UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di ingegneria dell'informazione ed elettrica e matematica applicata

Corso di laurea magistrale in ingegneria informatica

Project Work Algoritmi e Protocolli per la Sicurezza

GRUPPO 007

MISSION APS

Studente	Matricola	WP

Anno Accademico 2023/2024

SOMMARIO

1. WP1: MODELLO

1.1. DESCRIZIONE DEL PROBLEMA

Il progetto mira a creare un ambiente sicuro e protetto per un sito di aste online, sfruttando le tecnologie avanzate di identificazione, fornite dalla Carta d'Identità Elettronica (CIE) italiana. La CIE è equipaggiata con un chip elettronico che supporta la generazione di firme digitali, essenziale per validare l'identità degli utenti in modo sicuro ed efficace. L'utilizzo di questo strumento permette di assicurare che solo utenti verificati e in possesso dei requisiti necessari possano accedere alle funzionalità del sito, come la partecipazione alle aste, garantendo così l'integrità e la trasparenza delle transazioni.

L'approccio adottato prevede che gli utenti, attraverso un'interfaccia facile da usare, si autentichino con la loro CIE prima di accedere alle aree riservate del sito. Durante il processo di autenticazione, se l'utente supera la verifica, il server delle credenziali gli fornisce le informazioni necessarie per l'utilizzo dei servizi del server delle aste. Se l'autenticazione non riesce, il server delle credenziali nega l'accesso. Se invece l'autenticazione ha successo, il server invia le credenziali appropriate all'utente, che saranno poi confermate dal server delle aste.

Attraverso questo meccanismo, il sito d'aste può confermare l'identità di ciascun partecipante con certezza e precisione, riducendo notevolmente il rischio di frodi o accessi non autorizzati.

PREMESSE
Nel nostro esempio specifico, abbiamo selezionato come caso di studio un server di aste online che richiede, come credenziale di accesso, l'età dell'utente, e quindi la sua data di nascita. Da qui in avanti, il termine "server di interesse" sarà sinonimo di "server d'aste", e la "credenziale richiesta" si riferirà alla data di nascita dell'utente. È inoltre possibile che, in questo contesto, l'utente venga identificato come "acquirente". È importante sottolineare che l'analisi e le soluzioni proposte in questa relazione sono generalizzabili e applicabili a qualsiasi tipo di server che richieda qualsiasi tipo di credenziale.

1.2. PROCESSO PER ACCEDERE AL SITO D'ASTE

Quando l'utente vuole accedere al sito, deve seguire i seguenti passi:

1. **Collegamento al Server d'asta**: l'utente inizia il processo di accesso connettendosi al server d'asta, quest'ultimo reindirizza l'utente al server delle credenziali.
2. **Richiesta delle credenziali**: l'utente, quindi, richiede le sue credenziali al server delle credenziali che, di rimando, gli fornisce il modulo di richiesta delle credenziali da firmare.
3. **Firma e verifica**: l'utente utilizza il PIN e la propria carta d'identità elettronica (CIE) per firmare la richiesta. Questa firma viene inviata al server delle credenziali che verifica la validità di quest'ultima.
4. **Invio delle credenziali associate**: se la firma è valida, il server delle credenziali manda tutte le credenziali dell'utente a quest'ultimo.

5. **Estrazione delle credenziali di interesse**: l'utente controlla tutte le credenziali di suo interesse, comunicategli nel punto uno dal server d'aste, e le manda a quest'ultimo.
6. **Accesso autorizzato**: con la chiave di accesso ricevuta, l'utente finalmente accede al servizio.

1.3. ATTORI DEL SISTEMA

- **Utente** (acquirente, client);
- **Server d'asta** (server);
- **Server delle credenziali** (server);
- **IPZS e Ministero dell'Interno** (autorità competente e fidata);
- **Autorità giudiziaria** (autorità fidata);
- **Sviluppatori**.

1.3.1. Obiettivi degli attori

Utente:
Gli utenti possono accedere al servizio utilizzando le loro credenziali, ottenute attraverso l'autenticazione con Carta d'Identità Elettronica (CIE). Questo processo coinvolge l'uso del PIN associato alla propria CIE per confermare l'identità.

Server d'asta:
Il server d'asta fornisce un servizio online per l'organizzazione di aste. Gli utenti possono accedervi autenticandosi tramite credenziali apposite.

Server delle credenziali:
Il server delle credenziali fornisce le credenziali di accesso all'utente una volta autenticato tramite CIE.

IPZS e Ministero dell'Interno:
L'Istituto Poligrafico e Zecca dello Stato (IPZS) e il Ministero dell'Interno collaborano strettamente nella gestione delle Carte d'Identità Elettroniche (CIE).
L'IPZS, responsabile della produzione e della distribuzione fisica delle CIE, garantisce che ogni carta rispetti gli elevati standard di sicurezza e qualità, integrando tecnologie avanzate per prevenire frodi e falsificazioni. Questo include la distribuzione dei relativi PIN, essenziali per l'utilizzo sicuro delle carte.
Il Ministero dell'Interno stabilisce le linee guida per una corretta implementazione delle politiche di sicurezza in collaborazione con l'IPZS, garantendo un sistema di identificazione elettronica affidabile e sicuro per tutti i cittadini.

Autorità giudiziaria:
L'autorità giudiziaria è incaricata di perseguire eventuali reati commessi da individui malintenzionati coinvolti nel sistema.

Sviluppatori:
Gli sviluppatori sono responsabili della creazione e del mantenimento del sistema, con particolare attenzione alla sicurezza. Devono assicurarsi che il sistema sia resistente agli attacchi e preservi l'integrità dei dati e delle transazioni.

Note: Da traccia, il Ministero dell'Interno e l'IPZS sono entità sicure e fidate.

1.4. THREAT MODEL

Sniffy McSnifferson:

- **Tipologia:** Passivo
- **Descrizione:** Avversario interessato ad ottenere tutte le informazioni che passano sul canale (Dati personali, etc.) e non ha alcun interesse nel compromettere il funzionamento del sistema.
- **Risorse:** Risorse moderate, in quanto gli interessa solo l'accesso ai canali di comunicazione.

Buttafuori:

- **Tipologia:** Attivo
- **Descrizione:** Avversario interessato a manipolare i dati che passano sul canale in modo da negare agli utenti l'accesso al servizio (ad esempio modificare la firma o i requisiti posseduti dall'utente per accedere al servizio).
- **Risorse:** Risorse computazionali elevate, in quanto deve essere in grado di leggere, decriptare e modificare in maniera valida informazioni sensibili che passano sul canale.

Mariuolo:

- **Tipologia:** Attivo
- **Descrizione:** Avversario che ruba fisicamente la carta di identità ad un utente, cercando di spacciarsi per un'altra persona.
- **Risorse:** Risorse computazionali nulle, in quanto ha a che fare solo con l'utente fisico.

Sviluppatori:

- **Tipologia:** Attivo
- **Descrizione:** Avversari corrotti direttamente coinvolti nell'implementazione del servizio, che durante lo sviluppo inseriscono una backdoor.
- **Risorse:** Risorse computazionali discrete, in quanto conoscono la backdoor inserita e le debolezze del sistema.

Waldo (l'Intruso):

- **Tipologia:** Attivo
- **Descrizione:** Dipendenti o collaboratori che hanno accesso legittimo ai sistemi possono abusare delle loro autorizzazioni per accedere a informazioni sensibili o manipolare il processo di autenticazione.
- **Risorse:** Hanno accesso direttamente ai dati del server.

Mario in the middle:

- **Tipologia:** Attivo
- **Descrizione:** Avversario interessato ad ottenere tutte le informazioni che passano sul canale con l'intento di accedere al servizio anche se non possiede effettivamente i requisiti di accesso (ad esempio rubare la chiave di accesso o la firma, etc.). È interessato a fare un attacco di impersonificazione.
- **Risorse:** Risorse computazionali elevate, in quanto deve leggere e cercare di capire come utilizzare ciò che passa sul canale.

Bonnie & Clyde:

- **Tipologia:** Attivo
- **Descrizione:** Avversari che cercano di combinare le loro credenziali per accedere al servizio.
- **Risorse:** Risorse computazionali moderate, in quanto devono capire come unire le loro credenziali.

Pescatore:

- **Tipologia:** Attivo
- **Descrizione:** Avversario che tenta di ottenere le credenziali dell'utente attraverso tecniche di phishing, creando siti web falsi che imitano il sito d'aste legittimo/il server delle credenziali.
- **Risorse:** Risorse moderate, in quanto l'attaccante deve creare siti web simili a quelli legittimi e indurre gli utenti a visitarli.

Doszilla:

- **Tipologia:** Attivo
- **Descrizione:** Avversario che effettua attacchi di tipo Denial of Service (DoS) mirati a sovraccaricare i server, rendendoli inaccessibili agli utenti. L'obiettivo è interrompere il servizio bloccando o rallentando la rete attraverso richieste massive e simultanee.
- **Risorse:** Elevate, considerando che l'attaccante deve coordinare un gran numero di dispositivi o sistemi per generare traffico sufficiente a saturare la banda o le risorse del server.

Bugiardo ("Li faccio a Dicembre"):

- **Tipologia:** Attivo
- **Descrizione:** Avversario che cerca di modificare le proprie credenziali per accedere impropriamente a servizi da cui sarebbe normalmente escluso.
- **Risorse:** Moderatamente alte, poiché l'attaccante deve avere la capacità tecnica di alterare o falsificare documenti e/o dati digitali in modo convincente per eludere i controlli di sicurezza.

1.5. COMPLETENESS

Siano $u_1, \ldots, u_n$ gli utenti che vogliono accedere al sito d'aste.

Al tempo t_0 l'utente u_i *(i = 1, ..., n)* richiede l'accesso al servizio.

Al tempo t_1 il server S_1 reindirizza l'utente u_i *(i = 1, ..., n)* verso il server delle credenziali S_C.

Al tempo t_2 l'utente u_i *(i = 1, ..., n)* si genera una nuova coppia chiave privata e chiave pubblica ($S_{ka'}$, $P_{ka'}$) e manda la chiave pubblica al server delle credenziali S_C.

Al tempo t_3 il server delle credenziali S_C verifica la veridicità della chiave.

Al tempo t_4, l'utente u_i *(i = 1, ..., n)* richiede le credenziali per l'accesso.

Al tempo t_5, se la verifica precedente è andata a buon fine, il server delle credenziali S_C manda il modulo di richiesta delle credenziali *m* da firmare all'utente u_i *(i = 1, ..., n)* in modo da verificarne l'identità.

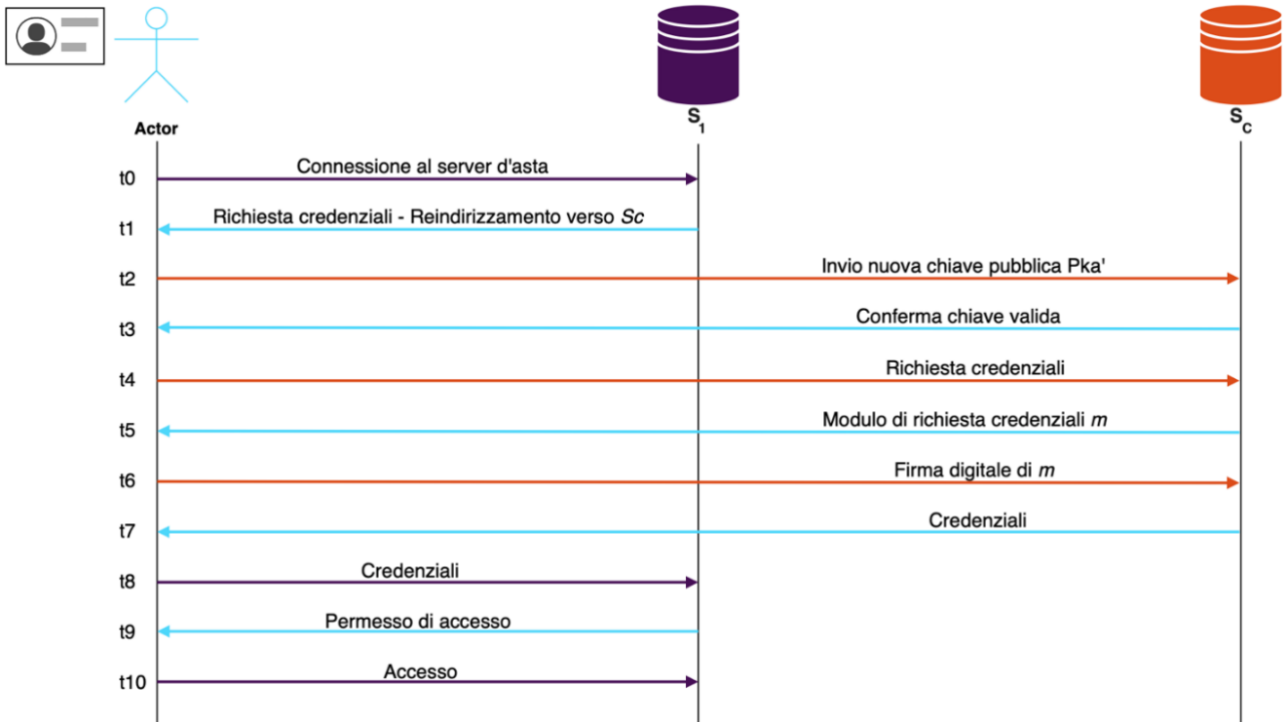
Al tempo t_6 l'utente u_i *(i = 1, ..., n)* firma *m* utilizzando il PIN della propria carta di identità.

Al tempo t_7 il server delle credenziali S_C verifica se la firma è valida, in caso affermativo manda tutte le credenziali associate alla CIE all'utente.

Al tempo t_8 l'utente valuta l'integrità delle credenziali ricevute e le manda al server d'asta S_1.

Al tempo t_9 il server delle credenziali valuta la credenziale per vedere se rispetta i requisiti imposti dal suo servizio.

Al tempo t_{10}, se l'utente possiede le credenziali corrette per usufruire del servizio, il server d'asta S_1 permette l'accesso all'utente.



1.6. FUNZIONALITÀ

1.6.1. Autenticazione tramite CIE

Il sistema utilizza la Carta d'Identità Elettronica (CIE), che è dotata di un chip elettronico avanzato, per garantire un'efficace autenticazione degli utenti. Per accedere al servizio, ogni utente deve compiere il passo cruciale di firmare digitalmente un certificato. Questo processo implica l'utilizzo del PIN personale associato alla propria carta d'identità. Il PIN funge da chiave per attivare il chip della CIE, permettendo così la generazione di una firma digitale sul modulo di richiesta delle credenziali che certifica l'autenticità dell'utente. Questa firma è essenziale per verificare non solo l'identità ma anche l'integrità dei dati inviati durante il processo di autenticazione.

1.6.2. Ottenimento delle credenziali

Il server delle credenziali (S_C) riceve dall'utente la seconda chiave pubblica generata e ne controlla la veridicità. In seguito, controlla la firma digitale inviata dall'utente per verificare che provenga da una CIE valida e che i dati non siano stati alterati durante il trasferimento. Una volta confermata l'autenticità della firma, il server manda tutte le credenziali associate alla CIE all'utente, appositamente firmate.

1.6.3. Utilizzo delle credenziali e controllo requisiti richiesti

Quest'ultimo, una volta ricevute le credenziali, le manda al server d'asta, che si occupa di verificare la coerenza con i suoi criteri di accesso. Se tale procedura va a buon fine, all'utente viene permesso l'accesso al server d'asta.

1.7. PROPRIETÀ

1.7.1. Autenticazione tramite CIE

Integrità

La funzionalità di accesso tramite Carta d'Identità Elettronica (CIE) è progettata per assicurare l'integrità, garantendo che i dati relativi all'utente e al processo di accesso rimangano inalterati e non manipolati durante le fasi di autenticazione e autorizzazione. L'elemento chiave da considerare è:

1. **Integrità dei dati:** durante la trasmissione dei dati dall'utente ai vari server, è fondamentale che questi non subiscano alterazioni, inclusi dati chiave come il modulo di richiesta e la firma digitale. Questo previene qualsiasi modifica non autorizzata ai dati, mantenendo la sicurezza dell'identità digitale dell'utente. La probabilità di violazione dell'integrità è da considerare trascurabile.

Confidenzialità

La funzionalità di accesso tramite CIE mira anche a proteggere la confidenzialità, assicurando che le informazioni personali e di autenticazione degli utenti siano accessibili solo agli enti autorizzati. L'aspetto specifico da considerare è:

1. **Protezione dei dati personali**: è essenziale che solo gli enti autorizzati possano accedere alle informazioni personali degli utenti, inclusi dati sensibili e l'identità stessa

dell'utente. La probabilità di violazioni della confidenzialità è considerata trascurabile, garantendo che non vi sia divulgazione non autorizzata di tali informazioni.

1.7.2. Ottenimento delle credenziali

Integrità

La funzionalità di ottenimento delle credenziali è essenziale per garantire che le credenziali fornite rimangano inalterate e provenienti da fonti affidabili. Elemento chiave da considerare è:

1. **Integrità della firma**: è cruciale verificare l'autenticità delle credenziali restituite all'utente. La probabilità di violazione dell'integrità è da considerare trascurabile.

Confidenzialità

La protezione della confidenzialità durante l'ottenimento delle credenziali è prioritaria, per evitare che le credenziali siano accessibili a entità non autorizzate. Aspetti specifici da considerare includono:

1. **Cifratura delle credenziali**: le credenziali di accesso devono essere cifrate prima della loro trasmissione all'utente, per prevenirne l'intercettazione e l'utilizzo non autorizzato.
2. **Canali di comunicazione sicuri:** è fondamentale utilizzare canali di comunicazione sicuri per la trasmissione di credenziali e altre informazioni sensibili, assicurando che queste siano protette durante il transito da eventuali attacchi o intercettazioni.

La probabilità di violazioni della confidenzialità è considerata trascurabile.

1.7.3. Utilizzo delle credenziali e controllo requisiti richiesti

Integrità

Questa fase è di cruciale importanza perché rappresenta il momento in cui l'utente, correttamente in possesso dei requisiti necessari, può effettivamente beneficiare di tutti i servizi offerti. Gli aspetti cruciali sull'integrità dell'operazione da considerare includono:

1. **Integrità delle credenziali trasmesse dall'utente:** è fondamentale che le credenziali di accesso rimangano inalterate per prevenire sia l'accesso non autorizzato da parte di utenti non qualificati, sia eventuali interruzioni del servizio per utenti legittimi a cui le credenziali sono state modificate.
2. **Impossibilità di aggregazione di credenziali:** deve essere impedito che più utenti malintenzionati possano combinare le loro credenziali per ottenere accessi non autorizzati al servizio.

La probabilità di violazione dell'integrità è da considerare trascurabile.

Confidenzialità

È di massima importanza, in questa come nelle precedenti fasi, impedire l'accesso alle credenziali da parte di individui non autorizzati. Nello specifico, è particolarmente necessario:

1. **Cifratura delle credenziali**: Le credenziali devono essere adeguatamente protette per evitare furti di identità, impersonificazioni e conseguenti accessi non autorizzati.

La probabilità di violazioni della confidenzialità è considerata trascurabile.

2. WP2: SOLUZIONE

2.1. INTRODUZIONE E PARTITI COINVOLTI

In questo capitolo presenteremo nel dettaglio il nostro progetto di soluzione. Cominceremo identificando i vari soggetti coinvolti nel processo, per poi illustrare le modalità di interazione tra di loro e le presupposizioni su cui si basano queste dinamiche. Ci concentreremo in particolare su due interazioni fondamentali: la prima tra l'utente e il server delle credenziali, che ha lo scopo di ottenere le credenziali necessarie, e la seconda tra l'utente e il server target, dove verrà esplicato come le credenziali vengono utilizzare per l'accesso al servizio. Questa disamina ci permetterà di comprendere meglio i flussi di comunicazione e le relative necessità di sicurezza in entrambe le fasi del processo.

2.2. ASSUNZIONI

- Si assume che i dispositivi di chi è onesto non siano corrotti, quindi, che l'hardware non sia compromesso ed il software non sia controllato da malware di qualunque tipo.
- Per garantire che la CIE non è manomessa e perfettamente funzionante, si assume che tutte le parti coinvolte dalla creazione alla consegna del documento all'utente finale siano oneste.
- Si assume inoltre che l'utente onesto faccia un uso esclusivo del documento.
- Si assume che i server delle credenziali, essendo forniti di dati sensibili dei vari utenti, siano delle fonti fidate che rilasciano dei certificati digitali sulle credenziali fornite e che utilizzino questi dati nel modo appropriato e non ne facciano un uso improprio.
- Si assume che inizialmente l'utente possiede un certificato digitale che attesta la validità della CIE e il Server delle Credenziali è incluso tra le Autorità di Certificazione (CA) fidate dall'utente.
- Si assume che il certificato digitale incluso nella Carta d'Identità Elettronica (CIE) sia un certificato X509v3, rilasciato dall'Istituto Poligrafico e Zecca dello Stato (IPZS), che funge anche da autorità di certificazione.
- Si assume che è compito del Server delle Credenziali verificare le scadenze delle credenziali rilasciate, una volta che il periodo di validità è scaduto.
- Si assume che è compito delle autorità competenti segnalare ai server coinvolti le carte di identità rubate, dismesse per motivi giuridici o smarrite. Così come è compito dell'utente onesto denunciare eventuali furti del documento alle autorità competenti.

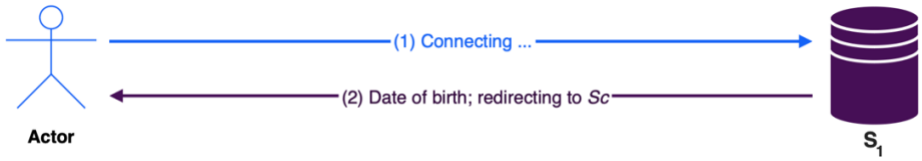
2.3. TRANSPORT LAYER SECURITY

Si è scelto di adottare il protocollo Transport Layer Security in versione 1.3 per tutte le comunicazioni tra le parti. Tale decisione è motivata dalla necessità di assicurare la confidenzialità e l'integrità delle comunicazioni tra i soggetti coinvolti. Inoltre, il protocollo

implica una fase di autenticazione dei server, effettuata mediante certificati o catene di certificati, essenziale per garantire che i server siano riconosciuti come affidabili dagli utenti.

Per i motivi sopra citati quindi si precisa che da questo punto in poi ogni comunicazione menzionata, se non specificato altrimenti, seguirà tale protocollo ereditandone i conseguenti benefici.

2.4. RICHIESTA CREDENZIALI



È necessario prevedere una fase in cui i potenziali utenti si rivolgono all'autorità che fornisca le credenziali (Server S_C), cioè un server CA (una CA nota e affidabile). Quando un utente viene reindirizzato al Server S_C, il client, dopo aver opportunamente verificato, grazie al protocollo TLS, l'identità del Server delle credenziali S_C, genera una nuova coppia di chiavi segreta e pubblica ($S_{ka'}$, $P_{ka'}$) ed invia la $P_{ka'}$ al server S_C.

Una volta ricevuta la chiave S_C, per confermare che l'utente U sia il legittimo proprietario della chiave inviata si esegue il protocollo *Zero-Knowledge* di *Schnorr*. L'obiettivo è dimostrare che solo il client conosce la chiave segreta ($S_{ka'}$) associata alla chiave pubblica ($P_{ka'}$). Il processo di autenticazione è il seguente:

- Sia (G, q, g, h) un'istanza del problema del logaritmo discreto, con $y = g^x$, dove $x \in z_q$.
- U invia $a = g^r$ a S_1, con r scelto casualmente in z_q.
- S_1 invia una challenge c scelta casualmente in z_q a U.
- U risponde inviando $z = r + c * x$ a S_1 per dimostrare la conoscenza di x.
- S_1 verifica se $g^z == a * Y^c$. Se la verifica ha successo, S_1 conferma che U conosce x senza ottenere ulteriori informazioni sulla chiave segreta.

Una volta concluso il protocollo, il server S_C invia un modulo di richiesta delle credenziali, denominato *m*, da firmare tramite la CIE. L'utente esegue l'algoritmo SHA-256 sul messaggio m, ottenendo così l'hash del messaggio, e applica la firma digitale su di esso eseguendo la funzione SIGN(PIN, HASH(m)). Il PIN è una stringa di 10 caratteri che solo il possessore di una specifica CIE dovrebbe conoscere.

La funzionalità di SIGN associata alla CIE verifica se il PIN inserito corrisponde a quello associato al documento dell'utente . Se la verifica ha esito positivo, viene applicata una firma ECDSA (Elliptic Curve Digital Signature Algorithm) sull'hash del messaggio, restituendo una coppia (firma, hash). In caso contrario, la firma viene negata.

A questo punto, il Server S_C deve verificare la validità della firma utilizzando la chiave pubblica (P_{ka}) contenuta nel certificato digitale rilasciato dalla CIE al momento della firma, che ne attesta pure l'autenticità. Questo processo avviene tramite la funzione verify(P_{ka}, HASH(m)), che restituisce true se la chiave pubblica corrisponde alla chiave privata utilizzata per firmare, altrimenti restituisce false e abortisce l'operazione.

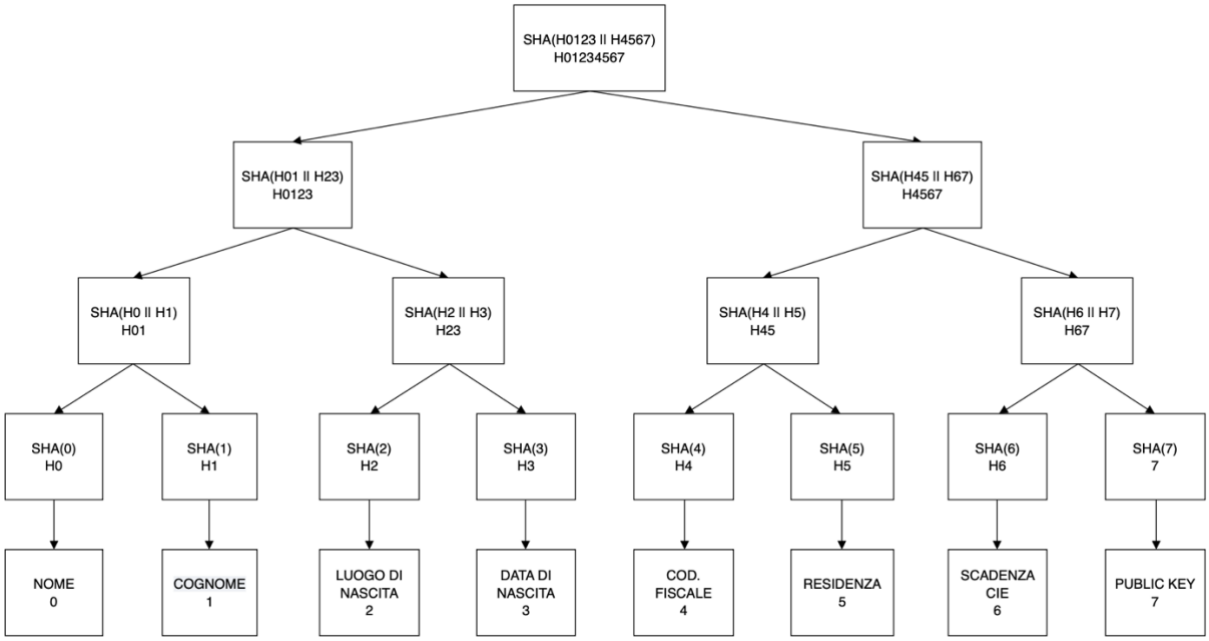
2.5. RILASCIO CREDENZIALI

Una volta verificata l'identità dell'utente e l'integrità della firma, il server S_C deve spedire le credenziali associate all'utente che ha effettuato la richiesta. Quest'ultime saranno praticamente le principali credenziali leggibili su una carta di identità, ovvero:

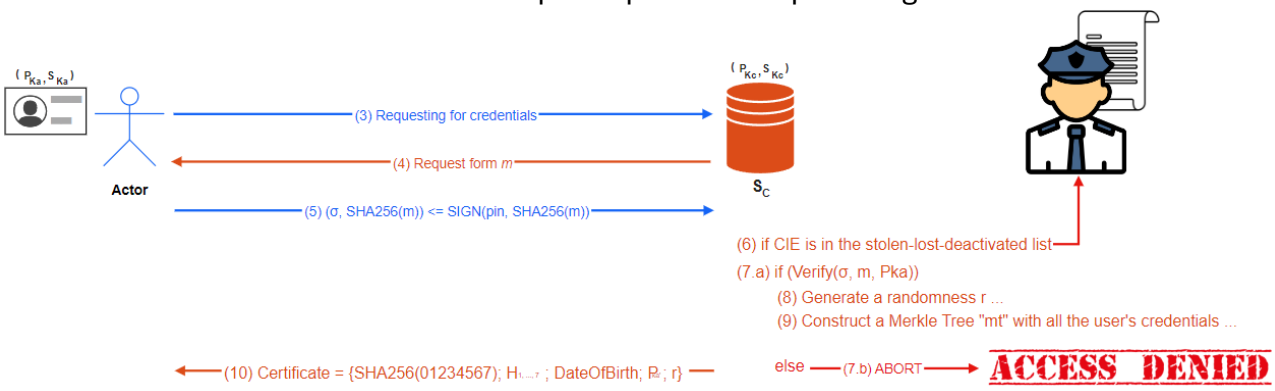
- Nome;
- Cognome;
- Luogo di nascita;
- Data di nascita;
- Codice fiscale nei formati alfanumerico;
- Indirizzo di residenza;
- Data di scadenza;
- $P_{ka'}$ generata dall'utente.

La procedura di rilascio avviene secondo i seguenti passaggi:

1. **Costruzione delle credenziali:** per garantire la tutela della privacy, le informazioni sensibili dell'utente vengono integrate all'interno di un certificato mediante l'utilizzo di un Merkle Tree. In questo approccio, ogni dato sensibile indicato precedentemente, concatenato con una stringa random r, è rappresentato da una foglia dell'albero. Il valore della radice dell'albero viene calcolato applicando l'algoritmo SHA-256 alle varie combinazioni di nodi fratelli di quest'ultimo.
   Per tutelare ulteriormente la privacy nel certificato non si includono direttamente tutti i dati sensibili ma solo il valore della radice del Merkle Tree. Questo sistema non solo salvaguarda la riservatezza delle informazioni ma permette anche di verificarne l'integrità, attraverso la convalida del valore della radice del Merkle Tree. Inoltre, si aggiunge la scadenza della carta d'identità, presa dal certificato digitale associato alla CIE, nel campo "not after" del certificato da mandare all'utente. Questa informazione è inserita in previsione di strategie legate alla scadenza delle credenziali, argomento che verrà trattato nel paragrafo successivo.



2. **Invio delle credenziali all'utente:** quando l'utente interagisce con il server delle credenziali, riceve un certificato che include il valore della radice del Merkle Tree e la scadenza del documento. Questo certificato contiene anche le credenziali necessarie per accedere al server di riferimento, che, nel nostro caso di studio, includono dati come la data di nascita. In aggiunta, sono forniti tutti gli hash intermedi del Merkle Tree, essenziali per consentire all'utente di verificare l'integrità delle credenziali ricevute, nonché la randomness utilizzata nel punto precedente per le foglie dell'albero.



2.6. GESTIONE DELLE SCADENZE DI VALIDITÀ DEI DOCUMENTI E FURTI

Come precedentemente menzionato, il certificato digitale associato alla CIE include la data di scadenza del documento. Questa data è inclusa nel certificato contenente la radice del Merkle Tree delle credenziali, che il server delle credenziali (S_C) invia all'utente. Il server target verifica i requisiti necessari per il servizio, mentre il server delle credenziali controlla la validità corrente delle credenziali. In caso di scadenza o invalidità delle credenziali, il server S_C è tenuto a negare l'accesso al servizio.
È importante inoltre precisare che il server S_C ha il compito di controllare se la CIE con la quale gli utenti firmano sia revocata:

- Periodicamente, l'IPZS rilascia una tabella delle CIE revocate a seguito di smarrimento o furto.
- Queste tabelle sono accessibili alle autorità responsabili del rilascio delle credenziali. Se durante la richiesta delle credenziali viene rilevato l'utilizzo di una CIE revocata, l'autorità ha la possibilità di segnalare l'incidente alle autorità giudiziarie.

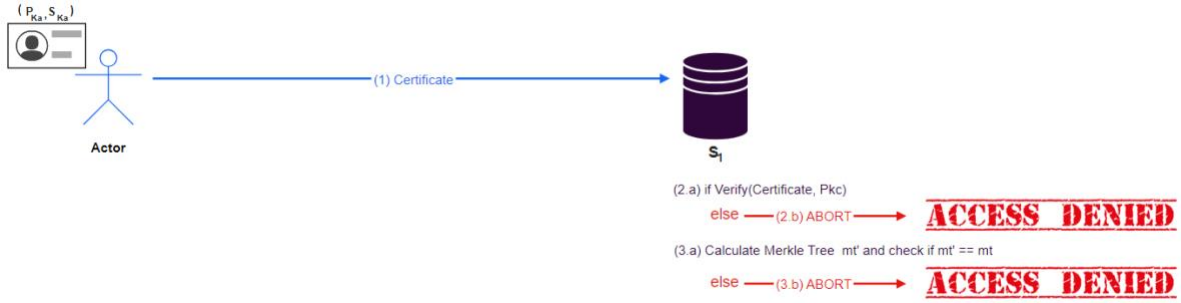
2.7. AUTENTICAZIONE AL SERVER D'ASTE

Per accedere al proprio account sul server delle aste, l'utente U stabilisce una connessione sicura tramite il protocollo TLS con il server S_1. Il processo di autenticazione si sviluppa attraverso diverse fasi cruciali:

1. **Invio del certificato digitale**: l'utente U invia al server S_1 un certificato digitale che contiene la root del Merkle Tree, le foglie delle credenziali rilevanti per S_1 (inclusa la chiave pubblica $P_{ka'}$ di U), gli hash intermedi e la randomness utilizzata nella sua creazione.

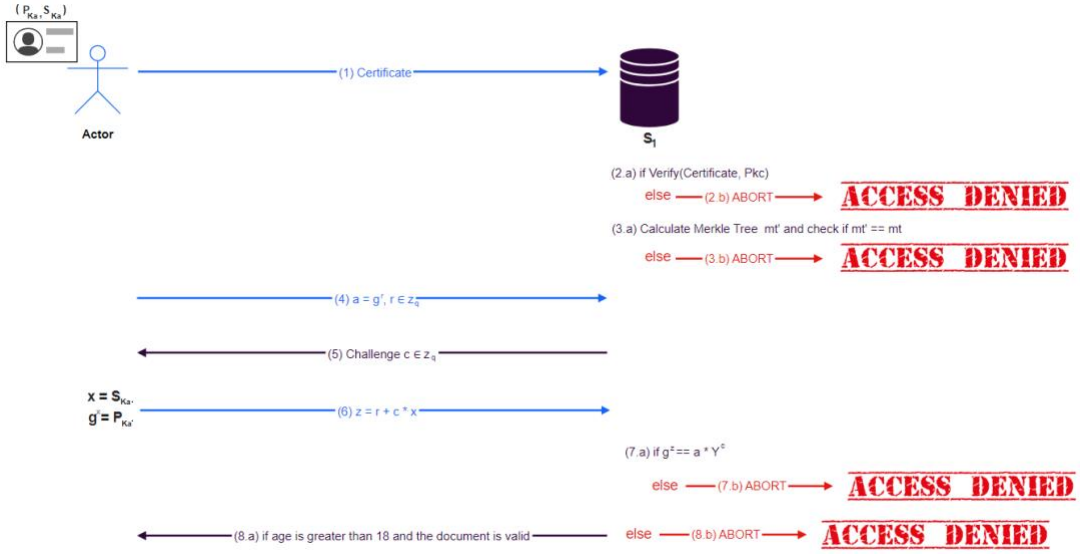
2. **Verifica del certificato**: il server S_1, una volta ricevuto il certificato, procede alla sua verifica per assicurarsi che provenga da un'autorità di certificazione (CA) riconosciuta e affidabile. Questo controllo include la verifica dell'autenticità del certificato tramite il confronto con i certificati delle autorità superiori, seguendo il percorso dei certificati fino alla root CA.

3. **Ricostruzione del Merkle Tree**: una volta convalidato il certificato, S_1 estrae i dati necessari dal certificato per ricostruire il Merkle Tree. Questa operazione permette di verificare l'integrità delle informazioni contenute nel certificato e di assicurarsi che rispettino tutte le policy di sicurezza stabilite. Se la root del Merkle Tree ricostruito corrisponde a quella indicata nel certificato, il processo di autenticazione può procedere. Se le due root non corrispondono, il processo viene interrotto e l'accesso negato per motivi di sicurezza.



4. **Verifica dell'identità dell'utente**: S_1 deve ora verificare che U sia il legittimo proprietario delle credenziali. Questo avviene tramite il protocollo *Zero-Knowledge* di *Schnorr*. L'obiettivo è dimostrare che solo il client conosce la chiave segreta ($S_{ka'}$) associata alla chiave pubblica ($P_{ka'}$) estrapolata dal Merkle Tree. Il processo di autenticazione è il seguente:
    - Sia (G, q, g, h) un'istanza del problema del logaritmo discreto, con $y = g^x$, dove $x \in z_q$.

- U invia $a = g^r$ a S_1, con r scelto casualmente in z_q.
- S_1 invia una challenge c scelta casualmente in z_q a U.
- U risponde inviando $z = r + c * x$ a S_1 per dimostrare la conoscenza di x.
- S_1 verifica se $g^z == a * Y^c$. Se la verifica ha successo, S_1 conferma che U conosce x senza ottenere ulteriori informazioni sulla chiave segreta.



5. **Controllo dei requisiti per l'accesso**: il server verifica che le credenziali presentate dall'utente soddisfino i requisiti necessari per accedere al servizio.

Alla fine del processo, il server S_1, a seguito di una richiesta di accesso riuscita, risponde con la pagina web dell'account personale dell'utente. Da qui, l'utente può partecipare alle aste e utilizzare le varie funzionalità del sito.
In un momento successivo, in cui l'utente vuole accedere nuovamente, dovrà rimandare a S_1 il Merkle Tree contenente le sue credenziali.

3. WP3: ANALISI DELLA SOLUZIONE

In questa sezione analizzeremo la nostra soluzione per verificarne la validità teorica e la resistenza contro eventuali attacchi perpetrati dagli avversari individuati nel threat model. Inizieremo descrivendo l'analisi della completezza del sistema proposto, assicurandoci che tutte le componenti siano correttamente implementate e funzionino come previsto. Successivamente, esamineremo in dettaglio le quattro proprietà fondamentali del sistema: confidenzialità, integrità, efficienza e trasparenza. Valuteremo ciascuna proprietà descrivendo possibili attacchi e analizzando come i nostri avversari potrebbero compromettere il sistema minando tale proprietà, determinando così il livello di sicurezza effettivo del sistema stesso.

3.1. ANALISI DELLA COMPLETEZZA

Uno studio della soluzione evidenzia che, in assenza di interferenze esterne, un utente onesto può accedere con successo. Pertanto, qualsiasi utente onesto, in possesso legittimo dei requisiti di accesso a S_1, riesce facilmente a richiedere le credenziali e ad accedere al server target. Al contrario, utenti malevoli non riescono ad ottenere l'accesso a meno di un attacco fisico diretto alla persona. Infatti, in caso di irregolarità da parte dell'utente, le credenziali non vengono correttamente rilasciate e l'accesso al servizio non viene consentito.

Si noti che l'intervento della giustizia è coinvolto solo quando ci sono problemi legati al mondo fisico (ad esempio, la consegna del bene/servizio o il furto della carta d'identità) e questo è inevitabile poiché il sistema digitalizzato non può prevenire le frodi nel mondo reale, a meno di eventuali notifiche da parte delle autorità al servizio. Nonostante i tentativi degli attori disonesti, il sistema di digitalizzazione, sotto opportune condizioni, fornisce prove evidenti di tali comportamenti.

3.2. ANALISI DELLA CONFIDENZIALITÁ

La confidenzialità, secondo lo standard ISO/IEC 27000:2018, è la proprietà delle informazioni di non essere rese disponibili o divulgate a individui, entità o processi non autorizzati. In un sistema che garantisce la confidenzialità, una terza parte che entri in possesso delle informazioni scambiate tra mittente e destinatario non è in grado di ricavarne alcun contenuto informativo intelligibile.

Nel nostro contesto specifico, la confidenzialità implica che solo l'utente possa accedere ai dati contenuti nella propria carta d'identità elettronica (CIE), e che al server siano trasmessi esclusivamente i dati strettamente necessari per l'accesso. Durante le comunicazioni tra client e server, le informazioni devono essere rese indecifrabili per eventuali utenti esterni in ascolto. Questo riguarda sia la fase iniziale, in cui il client richiede le credenziali al server S_C, sia la fase in cui il client comunica con il server d'asta (S_1).

3.2.1 Autenticazione tramite CIE

In questa prima fase è cruciale che i dati sensibili dell'utente come quelli contenuti nella CIE, le informazioni sensibili contenute nei moduli di richiesta o qualsiasi altra informazione

trasmessa, siano protetti contro possibili intercettazioni da parte di avversari. Anche se tali dati venissero intercettati, dovrebbero essere resi incomprensibili attraverso adeguate tecniche di cifratura.

Tra i principali attacchi alla confidenzialità dei dati trasmessi rientrano:

- **Intercettazione delle informazioni sensibili attraverso il canale di comunicazione:** Un avversario, come ***Sniffy McSnifferson***, potrebbe posizionarsi sui canali di comunicazione tra l'utente e i due server, intercettando eventuali informazioni sensibili appartenenti all'utente, come le credenziali rilasciate da S_C, o altre informazioni utili a beneficio proprio o di altri.

- **Intercettazione dei dati a seguito di un reindirizzamento malevolo:** Durante la fase in cui l'utente si connette al server d'asta S_1 o viene reindirizzato a S_C per ottenere le credenziali necessarie, la connessione o il reindirizzamento può essere vulnerabile a un attacco malevolo da parte di un avversario come il ***Pescatore***. L'avversario può intervenire in questo passaggio reindirizzando l'utente verso un server malevolo controllato dall'attaccante stesso. Con l'aiuto di un avversario come ***Mario in the middle***, può intercettare e modificare le richieste e le risposte tra l'utente e S_1/S_C, facendo sembrare che il suo server malevolo sia legittimo.

Per entrambi questi due tipi di attacchi il sistema è capace di proteggersi grazie all'utilizzo di ***TLS 1.3***. Per la prima situazione, TLS garantisce il criptaggio delle informazioni che passano sul canale, rendendo difficoltosa la comprensione delle informazioni intercettate. Inoltre, è efficace anche contro il secondo tipo di attacco, poiché il protocollo richiede al server di presentare un certificato digitale che l'utente verifica per autenticare l'identità del server. Questo certificato fornisce un metodo per verificare l'identità del server, prevenendo attacchi di impersonificazione. Anche nel remoto caso in cui il Pescatore riesca a presentare un certificato valido, non riuscirebbe comunque a interpretare i dati trasmessi, poiché non avrebbe accesso alla chiave di crittografia condivisa durante il key share, e quindi non potrebbe accedere alle informazioni sensibili di suo interesse.

3.2.2 Ottenimento delle credenziali

In questa fase, la confidenzialità è particolarmente cruciale perché, in sua assenza, soggetti malevoli potrebbero sfruttare la situazione per accedere a dati sensibili o, peggio ancora, riutilizzarli nel tentativo di commettere un furto di identità.

Tra i principali attacchi alla confidenzialità delle credenziali rientrano:

- **Intercettazione delle informazioni sensibili attraverso il canale di comunicazione:** Anche in questo contesto, un avversario come ***Sniffy McSnifferson*** potrebbe intercettare le credenziali trasmesse sul canale e utilizzarle in modo improprio, seguendo le modalità già descritte in precedenza.
  Specificamente, in questa fase, un attaccante come il ***Mario in the Middle*** potrebbe intercettare le informazioni sul canale, rubarle e riutilizzarle per accedere abusivamente al servizio in un secondo momento.

- **Attacco alla randomness:**
  Un attacco da parte degli ***Sviluppatori*** potrebbe compromettere il processo di generazione della casualità utilizzando numeri prevedibili, rendendo più facile eseguire un attacco di tipo *Brute Force* per scoprire i dati sensibili contenuti all'interno delle foglie del Merkle Tree.

Il primo tipo di attacco può essere facilmente sventato dal sistema grazie all'utilizzo di ***TLS 1.3*** che come specificato in precedenza in tale elaborato, garantisce la confidenzialità dei messaggi che passano sul canale. Dato che si è già provveduto ad esplicare i benefici del protocollo si evita di entrare in dettagli inutili.

D'altro canto, il secondo tipo di attacco non può essere direttamente contrastato dalla nostra soluzione, poiché dipende interamente dall'integrità morale ed etica dei progettisti coinvolti. Questi devono impegnarsi a utilizzare ***generatori di casualità robusti*** per prevenire il successo di attacchi di tipo brute force, indipendentemente dal fatto che l'inserimento di una *backdoor* sia intenzionale o meno. Tuttavia, essendo noti i progettisti del sistema, qualsiasi manomissione, come l'inserimento di una backdoor, porterebbe alla segnalazione del progettista disonesto alle autorità competenti.

3.2.3 Utilizzo delle credenziali e controllo requisiti richiesti

Come nella fase precedente, la confidenzialità delle credenziali trasmesse dall'utente al server è essenziale per prevenire accessi illeciti.

I principali attacchi che minacciano questa confidenzialità includono:

- **Intercettazione delle informazioni sensibili attraverso il canale di comunicazione:**
  Anche in questa fase, ***Sniffy McSnifferson*** potrebbe posizionarsi sul canale di comunicazione, intercettando le credenziali inviate dall'utente al server per ottenere l'accesso. Successivamente, potrebbe collaborare con altri malintenzionati, fornendo loro le informazioni necessarie per orchestrare ulteriori attacchi. Un esempio particolarmente rilevante potrebbe vedere Sniffy McSnifferson assistere un potenziale attaccante ***Buttafuori***, il quale potrebbe utilizzare le credenziali criptate ricevute per tentare di modificarle e impedire così l'accesso al servizio.

- **Intercettazione delle informazioni sensibili da parte di un server malevolo:**
  Un attaccante come il ***Pescatore*** potrebbe sfruttare una procedura di reindirizzamento malevolo, come descritto in precedenza, per ricevere le credenziali dell'utente fingendosi il server S_1. Questo potrebbe essere fatto a proprio vantaggio o per assistere altri, ad esempio per condurre attacchi di phishing o supportare complici come un ***Mario in the Middle***, mirando a interrompere la comunicazione con il vero server e facilitandogli il furto d'identità.

Il sistema è in grado di contrastare questi attacchi grazie ancora una volta all'impiego di ***TLS 1.3***, per le stesse ragioni dettagliate nel primo sottoparagrafo.

3.3. ANALISI DELL'INTEGRITÁ

L'integrità è la capacità di preservare la veridicità e la sicurezza dei dati e delle risorse, assicurando che questi non vengano alterati o eliminati se non da soggetti autorizzati. In un sistema che salvaguarda l'integrità, gli utenti non autorizzati non possono modificare i dati a loro discrezione per compiere azioni illecite o generare disservizi.

Nel nostro sistema, l'integrità si manifesta attraverso la garanzia che dati sensibili, chiavi di cifratura, certificati e credenziali trasmesse attraverso i canali non possano essere modificati. Questo impedisce la possibilità di eseguire attacchi attivi al sistema sfruttando tali modifiche per alterarne il comportamento o commettere atti illeciti.

3.3.1 Autenticazione tramite CIE

È essenziale che tutte le informazioni sensibili scambiate durante questa fase, così come eventuali firme o chiavi crittografiche, rimangano inalterate per prevenire attacchi attivi da parte di soggetti malintenzionati.

Tra i principali attacchi che in questa fase potrebbero abbattere tale integrità abbiamo:

- **Modifica dell'URL di reindirizzamento dell'utente:**
  Per fini di phishing e intercettazione dei dati, un avversario come il ***Pescatore*** potrebbe modificare l'URL utilizzato da S_1 per reindirizzare l'utente al Server delle credenziali. Questa alterazione avrebbe lo scopo di deviare l'utente verso un server malevolo controllato dall'avversario stesso per gli obiettivi definiti nella sezione sulla confidenzialità.

- **Modifica della firma:**
  Un attaccante come il ***Buttafuori*** potrebbe alterare la firma apposta su un documento digitale per invalidarla, impedendo così la verifica della firma tramite la chiave pubblica dell'utente e causando l'interruzione della comunicazione.

- **Manipolazione della chiave pubblica:**
  Durante la trasmissione della chiave pubblica, un attaccante come il ***Mario in the Middle*** potrebbe intercettare il messaggio contenente la nuova chiave pubblica generata dall'utente e sostituirla con una chiave di sua proprietà. Se riesce a convincere la parte ricevente che la chiave manipolata è autentica, può decifrare i messaggi cifrati destinati all'utente originale e cifrare messaggi con la sua chiave, ingannando ulteriormente la parte ricevente.

- **Saturazione delle richieste al server:**
  Un avversario come il ***Doszilla*** potrebbe lanciare un attacco *Denial of Service (DoS)* inondando il server S_1 o S_C con un volume eccessivo di richieste, rendendo il server inaccessibile agli altri utenti.

- **Furto di identità:**
  L'avversario conosciuto come ***Mariuolo*** potrebbe appropriarsi indebitamente di un documento d'identità per accedere a un servizio al quale non ha diritto.

Nel caso del Pescatore e del Buttafuori, il sistema può facilmente contrastare l'attacco utilizzando ***TLS 1.3***. Questo protocollo crittografa la comunicazione tra il server e l'utente, prevenendo la lettura dei dati da parte degli attaccanti sul canale di comunicazione. Inoltre, TLS 1.3 assicura l'integrità dei messaggi mediante algoritmi di hashing crittografici, che generano codici di autenticazione dei messaggi (MAC). Ogni messaggio scambiato viene controllato e, qualora anche una minima parte di esso venga modificata, il MAC non corrisponderà, portando alla considerazione del messaggio come alterato e alla sua conseguente reiezione.

Per quanto riguarda l'attacco del Mario in the Middle, il sistema non lo blocca automaticamente; se l'attaccante riesce a completare la prova di Schnorr al posto dell'utente, può inserire la sua chiave pubblica all'interno del Merkle Tree e, appropriandosene, superare la seconda prova di Schnorr per ottenere l'accesso con le credenziali inviate al server. Tuttavia, a vantaggio della nostra soluzione, è vero che TLS 1.3 incorpora meccanismi che riducono la probabilità di un replay attack di questo tipo. Per contrastare ulteriormente questo problema, la soluzione include la ***fase di verifica aggiuntiva*** in cui si controlla se, partendo dalla sua chiave pubblica, si riesce a risalire alla radice nel Merkle Tree fornito dallo S_C.

D'altro canto, il sistema non può contrastare un avversario come Doszilla, il cui scopo è sovraccaricare i server di richieste per impedire l'accesso al servizio, a meno di non adottare procedure di ***decentralizzazione*** dei server. Benché si sarebbe potuto implementare tale strategia, si è scelto un approccio più incline alla ***protezione dei dati scambiati***, trovando un compromesso tra efficienza e sicurezza. Sebbene possano verificarsi disservizi, i dati rimangono confidenziali grazie alla protezione offerta da TLS 1.3, indipendentemente dal server bersagliato da Doszilla. Ricordiamo che tale tipo di attacco avverrà anche nelle due fasi descritte di seguito dunque si farà a meno di ripetere tale descrizione.

Per quanto riguarda l'avversario Mariuolo, egli può essere fermato solo se la carta impiegata illecitamente è stata inserita nella ***lista delle carte revocate dall'IPZS***. In questo caso, è possibile ottenere una prova del reato, poiché il sistema nega l'accesso e avvisa le autorità competenti. Tuttavia, in situazioni estreme, come la mancata denuncia di un furto o i casi di estorsione, la prevenzione non è garantita, poiché si tratta di illeciti fisici che un sistema digitale non può contrastare efficacemente.

3.3.2 Ottenimento delle credenziali

Durante la fase in cui le credenziali sono fornite all'utente, è essenziale che rimangano inalterate per garantire un accesso efficace al server S_1.

Ecco i principali attacchi che possono emergere in questa fase:

- **Modifica delle credenziali ritornate all'utente:**

Un avversario come il ***Buttafuori*** potrebbe tentare di modificare le credenziali fornite dal server S_C nel Merkle Tree, con l'obiettivo di ostacolare o negare l'accesso all'utente. Lo stesso risultato potrebbe essere ottenuto da un avversario come ***Waldo***, che essendo uno dei dipendenti del server e avendo accesso alle dinamiche interne, potrebbe alterare le informazioni prima che vengano inserite nel Merkle Tree.

- **Generazione di certificati falsi:**
  Il ***Pescatore*** potrebbe fingere di essere il server delle credenziali e tentare di rilasciare certificati falsi che contengono la radice di un Merkle Tree da lui creato. Non avendo accesso alle informazioni riservate a cui Sc può accedere, non sarebbe in grado di includere le corrette informazioni dell'utente all'interno del Merkle Tree.

Per il primo tipo di attacco, condotto dal Buttafuori, il sistema è in grado di sventarlo grazie a **TLS 1.3** che, per definizione, non permette modifiche ai dati trasmessi sul canale. Se l'attacco fosse invece condotto da Waldo, la situazione sarebbe più complessa poiché l'avversario ha un livello di accesso elevato all'interno del sistema. Non esistendo soluzioni definitive a tale minaccia, si confida nell'**integrità degli individui con accesso autorizzato**, monitorando e segnalando alle autorità qualsiasi comportamento sospetto.

Per quanto concerne il secondo tipo di attacco, se il Pescatore riuscisse in qualche modo a eludere i **controlli di autenticazione** di TLS, rimane il fatto che qualsiasi certificato con le credenziali debba essere necessariamente firmato dal server. Qualora il server non disponga di una chiave pubblica riconosciuta, l'utente si accorgerebbe della presenza di un server malevolo durante il **processo di verifica della firma**, visto che il Pescatore non potrebbe fornire una chiave privata validamente associata.

3.3.3 Utilizzo delle credenziali e controllo requisiti richiesti

Allo stesso modo della fase precedente, è essenziale per finalizzare l'accesso, l'inalterabilità delle credenziali che l'utente e il server si scambiano.

Tra gli attacchi chiave che possono accadere abbiamo:

- **Modifica delle credenziali ritornate all'utente:**
  Similmente all'attacco descritto in precedenza, un avversario come il ***Buttafuori*** potrebbe tentare di modificare le credenziali inserite dal server S_C nel Merkle Tree, mirando a ostacolare o negare l'accesso all'utente.
  Inoltre, in questa fase, potrebbe emergere un ulteriore rischio se il ***Bugiardo*** decidesse di alterare autonomamente le proprie credenziali per soddisfare i requisiti di accesso al server delle aste.

- **Aggregazione di credenziali:**
  Due utenti, come ***Bonnie e Clyde***, potrebbero essere intenzionati a combinare le proprie credenziali per ottenere accesso improprio a un servizio. Se ognuno di loro possedesse solo parte dei requisiti necessari per l'accesso, potrebbero pensare di unire le loro credenziali per soddisfare complessivamente i criteri richiesti e quindi accedere al servizio, anche se non ne avrebbero individualmente il diritto.

- **Presentazione di certificati falsi:**
  Un ***utente innocente***, precedentemente indotto in errore dal ***Pescatore*** a credere che il server da lui fornito fosse valido, potrebbe tentare di utilizzare un certificato rilasciato da tale server per accedere al servizio. Se ciò dovesse accadere, ne deriverebbe un grave compromesso della sicurezza, poiché l'accesso dovrebbe essere consentito solamente attraverso credenziali rilasciate da server riconosciuti dal server S_1.

- **Furto di credenziali:**
  Un avversario ***Mario in the Middle*** potrebbe intercettare il certificato contenente tutte le credenziali dell'utente e inviarlo al server S_1 per tentare di accedere al posto dell'utente in un secondo momento.

- **Aggiramento dei controlli di accesso:**
  Un ***Bugiardo*** potrebbe collaborare con un avversario come ***Waldo***, un dipendente corrotto che ha accesso al server S_1, per aggirare i controlli sui requisiti di accesso e ottenere un'entrata diretta al sistema.

Come già menzionato, il primo tipo di attacco portato dall'avversario Buttafuori può essere facilmente sventato come spiegato in precedenza; quindi, non ci soffermeremo ulteriormente su questo punto. Per quanto riguarda il Bugiardo che intende modificare le proprie credenziali per accedere a un servizio non autorizzato, tale tentativo sarà infruttuoso poiché la **verifica della radice del Merkle Tree** contenente le credenziali fallirebbe: qualsiasi alterazione delle credenziali modificherebbe infatti la radice del Merkle Tree, rendendo il certificato non valido.

Analogamente, anche Bonnie e Clyde, nel loro tentativo di combinare e modificare le rispettive credenziali nel certificato, finirebbero per alterare la radice del Merkle Tree, invalidando così la **verifica della loro autenticità**.

Nel terzo scenario, dove un utente ingenuo usa un certificato fornito dal Pescatore, la sicurezza del sistema richiede che tale certificato venga rifiutato, poiché **la firma non corrisponderà a quella di un server attendibile**, interrompendo così eventuali tentativi di phishing. Se il Pescatore si spaccia per il server S_1, il meccanismo di autenticazione e verifica previsto e spiegato in precedenza impedirà l'apertura della comunicazione in primis.

Inoltre, in caso di un attacco del Mario in the Middle, la tecnica di **prova di Schnorr** viene utilizzata per confermare l'autenticità delle credenziali della parte comunicante. A meno che l'attaccante non sia riuscito a inserire preventivamente la sua chiave pubblica nel Merkle Tree, situazione che abbiamo coperto in precedenza, il sistema riconoscerà la chiave pubblica dell'utente, priva della corrispondente chiave segreta necessaria per dimostrare il possesso delle credenziali, bloccando così efficacemente l'attacco.

Nel caso dell'ultimo tipo di attacco, purtroppo, il sistema non offre una soluzione specifica per contrastare questi attaccanti che agiscono in collaborazione. Per questo motivo, si fa affidamento sull'onestà dei soggetti che hanno un accesso legittimo ai sistemi. Inoltre, dato che i dipendenti sono identificabili, qualsiasi tentativo di frode verrebbe prontamente rilevato e segnalato alle autorità competenti.

3.4. ANALISI DELL'EFFICIENZA

Di seguito si riporta un'analisi della complessità computazionale dei principali passi che la soluzione coinvolge:

- **Generazione del Merkle-Tree**: La generazione del Merkle-Tree avviene in tempo O(n) dove n è il numero di foglie (ovvero le informazioni) che devono essere inserite nell'albero. Ogni foglia rappresenta una credenziale o un'informazione dell'utente. L'altezza del Merkle-Tree è $\log_2(n)$, che rappresenta il numero di livelli di hashing necessari per arrivare alla radice. L'invio continuo del Merkle-Tree tra i vari server e l'utente, anche se può sembrare dispendioso, non lo è particolarmente grazie alla proprietà intrinseca dei Merkle-Tree di mantenere una dimensione costante di 256 bit, anche in presenza di un gran numero di credenziali.

- **Verifica delle credenziali**: La verifica delle credenziali tramite il Merkle-Tree avviene in tempo $O(\log_2(n))$, dove n è il numero totale di informazioni. Questo tempo è determinato dalla necessità di ricostruire il percorso dalla foglia alla radice, verificando ogni nodo intermedio. La verifica della firma digitale sulla radice del Merkle-Tree avviene in tempo O(1), grazie all'efficienza degli algoritmi di firma digitale utilizzati.

- **Accesso al server d'aste**: L'accesso al server d'aste richiede la verifica del certificato digitale e delle credenziali dell'utente. La verifica del certificato avviene in tempo O(1) poiché si tratta di una singola operazione di controllo. La ricostruzione del Merkle-Tree e la verifica delle credenziali avviene in tempo $O(\log_2(n))$, garantendo che anche con un numero elevato di credenziali, il processo rimanga efficiente.

- **Generazione delle credenziali**: La generazione delle credenziali da parte del server delle credenziali avviene in due fasi. Prima, viene generato il Merkle-Tree in tempo O(n). Successivamente, le credenziali vengono inviate all'utente in tempo O(1) poiché il certificato e gli hash intermedi possono essere trasmessi rapidamente.

- **Richiesta delle credenziali**: La richiesta delle credenziali coinvolge l'uso del protocollo Zero-Knowledge di Schnorr, che avviene in tempo O(1) per ogni interazione (invio della chiave pubblica, sfida, risposta, e verifica). Questo garantisce che il processo di autenticazione sia rapido e scalabile.

- **Invio e ricezione delle credenziali**: L'invio delle credenziali dall'utente al server d'aste, così come la loro ricezione, avviene in tempo O(1). Il processo di autenticazione, che include la verifica della firma digitale e delle credenziali tramite il Merkle-Tree, avviene in tempo $O(\log_2(n))$.

Possiamo concludere che il sistema è complessivamente efficiente, poiché le comunicazioni, fatta eccezione per alcuni specifici casi, risultano essere rapide, a meno che i server non si trovino in condizioni di sovraccarico. È vero che la quantità delle verifiche necessarie potrebbe rallentare l'intero processo; tuttavia, si è scelto di privilegiare la sicurezza a scapito di una leggera perdita in efficienza. Questa decisione è giustificata soprattutto alla luce degli attacchi

potenziali che abbiamo analizzato in precedenza, mostrando come un rinforzo nella sicurezza sia essenziale per prevenire violazioni e garantire la protezione dei dati.

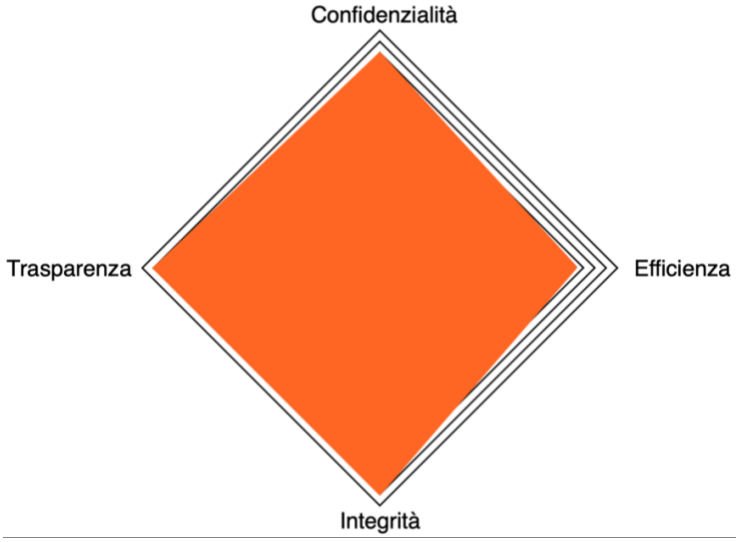
3.5. ANALISI DELLA TRASPARENZA

Le funzionalità del sistema sono state concepite con chiarezza e trasparenza:

- Le credenziali sono emesse da S_C, considerata un'autorità di fiducia, seguendo una struttura chiara e ben definita.
- L'accesso ai servizi è garantito esclusivamente agli utenti che detengono credenziali valide e che soddisfano i requisiti precisamente definiti dal server.

Queste misure sono efficaci a meno che non emergano situazioni particolari, come la presenza di un attacco in atto o la commissione di un reato. Il design del sistema tiene conto della maggior parte delle situazioni in cui gli utenti possono agire in modo scorretto, riducendo la necessità di fare affidamento su parti terze fidate. Tutte le operazioni, inclusi i processi di TLS e le transazioni che implicano l'uso di informazioni come chiavi pubbliche e credenziali, sono eseguite con verificabilità pubblica, assicurando un alto livello di trasparenza.

3.6. RIEPILOGO



- Confidenzialità: 8
- Efficienza: 6.5
- Integrità: 9
- Trasparenza: 9

Osserviamo un calo nell'efficienza, attribuibile alla scelta di privilegiare la sicurezza in termini di confidenzialità e integrità. Abbiamo preferito sacrificare un po' di efficienza per evitare la fuga di dati, ritenendo questa priorità più adeguata al nostro applicativo.

4. WP4: IMPLEMENTAZIONE

In questa sezione verrà presentata una possibile implementazione della soluzione proposta nel WP2. Per motivi di semplicità, si è scelto di implementare il tutto mediante script bash, simulando la comunicazione tra i server e un potenziale client attraverso un insieme di apposite cartelle.

4.1. COME AVVIARE LA SIMULAZIONE

Per avviare la simulazione, è necessario, tramite terminale:

- Spostarsi nella cartella Wp4
- Eseguire la simulazione mediante il comando: bash ./simulazione.sh
- Seguire le istruzioni riportate a video

4.2. SCENARIO DI SIMULAZIONE

Ciò che andremo a simulare è uno scenario in cui un utente, User1 (anche noto come Giammarco), desidera accedere al server d'asta. Giammarco possiede i requisiti necessari per l'accesso, ovvero la maggiore età.
Gli attori coinvolti in questo processo sono:

- **Utente (User1):** che cerca di accedere al server d'asta per partecipare alle attività di vendita e acquisto.
- **Server d'asta:** che gestisce le aste online, ricevendo richieste dagli utenti e fornendo loro accesso alle varie operazioni d'asta.
- **Server delle credenziali:** responsabile del rilascio delle credenziali degli utenti, verificando l'identità dell'utente.

In questo contesto, la simulazione mira a replicare le interazioni tra questi attori, dimostrando sia il flusso di comunicazione quando l'utente ha già le credenziali necessarie sia il processo di ottenimento delle credenziali in caso contrario. Questo ci permetterà di osservare come il sistema gestisce diverse situazioni e garantisce un accesso sicuro e controllato al server d'asta.

4.3. ASSUNZIONI

- Il protocollo Zero-Knowledge di Schnorr viene simulato attraverso la firma e la verifica della firma di un file, garantendo la riservatezza delle informazioni trasmesse.
- Le chiavi crittografiche utilizzate dagli attori coinvolti in questa simulazione sono di tipo ECDSA, assicurando un elevato livello di sicurezza ed efficienza nelle operazioni crittografiche.
- Per quanto riguarda la simulazione del TLS 1.3, la comunicazione viene rappresentata copiando i file da scambiare nelle apposite cartelle. In un ambiente reale, questa comunicazione sicura avverrebbe tramite un canale TLS 1.3, garantendo l'integrità e la riservatezza dei dati scambiati.

- Il server delle credenziali opera come una Certification Authority (CA) radice fidata, capace di generare certificati auto firmati attraverso il comando descritto nel punto precedente. Questo gli conferisce l'autorità di convalidare le chiavi generate dagli utenti, assicurando che solo entità legittime possano accedere al sistema.
- Per motivi di semplicità si evita di implementare la logica delle CIE dismesse/rubate in tale WP.

4.4. IMPLEMENTAZIONE

L'implementazione proposta si suddivide su più sottoscript, tutti richiamati dallo script principale *simulazione.sh*:

- **start_user.sh:** tale script si occupa di generare la cartella contenente la struttura di sottocartelle per gestire l'utente, di importare in tale cartella il file di configurazione dell'utente (dove verrà salvata la root del Merkle Tree contenente le credenziali e l'input password) e di generare le chiavi pubbliche e private rispettivamente per la CIE e per l'utente.

- **start_servers_web.sh:** tale script si occupa di creare le directory necessarie per due server e una Certification Authority (CA), genera una chiave privata EC per la CA, e crea un certificato autofirmato per la CA. Al termine, conferma che le chiavi sono state generate con successo per entrambi i server.

- **gen_mt_u1.sh:** si occupa di creare il Merkle Tree con i dati dell'utente 1. La root del Merkle Tree verrà poi inserita nel certificato.

- **schnorr_proof_sc.sh**: tale script si occupa di simulare l'invio e la verifica di una chiave pubblica, e l'implementazione del protocollo di Schnorr tra l'utente e il server delle credenziali, per certificare la coppia di chiavi che si crea l'utente, tramite la firma e verifica di un messaggio. Controlla e copia i file necessari tra l'utente e il server delle credenziali, e verifica la firma utilizzando OpenSSL.

- **richiesta_credenziali.sh**: tale script si occupa di gestire la richiesta di credenziali da parte di un utente verso il server delle credenziali, generando e verificando una richiesta di certificato e firme digitali, e interagendo con un server delle credenziali. Se la verifica ha successo, procede con la generazione e verifica di un Merkle Tree; in caso contrario, o su comando dell'utente, pulisce le risorse e termina l'operazione.

- **verifica_pin.sh**: tale script si occupa di simulare l'uso di una carta NFC per verificare un PIN inserito dall'utente, autorizzando l'accesso se il PIN è corretto e negandolo se il PIN è errato.

- **invio_credenziali_server_aste.sh**: tale script si occupa di simulare l'invio del certificato contenente il Merkle Tree al server d'aste, verificando sia il certificato stesso che la ricostruzione per confermare la root. In caso di errore nella verifica, esegue la pulizia delle risorse e termina l'operazione.

- **verifica_merkle_tree.sh:** tale script si occupa di verificare un certificato contenente le credenziali e il Merkle Tree, nonché di estrarre e calcolare i vari hash dai dati del certificato per ricostruire e verificare la radice del Merkle Tree. Se la radice calcolata corrisponde a quella nel certificato, la verifica è considerata riuscita; in caso contrario, segnala un errore.

- **verifica_identità_utente.sh:** Tale script verifica l'esistenza del certificato contenente le credenziali, estrae i campi di interesse, esegue il protocollo di Schnorr per confermare che l'utente sia effettivamente il proprietario delle credenziali attraverso la chiave pubblica contenuta nel Merkle Tree e, infine, controlla se l'utente è maggiorenne calcolando l'età dalla data di nascita. Se tutte le verifiche risultano corrette, l'accesso viene confermato; in caso contrario, segnala errori e termina l'operazione.

5. CAMBIAMENTI

In questa sezione verranno elencati i cambiamenti effettuati rispetto alla prima consegna.

5.1. WP1

All'interno del WP1 sono stati effettuati i seguenti cambiamenti:

- È stato eliminato l'avversario "Il molesto";
- Sono stati aggiunti tre nuovi avversari: "Pescatore", "Doszilla", "Bugiardo".
- Sono state aggiunte nella completeness alcuni scambi. In questa nuova versione l'utente si crea una nuova coppia di chiave pubblica e privata. Inoltre, sono state eliminate le transizioni del processo in cui l'utente effettuava una firma di un certificato mandato dal server d'asta per attestare la propria proprietà sulle credenziali, questa funzionalità nella nuova versione viene comunque coperta grazie alla presenza della chiave pubblica dell'utente all'interno del Merkle Tree.

5.2. WP2

All'interno del WP2 la soluzione risulta simile alla prima consegna, con l'aggiunta di una generazione di un'ulteriore chiave pubblica e privata.