specificato dal Service Provider, verrà incrementato il contatore corrispondente a k. Questa modifica non altera il funzionamento del server, che

riceve sempre un valore di contatore unico, indi per cui non è stato necessario operare modifiche in questo senso dal lato del server.

3.1.3 Autenticatore

L'autenticatore hardware è un dispositivo che, tramite l'interazione fisica con l'utente, permette l'accesso al servizio web richiesto. Durante la fase di creazione delle credenziali, l'autenticatore si occupa di creare una coppia di chiavi crittografiche, una pubblica e una privata. Utilizzerà la chiave privata per firmare le challenge che gli vengono sottoposte mentre la pubblica verrà inviata al server così che esso possa verificare l'autenticità delle firme.

Per la fase di creazione e le successive di autenticazione viene richiesto all'utente di compiere un'azione: essa può essere la pressione di un pulsante sulla chiavetta stessa, un collegamento NFC oppure ancora l'identificazione tramite impronta digitale. Così facendo l'operazione in corso viene autenticata.

Nella variante *survivable* viene inviato il digest ottenuta dall'hashing delle n challenge dei server che partecipano all'operazione. Esso appare all'autenticatore come una challenge unica, motivo per il quale non è stato necessario modificare il codice dell'autenticatore per richiedere l'interazione ad ogni singola challenge.

Durante la fase di registrazione viene inizializzato un contatore tramite cui vengono tenute traccia delle operazioni conseguite correttamente. Ad ogni operazione il contatore, al corrispondente livello di sicurezza, viene incrementato mono-atomicamente. Come detto in precedenza, si è reso necessario implementare un contatore specifico per ogni livello di sicurezza richiesto, sia in fase di autenticazione che di creazione. Prima di fare ciò, è stato però necessario introdurre un contatore per ogni credenziale creata, poiché allo stato attuale il codice della Solokey, per questioni di spazio, prevede un contatore globale unico.

3.1.4 Client

Un dispositivo client che sfrutta un User Agent conforme ad implmentare le specifiche FIDO per il dialogo con il Relying Party e con l'autenticatore, in collaborazione con il

dispositivo hardware sottostante su è installato l'User Agent, tipicamente un sistema operativo. Il client è quindi interposto tra il FIDO Server e l'autenticatore fisico, agendo da intermediario. Il suo compito è duplice:

- Comunicare con il server al fine di iniziare, e successivamente terminare, le operazioni di autenticazione e di creazione delle credenziali
- Comunicare con l'autenticatore allo scopo di creare le chiavi crittografiche e firmare le challenge ricevute dal server

La comunicazione è bidirezionale, duplice e segue i costrutti specificati dalle API definite nei relativi standard: CTAP per l'interazione con l'autenticatore e WebAuthn per la comunicazione con il FIDO Server.

Nel caso particolare dell'estensione survivable il client si occupa di replicare le operazioni su n FIDO Server distinti, computando l'hash delle challenge ricevute e fornendolo all'autenticatore come un digest unico su cui apportare la firma.

3.2 Flusso operativo

Il flusso operativo si compone di due operazioni distinte: una fase di creazione delle credenziali e una fase di autenticazione dell'utente. Tali operazioni vengono svolte, rispettivamente, durante la registrazione al servizio del Service Provider e a tutti i login successivi.

3.2.1 Fase di registrazione

La fase di registrazione si origina a partire dalla richiesta dell'utente, utilizzando un User Agent, di registrarsi ad un servizio, offerto da un Service Provider, che supporti l'autenticazione passworldess, tramite degli Identity Provider. Il Service Provider fornisce all'utente il livello di sicurezza n necessario per completare l'operazione. Il processo messo in atto è il seguente:

1. Ogni Identity Server crea il proprio stato interno e la challenge

2. Ogni Identity Server invia al Client una serie di requisiti secondo cui deve essere svolta la cerimonia di registrazione, e la challenge generata

3. Il Client salva tutte le challenge ricevute in un vettore e computa l'hash dello stesso

4. Il Client effettua una chiamata al metodo opportuno dell'autenticatore, fornendo i requisiti di creazione richiesti dal Relying Party e il digest computato come challenge

5. L'autenticatore procede a generare la coppia di chiavi crittografiche seguendo le imposizioni del server e invia al client la challenge firmata accompagnata dalla chiave pubblica e dal signature counter

6. Il Client invia ad ogni Identity Server il vettore con le challenge, l'hash dello stesso, la firma. la chiave pubblica e il signature counter ricevuto

7. Ogni Identity Server controlla che la challenge da lui generata sia presente all'interno del vettore e controlla, computando lui stesso l'hash del vettore, l'integrità di quanto ricevuto. Infine, verifica tramite la chiave pubblica fornitagli l'autenticità della firma.

8. Qualora il processo sia andato a buon fine, gli Identity Server salveranno le informazioni ricevute (contatore, chiave pubblica, identificatore del client) al proprio interno e forniranno al Client l'attestazione con cui poter completare la registrazione presso il Service Provider

3.2.2 Fase di Autenticazione

La fase di autenticazione ricalca i passaggi di quella di registrazione con la differenza che gli Identity Server sono già in possesso della chiave pubblica con cui verificare l'autenticità della firma apportata alle challenge.

In questa fase viene comunicato un security level pari a k da parte del Service Provider, cioè il numero di Identity Server presso cui è necessario autenticarsi. Questo valore prende in considerazione la tollerabilità alle intrusione che ha il Service Provider.

In particolare: k può essere *(2j+1)*, dove j rappresenta il numero di Identity Server di cui si può tollerare la compromissione, oppure può essere *(2j+2)* in caso di requisiti particolarmente stringenti.

1. Ogni Identity Server crea il proprio stato interno e la challenge
2. Ogni Identity Server invia al Client una serie di requisiti, secondo cui deve essere svolta la cerimonia di autenticazione, e la challenge generata
3. Il Client salva tutte le challenge ricevute in un vettore e computa l'hash dello stesso
4. Il Client effettua una chiamata al metodo opportuno dell'autenticatore, fornendo i requisiti di autenticazione richiesti dal Relying Party e il digest computato come challenge
5. L'autenticatore invia al Client la challenge firmata e il contatore, specifico del livello di sicurezza stabilito, aggiornato
6. Il Client invia ad ogni Identity Server il vettore con le challenge, l'hash dello stesso, la firma ricevuta e il signature counter
7. Ogni Identity Server controlla che la challenge da lui generata sia presente all'interno del vettore e controlla, computando lui stesso l'hash del vettore, l'integrità di quanto ricevuto. Infine, verifica tramite la chiave pubblica, memorizzata precedentemente, l'autenticità della firma
8. Ogni Identiy Server controlla che il *signature counter* ricevuto sia maggiore di quello memorizzato in precedenza e in tal caso aggiorna quest'ultimo
9. Se i passaggi precedenti sono avvenuti con successo rilasciano al Client l'attestazione tramite cui può completare l'autenticazione presso il Service Provider

Bibliografia

[1] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of cryptology*, 1(2):77–94, 1988.

[2] Google. Google scholar. `https://scholar.google.it/`, visited in Sep. 2016.

[3] Wikipedia contributors. Passwordless authentication — Wikipedia, the free encyclopedia, 2022. [Online; accessed 15-September-2022].