Università degli Studi di Salerno

CORSO DI LAUREA MAGISTRALE: INGEGNERIA INFORMATICA

.DIEM

Progetto Algoritmi e Protocolli per la Sicurezza: Condivisione Selettiva di Credenziali Accademiche

Group Number 28:
Anzivino Giuseppe Fabrizio
Calabrese Raffaele

Anno Accademico 2024/2025

Assegnazione dei WP

	WP1	WP2	WP3	WP4
Studente	Giuseppe Anzivino	Raffaele Calabrese	Raffaele Calabrese	Giuseppe Anzivino

Indice

1. WP1

1.1. Contesto

Il progetto si presenta nel complicato contesto della mobilità studentesca internazionale, con particolare interesse nello scenario del programma Erasmus. La sfida più ardua da affrontare è la condivisione selettiva di credenziali accademiche, come esami superati, titoli ottenuti e attestazioni di frequenza, tra diversi enti di istruzione quali le Università. Al momento i meccanismi in uso prevedono lo scambio manuale o l'uso di sistemi centralizzati che presentano problematiche e limitazioni in termini di sicurezza, privacy e revocabilità. Tali sistemi comportano fiducia nell'autorità centrale che si occupa di condividere un insieme di informazioni che non garantiscono nemmeno la privacy dello studente costringendo a rivelare più informazioni del necessario. L'obiettivo del progetto è quello di superare questo problema attraverso la realizzazione di un sistema decentralizzato che prevede le funzionalità per:

- il rilascio delle credenziali,
- la presentazione delle credenziali,
- la revoca delle credenziali.

Inoltre il sistema deve permettere di condividere le proprie informazioni in modo selettivo e verificabile, facilitandone il riconoscimento, snellendo i processi burocratici transazionionali e risolvendo le problematiche legate alle diversità di sistemi accademici.

1.2. Attori

Nel contesto generale appena definito, risulta importante anche identificare i principali attori coinvolti con gli obiettivi principali del sistema da realizzare. Tali attori, considerati tutti autenticati e benevoli, svolgono ruoli distinti e interagiscono per garantire il corretto funzionamento e la sicurezza del sistema.

Obiettivi

Nella seguente tabella sono riportati i diversi attori identificati e i rispettivi obiettivi:

Attori Benevoli (Autenticati)	
Attore	**Obiettivi**
Studente	Ottenere credenziali accademiche verificate, conservarle in modo sicuro e protetto.
	Presentare credenziali accademiche alle università o enti interessati, condividendo solo le informazioni necessarie.
	Essere sicuri che la propria privacy sia garantita e che le informazioni presentate siano verificabili senza esporre dati non richiesti.
Università Ospitante (*Università di Rennes*)	Emettere credenziali accademiche verificate e accurate per i propri studenti.
	Garantire che le informazioni rilasciate siano affidabili e che possano essere verificate da terze parti senza inficiare sulla privacy dello studente.
	Avere la possibilità di revocare le credenziali in caso di comportamenti scorretti o scadenze, notificando lo stato di revoca in modo trasparente.
Università di Origine (*Università di Salerno*)	Ricevere le credenziali digitali dagli studenti in mobilità.
	Verificare l'autenticità e la validità delle credenziali presentate senza dover interrogare direttamente l'università di origine.
	Gestire in modo sicuro e privato le informazioni ricevute, assicurando che lo studente possa controllare i dati condivisi.
Enti di accreditamento	Certificare e verificare l'autenticità delle credenziali rilasciate dalle università stabilendo standard e linee guida per l'emissione e la gestione delle credenziali.

Interazioni

Il sistema è progettato in modo da supportare un insieme ben definito di interazioni tra i diversi attori coinvolti. Le principali interazioni identificate sono:

- Certificazione da parte dell'Ente di Accreditamento: l'Università ospitante prima di emettere le credenziali deve essere riconosciuta come entità fidata da un Ente di Accreditamento, che ne rilascia un certificato.
- Emissione delle credenziali: l'Università ospitante emette e firma digitalmente le credenziali accademiche per un determinato studente, attestanti il superamento di esami o il conseguimento di titoli.
- Ricezione e conservazione: lo Studente riceve la credenziale digitale dall'Università ospitante e la conserva (es. tramite uno smartphone o pc) permettendogli di gestire e controllare i propri dati.
- Presentazione selettiva: quando lo Studente deve dimostrare le proprie qualifiche, può presentare in modo selettivo solo le informazioni strettamente necessarie, criptandole e senza rivelare la credenziale nella sua interezza.
- Verifica della credenziale: l'Università di origine riceve la presentazione selettiva dello studente e ne verifica la validità, l'integrità e controlla lo stato di revoca. Questa verifica avviene senza contattare in maniera diretta l'Università ospitante.
- Revoca della credenziale: in caso di errore o compromissione, l'Università ospitante può revocare una credenziale precedentemente emessa.

1.3. Funzionalità

Con una chiara identificazione degli attori e delle loro interazioni, il gruppo ha deciso di definire anche le funzionalità essenziali che il sistema deve implementare per soddisfare i requisiti richiesti. Nella seguente tabella sono riportate le funzionalità identificate con una breve descrizione:

Funzionalità	Descrizione
Autenticazione	Solo gli utenti legittimi e autenticati possono accedere e usufruire dei servizi offerti dal sistema. Per tal motivo è necessario garantire un processo di autenticazione robusto e sicuro, minimizzando il rischio di impersonificazione e accesso non autorizzato.
Certificazione Università	Per essere riconosciute come entità affidabili, le università devono essere munite di un certificato rilasciato da un Ente di Accreditamento.
Emissione Credenziali Digitali	Le Università Ospitanti possono creare e firmare digitalmente credenziali accademiche per i propri studenti. Esse conterranno le informazioni ben strutturate e verificate che si desidera condividere.
Conservazione Credenziali Digitali	Gli studenti devono poter archiviare in modo sicuro e privato le proprie credenziali digitali. Il sistema deve garantire la protezione contro l'accesso non autorizzato e la perdita di dati, consentendo agli studenti di mantenere il controllo sulle proprie informazioni accademiche.
Divulgazione Selettiva	Gli studenti hanno la capacità di scegliere e condividere solo un sottoinsieme specifico delle informazioni contenute nelle credenziali. Questo è cruciale per garantire la privacy dello studente.
Verifica Credenziali Digitali	Le università o gli enti riceventi possono verificare l'autenticità e la validità delle informazioni condivise dallo studente senza dover interrogare in maniera diretta l'ente emittente per ogni verifica.
Revoca Credenziali Digitali	Le credenziali accademiche possono essere invalidate in modo sicuro in seguito a scadenze o comportamenti illeciti. La revoca deve essere pubblicamente verificabile, permettendo a qualsiasi parte di controllare lo stato di validità di una credenziale.
Consumo minimo di memoria e di rete da parte del sistema	Il sistema deve poter essere progettato per minimizzare i consumi di memoria e di rete durante l'emissione, la conservazione, la presentazione e la verifica delle credenziali.

1.4. Threat Model

Una volta stabilite le funzionalità, è fondamentale analizzare anche le possibili minacce che potrebbero compromettere la sicurezza e l'affidabilità del sistema. A tal fine il Threat Model è una componente importante per la progettazione di un sistema sicuro che aiuta a comprendere e a combattere le possibili minacce che potrebbero compromettere la sicurezza e l'affidabilità del sistema.

Tipologie di Attaccanti

Il threat model realizzato prende in considerazione sia la presenza di attaccanti interni al sistema che attaccanti esterni ad esso dividendoli a seconda di come agiscono in due tipologie:

- Attaccanti passivi, ossia che si limitano ad osservare ed intercettare le informazioni importanti all'interno della rete
- Attaccanti attivi, ossia chi cerca di manomettere l'intero sistema impedendone il corretto funzionamento.

Il Thread model identificato è riportato nella seguente tabella:

Attaccante	Descrizione	Tipo
Studente malevolo	Uno studente munito di credenziali legittime che tenta di modificarle a fini illegittimi (es. aggiunta esami non sostenuti, cambio voti) per ottenere vantaggi accademici.	ATTIVO
Eavesdropper	Un attaccante passivo che limita ad osservare ed intercettare le comunicazioni di rete per ottenere dati sensibili o pattern comportamentali nella comunicazione tra studente ed istituzioni.	PASSIVO
Man in the Middle	Un attaccante che si intromette attivamente nella comunicazione tra studente ed università, potendo alterare o intercettare informazioni in transito. Può tentare di riutilizzare e far sembrare ancora valida una presentazione registrata in precedenza (Replay Attack).	ATTIVO
Attaccante DoS	Un attaccante che mira a rendere non disponibili o inaccessibili i servizi del sistema, sfruttando vulnerabilità o inviando un numero elevato di richieste per causare rallentamenti, sovraccarichi o rotture.	ATTIVO
Impersonificatore	Un attaccante che si finge un attore legittimo al fine di ottenere vantaggi illegittimi (es. ottenimento credenziali che non gli spettano, presentazione di credenziali altrui).	ATTIVO
Verificatore malintenzionato	Un membro di un'università ricevente o un altro verificatore autorizzato, legittimato a ricevere credenziali studentesche, che tenta di compiere azioni illegittime (es. forzare lo studente a divulgare più attributi della propria credenziale).	ATTIVO/ PASSIVO

Capacità e Obiettivi

Dopo aver categorizzato le diverse tipologie di attaccanti, risulta utile anche delineare le loro capacità e gli obiettivi che intendono raggiungere eseguendo un attacco. Infatti comprendere cosa un attaccante può fare e cosa intende ottenere è cruciale per sviluppare contromisure efficienti per garantire che il sistema sia sicuro, disponibile e correttamente funzionante. La seguente tabella riporta le risorse e le abilità a disposizione di ogni attaccante, insieme ai loro scopi nel compromettere il sistema:

Attaccante	Capacità e risorse	Obiettivi
Studente malevolo	Potenziali conoscenze degli strumenti e protocolli utilizzati dal sistema.	Falsificazione di documenti.
	Capacità di utilizzare software di manipolazione dei dati per alterare le firme digitali.	Ottenimento illecito di titoli o riconoscimenti.
	Limitate risorse computazionali.	Compromissione dei dati.
Eavesdropper	Capacità di monitorare il traffico di rete.	Raccolta illecita di informazioni sensibili.
	Competenze di analisi del traffico per ottenere informazioni.	
	Intercettazioni di chiavi o altri dati non cifrati in transito.	
	Discrete risorse computazionali.	
Man in the Middle	Intercettazione e manipolazione dei pacchetti (Packet injection, packet rewriting).	Alterazioni di credenziali.
		Replay Attack.
	Creazione di certificati falsi o possesso di certificati tramite terze parti.	Furto di dati.
	Conoscenza delle vulnerabilità e bypassing dei meccanismi di protezione.	Interruzione di comunicazione.
	Maggiore potenza di calcolo rispetto all'Eavesdropper.	
Attaccante DoS	Conoscenza delle debolezze architetturali del sistema.	Interruzione del servizio.
	Possibilità di generare traffico massivo.	Danni reputazionali.
	Possibilità di sferrare attacchi distribuiti.	Potenziali danni economici.
	Elevate risorse computazionali.	

Impersonificatore	Accesso a credenziali legittime tramite phishing, furto di dati, malware.	Furto di identità
	Simulazione del comportamento di un utente autenticato tramite le credenziali rubate.	Accesso non autorizzato ai dati.
	Elusione di controlli sull'identità del possessore delle credenziali tramite Replay Attack.	Potenziali danni reputazionali.
	Discrete risorse computazionali.	
Verificatore malintenzionato	Accesso legittimo alle credenziali.	Violazione della privacy.
	Capacità di tracciamento delle presentazioni ricevute nel tempo tramite strumenti di pattern analysis, logging, fingerprinting.	Abuso di dati personali.
	Discrete risorse computazionali.	

Schemi di Flusso

E' possibile mostrare visivamente come avvengono i diversi tipi di attacchi attraverso degli schemi di flusso, di seguito riportati. Per prima cosa mostriamo come dovrebbe avvenire normalmente il meccanismo di condivisione delle credenziali in assenza di problematiche e/o attacchi.

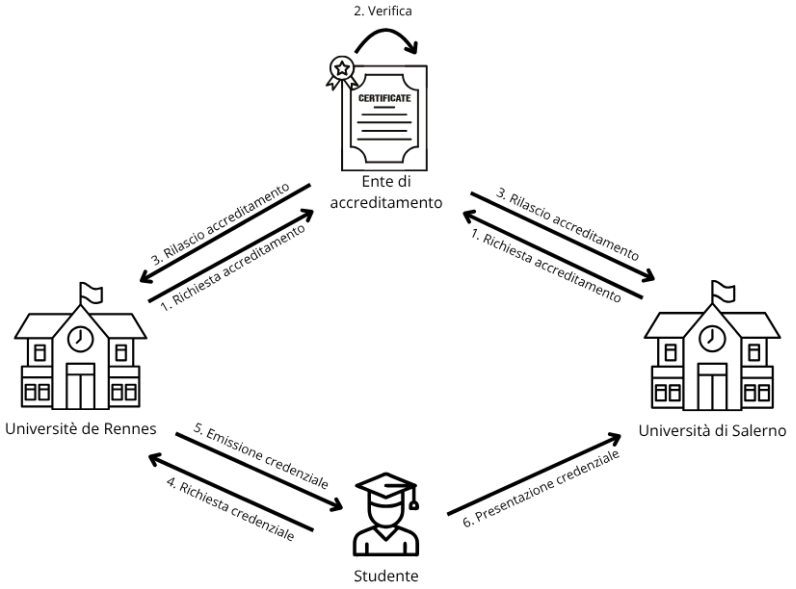


Figura 1: Scenario generale

Passiamo ora ai flussi delle casistiche di attacchi passivi e attivi.

Studente malevolo

Nel caso dello studente malevolo, l'accesso è consentito in quanto egli è munito di credenziali di accesso, tuttavia può compromettere le informazioni ricevute dalle Università e quindi falsificare documenti.

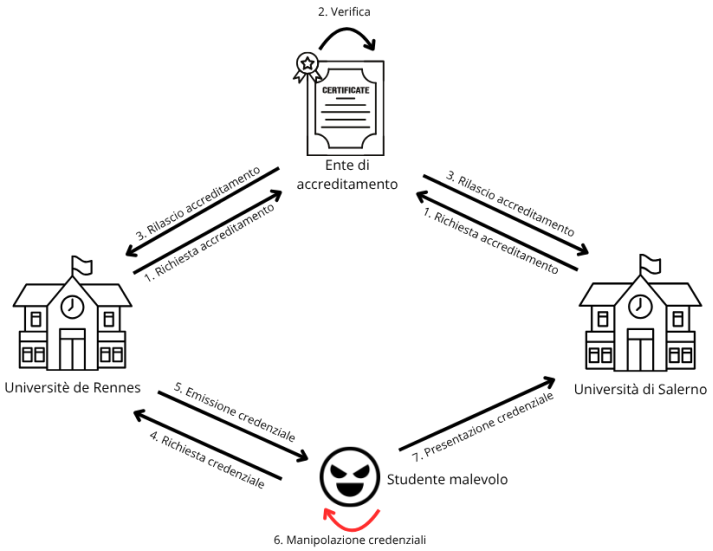


Figura 2: Studente malevolo

Eavesdropper

L'eavesdropper effettua packet sniffing al fine di ottenere informazioni private e sensibili. Può monitorare il traffico di informazioni tra due enti in comunicazione. La figura 3 mostra solo una delle possibili casistiche, ovvero l'intercettazione di informazioni tra Università di Rennes e lo studente.

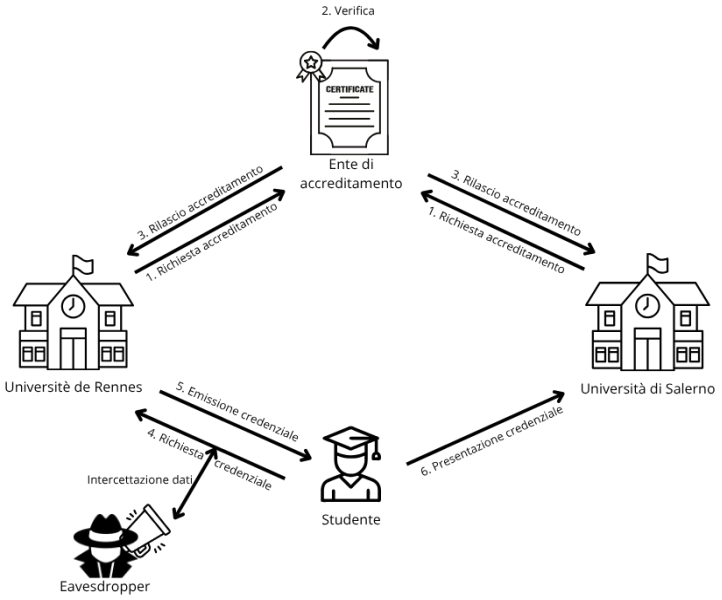


Figura 3: Eavesdropper

Man in the Middle

Il Man in the Middle oltre ad intercettare i dati può anche manipolarli, è quindi un rischio ben maggiore dell'eavesdropper. La figura 4 mostra, come esempio, la manipolazione delle informazioni scambiate tra l'Università di Rennes e lo studente.

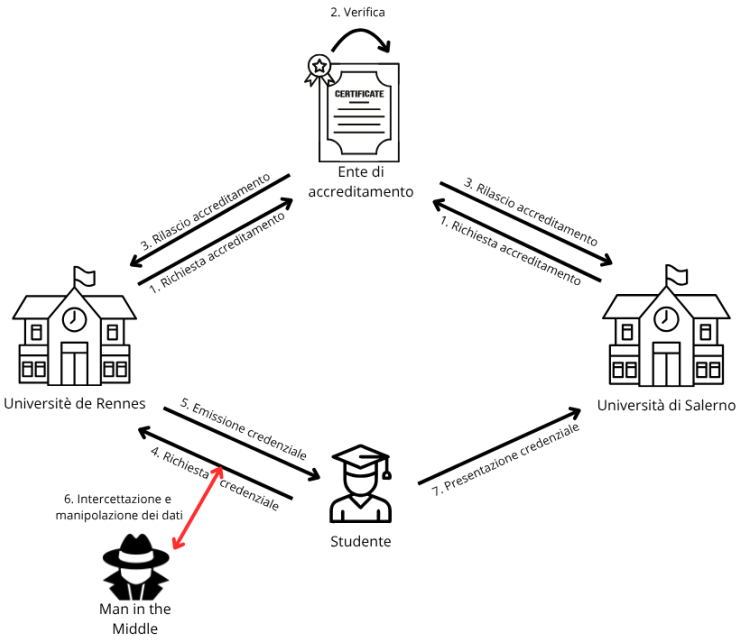


Figura 4: Man in the Middle

Attaccante DoS

L'attaccante DoS può bersagliare qualsiasi ente coinvolto nel sistema di condivisione delle credenziali. In figura 5 il bersaglio è l'Università di Salerno, che in seguito al traffico massivo generato dall'attaccante, subisce interruzioni del suo servizio.

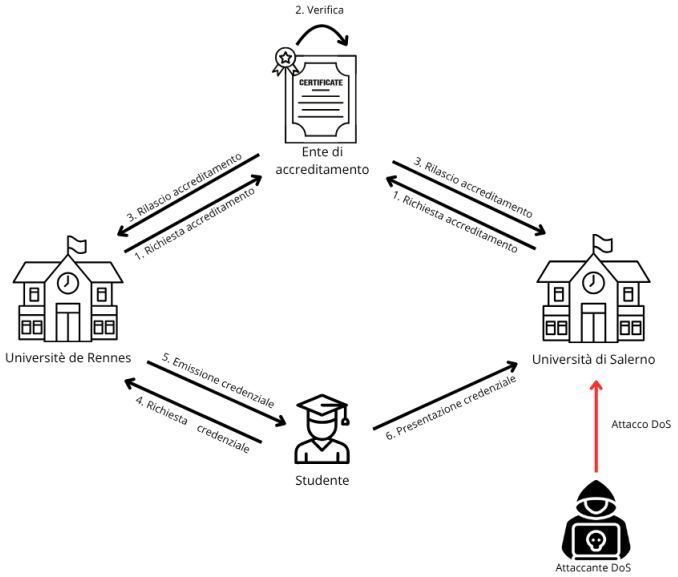


Figura 5: Attaccante DoS

Impersonificatore

Un impersonificatore riesce ad accedere in modo fraudolento al sistema dopo aver messo le mani su delle credenziali legittime, tramite phishing, malware o furto di dati. Si finge quindi un utente legittimo a tutti gli effetti.

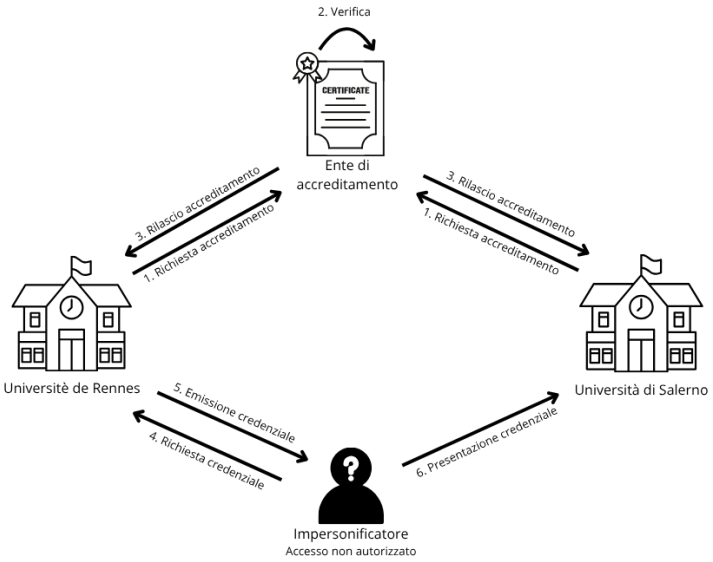


Figura 6: Impersonificatore

Verificatore malintenzionato

Oltre a persone esterne, una minaccia possibile può essere il personale munito di credenziali legittime ma che ha fini illegittimi. Il verificatore malintenzionato può violare la privacy delle parti oneste e abusare dei dati personali raccolti. Nell'esempio in figura 7 il verificatore possiede credenziali dell'Università di Salerno.

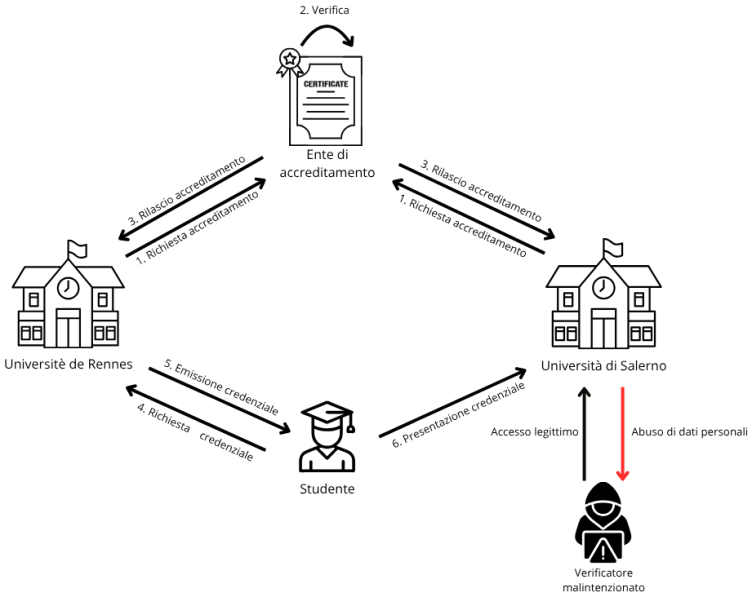


Figura 7: Verificatore malintenzionato

1.5. Proprietà Desiderate

Per contrastare in modo efficace le minacce individuate nel paragrafo precedente sono state definite anche le proprietà che il sistema deve garantire per assicurare la sicurezza, la privacy e l'affidabilità delle credenziali.

Definizione

Queste proprietà sono fondamentali per valutare se il sistema soddisfa i requisiti richiesti e se può resistere agli attacchi o ai tentativi di compromissione da parte di attori malevoli. Le proprietà identificate sono state riportate nella seguente tabella:

Proprietà	Descrizione	Minacce Affrontate
Confidenzialità	Garantisce che solo le parti autorizzate possano accedere alle informazioni sensibili che si vogliono presentare. Le informazioni non necessarie devono rimanere private.	Eavesdropper
		Verificatore malintenzionato
		Impersonificatore
Integrità	Garantisce che i dati contenuti nelle credenziali non possano essere modificati o alterati in modo non autorizzato dopo l'emissione.	Studente malevolo
		Man in the Middle
		Impersonificatore
Autenticazione	Consente di verificare che le informazioni provengano effettivamente dal mittente designato.	Impersonificatore
		Verificatore malintenzionato
		Man in the Middle
Non-Ripudio	Impedisce che il mittente possa negare la validità dell'azione di invio o presentazione delle informazioni.	Studente malevolo
		Impersonificatore
Revocabilità	Garantisce che una credenziale possa essere invalidata in modo sicuro e permanente in caso di informazioni errate/obsolete o di comportamenti illeciti da parte del titolare.	Studente malevolo
		Man in the Middle
		Impersonificatore
Trasparenza	Assicura che tutte le azioni relative alla gestione delle credenziali siano verificabili, in modo che tutte le parti possano verificarne l'autenticità e la validità.	Verificatore malintenzionato
		Attaccante DoS
		Studente malevolo

Privacy dello Studente	Garantisce che lo studente abbia il controllo granulare sulla divulgazione delle proprie informazioni accademiche, potendo rivelare solo il sottoinsieme strettamente necessario e proteggendo i dati non pertinenti.	Verificatore malintenzionato
		Eavesdropper
		Man in the Middle
Decentralizzazione	Assicura che il sistema non dipenda da una singola autorità centrale per il suo funzionamento, l'autenticazione, la validazione o la revoca delle credenziali, migliorando robustezza, resilienza e resistenza alla censura.	Attaccante DoS
		Man in the Middle
Efficienza	Garantisce che tutte le operazioni del sistema siano eseguite in modo rapido e con un minimo consumo di memoria e rete, permettendo l'utilizzo anche a dispositivi con risorse limitate.	Attaccante DoS

Sotto Proprietà

Di seguito, per ciascuna proprietà principale, vengono specificate delle sotto proprietà che delineano in modo più preciso i requisiti di sicurezza, privacy e funzionalità del sistema. Ognuna di essa verrà accompagnata da un identificativo:

1. Confidenzialità
    - C.1: I dati completi delle credenziali accademiche devono essere accessibili solo allo studente che ne è titolare e, in modo controllato, alle parti autorizzate (università o enti riceventi) che richiedono una specifica divulgazione.
    - C.2: Le comunicazioni tra lo studente e le università/enti devono essere protette da possibili intercettazioni non autorizzate.
    - C.3: Il meccanismo di divulgazione selettiva deve impedire la rivelazione di dati non richiesti, anche se la credenziale completa contiene tali informazioni.
    - C.4: Un verificatore malintenzionato non deve poter forzare la divulgazione di attributi aggiuntivi rispetto a quelli esplicitamente scelti dallo studente.
2. Integrità
    - I.1: Le credenziali accademiche, una volta emesse e firmate digitalmente dall'Università Ospitante, non devono poter essere falsificate, manomesse o alterate.
    - I.2: Qualsiasi alterazione avvenuta sulla credenziale deve essere rivelabile dal verificatore.
    - I.3: Le informazioni presentate in modo selettivo dallo studente devono mantenere la loro integrità e non devono poter essere alterate in transito.
    - I.4: Il processo di verifica deve confermare l'integrità delle informazioni presentate rispetto alle credenziali originali firmate.
3. Autenticazione
    - A.1: Solo gli utenti legittimi e autenticati possono accedere ed interagire con i servizi offerti dal sistema.
    - A.2: L'identità dello studente che presenta le credenziali deve essere verificabile e non impersonificabile.
    - A.3: L'identità dell'università deve essere verificabile durante il processo di verifica delle credenziali, assicurando che l'emittente sia chi dichiara di essere.

- A.4: Le chiavi delle università e degli enti devono essere autenticate e gestite in modo affidabile per prevenire possibili attacchi.

4. Non Ripudio
    - NR.1: L'università Ospitante non può negare di aver rilasciato una credenziale valida.
    - NR.2: Lo studente non può negare di aver presentato un sottoinsieme specifico di informazioni dalla sua credenziale in modo da evitare che uno studente malevolo o un impersonificare possano rinnegare un'azione fraudolenta svolta.
    - NR.3: Le azioni di revoca di una credenziale devono essere attribuibili all'entità che le ha eseguite.
5. Revocabilità
    - R.1: Le università emittenti devono avere la capacità di invalidare credenziali specifiche in caso di errore, scadenze o comportamenti illeciti.
    - R.2: Lo stato di revoca di una credenziale deve essere pubblico, in modo da essere facilmente verificabile da qualsiasi parte interessata.
    - R.3: La revoca deve essere irreversibile e deve essere possibile aggiornare lo stato in tempo utile per prevenire l'uso di credenziali invalidate.
    - R.4: Il meccanismo di revoca non deve compromettere la privacy dello studente rivelando informazioni non pertinenti al di là dello stato di validità della credenziale
6. Trasparenza
    - T.1: I meccanismi e i protocolli per l'emissione, la presentazione, le verifica e la revoca delle credenziali devono essere basati su standard aperti e pubblicamente noti, consentendo a tutti gli attori di comprendere e verificare il funzionamento del sistema.
    - T.2: La validità e l'autenticità delle credenziali devono essere verificabili da terze parti senza richiedere l'interazione diretta dell'Università Ospitante per ogni verifica.
    - T.3: Le procedure e le politiche relative alla gestione delle credenziali devono essere chiare, documentate e comprensibili a tutti gli attori.
7. Privacy dello Studente
    - P.1: Lo studente deve avere il controllo granulare sulla divulgazione delle proprie informazioni accademiche, in modo da poter scegliere solo un sottoinsieme strettamente necessario di informazioni da condividere.
    - P.2: La presentazione selettiva non deve rivelare implicitamente altri attributi non scelti dallo studente, proteggendo da tecniche di correlazione.
    - P.3: Il sistema deve proteggere da tecniche di profilazione o fingerprinting basate sui dati divulgati.
8. Decentralizzazione
    - D.1: Il sistema deve operare senza un singolo punto di fallimento o un'unica autorità centrale di controllo per l'emissione, la gestione e la verifica delle credenziali.
    - D.2: La verifica delle credenziali non deve dipendere dall'interrogazione diretta e continua dell'Università Ospitante.
    - D.3: Il sistema deve resistere ad attacchi mirati a singole entità, garantendo la disponibilità complessiva del servizio di verifica.
    - D.4: La gestione delle chiavi crittografiche e delle identità deve essere distribuita tra i diversi attori coinvolti.
9. Efficienza
    - E.1: L'emissione delle credenziali deve essere efficiente in termini di tempo e risorse computazionali consentendo un rilascio rapido e scalabile.

- E.2: La conservazione delle credenziali sul dispositivo dello studente deve minimizzare il consumo di memoria, rendendo il sistema utilizzabile anche su dispositivi con risorse limitate.
- E.3: La presentazione selettiva delle credenziali deve essere rapida e deve minimizzare il traffico di rete.
- E.4: La verifica delle credenziali ricevute deve essere veloce ed efficiente in termini di risorse computazionali.

1.6. Struttura delle Credenziali

Le credenziali rilasciate dall'Università allo studente sono in formato JSON, in modo da suddividere le informazioni in gruppi e permettere una chiara leggibilità.

- credentialID è l'identificatore univoco della credenziale
- issuer è l'università che ha rilasciato la credenziale
    - name
    - code
- emissionDate, la data di emissione della credenziale
- valid, indica se la credenziale è stata revocata o meno
- studentInfo, relativo alle informazioni anagrafiche dello studente
    - name
    - surname
    - studentId, la matricola
    - birthdate
    - nationality
    - email
    - degreeCourse, il corso di laurea
    - courseDuration, la durata del corso di laurea
    - homeUniversity, l'università di appartenenza
        - name
        - code
        - country
- erasmusInfo, contiene le informazioni che riguardano strettamente il periodo Erasmus dello studente
    - hostUniversity, l'università ospitante
        - name
        - code
        - country
    - erasmusStartDate, l'inizio del periodo di Erasmus
    - erasmusEndDate, la fine del periodo di Erasmus
    - learningAgreement, contiene le info relative al learning agreement stipulato
        - period, il periodo in quale si svolge l'Erasmus
        - agreedCourses, i corsi previsti dall'agreement
            - courseName
            - courseCode
            - ects, i crediti del corso
            - status, indica se è stato superato o meno
            - grade, voto di superamento
            - honor, la lode
            - completionDate, data di superamento
        - agreedCredits, i crediti previsti dall'agreement
        - completedCredits, i crediti effettivamente ottenuti
    - languageCertificates, eventuali certificati linguistici acquisiti
        - language, la lingua in questione
        - level, il livello della certificazione
        - certification, il nome della certificazione (es. IELTS)

        - certificationScore, il punteggio della certificazione
    - otherActivities, eventuali attività extra svolte (workshop, seminari…)
        - title, nome dell'attività
        - provider, chi ha fornito l'attività
        - hours, durata dell'attività
        - completionDate, data di completamento dell'attività

Esempio

Di seguito un esempio in formato JSON delle credenziale

```
{
 "credentialID": "cred-1234567890",
 "issuer": {
   "name": "Université de Rennes",
   "code": "UNIRENNESO1"
 },
 "emissionDate": "2025-01-01T00:00:00Z",
 "valid": true,
 "studentInfo": {
   "name": "Mario",
   "surname": "Rossi",
   "studentId": "123456",
   "birthdate": "2002-02-28",
   "nationality": "IT",
   "email": "mario.rossi@studenti.unisa.it",
   "degreeCourse": "Corso di Laurea Magistrale in Ingegneria Informatica",
   "courseDuration": "2 anni",
   "homeUniversity": {
     "name": "Università degli Studi di Salerno",
     "code": "UNISALERN001",
     "country": "Italia"
   }
```

```
},
"erasmusInfo": {
  "hostUniversity": {
    "name": "Université de Rennes",
    "code": "UNIRENNES01",
    "country": "France"
  },
  "erasmusStartDate": "2025-09-01",
  "erasmusEndDate": "2026-02-01",
  "LearningAgreement": {
    "period": "2025/2026 - Primo semestre",
    "agreedCourses": [
      {
        "courseName": "Machine Learning",
        "courseCode": "ML123",
        "ects": 9,
        "status": "superato",
        "grade": "27",
        "honor": false,
        "completionDate": "2025-10-20"
      },
      {
        "courseName": "Automation",
        "courseCode": "AU456",
        "ects": 9,
        "status": "superato",
        "grade": "30",
        "honor": true,
```

```
        "completionDate": "2025-11-22"
      }
    ],
    "agreedCredits": 18,
    "completedCredits": 18
  },
  "languageCertificates": [
    {
      "language": "Inglese",
      "Level": "B2",
      "certification": "IELTS",
      "certificationScore": 8
    }
  ],
  "otherActivities": [
    {
      "title": "Workshop sur Blockchain",
      "provider": "Université de Rennes",
      "hours": 10,
      "completionDate": "2023-10-10"
    }
  ]
 }
}
```

2. WP2

2.1. Tecnologie Abilitanti

Per concretizzare la visione che il gruppo ha per il sistema di gestione delle credenziali accademiche, è stata pensata un architettura decentralizzata in modo da ridurre al minimo le dipendenze delle singole autorità centrali. Avere un sistema bene distribuito ci permette di non concentrarlo in un unico punto vulnerabile potenziando la robustezza e la resilienza del sistema. Il sistema viene reso meno suscettibile a guasti e/o attacchi mirati e garantisce la privacy dello studente, tenendo sempre sotto controllo i diversi dati sensibili. La decentralizzazione, inoltre, spalanca le porte e permette alle università e alle istituzioni di dialogare superando le barriere dei sistemi tradizionali e spesso isolati. Per dar vita a questa architettura ci affidiamo ad un insieme di tecnologie crittografiche e standard che giocano un ruolo importante per il sistema che vogliamo realizzare:

Distributed Ledger Technologies (DLT) e Blockchain Pubblica

Le Distributed Ledger Technologies (DLT) costituiscono la base per la decentralizzazione del sistema. Queste tecnologie garantiscono un registro immutabile, trasparente e resistente, proprietà fondamentali per la gestione affidabile delle credenziali nel contesto distribuito in cui vengono usate. Il sistema farà uso di una blockchain pubblica, selezionata in base alla sua capacità di offrire un ambiente robusto e sicuro per l'esecuzione di Smart Contract. In particolare, la blockchain utilizzata opererà come un registro di revoca decentralizzato e pubblico, oltre ad ospitare le informazioni di identità decentralizzate, i DID Document (come descritto nella sezione successiva). L'uso di tale blockchain ci permette di massimizzare la trasparenza e la resilienza dei dati. Quando una credenziale dovrà essere invalidata, il suo identificativo univoco verrà registrato sulla blockchain tramite lo Smart Contract appositamente progettato. Questo registro dovrà risultare permanentemente accessibile e verificabile da chiunque, eliminando la necessità di interrogare direttamente le università emittenti per controllare ogni volta la validità delle credenziali.

Certificato rilasciato dall'Ente di Accreditamento

Questi certificati sono dei token auto-contenuti e firmati digitalmente che attestano che un determinato DID è stato riconosciuto e accreditato dall'Authority. La loro struttura segue lo standard JWT e include i seguenti campi chiave nel loro payload:

- "sub" (subject): Indica il DID dell'entità che è stata accreditata.
- "iss" (issuer): Specifica il DID dell'Accreditation Authority stessa, che ha emesso e firmato il certificato..
- "iat" (issued at): Un timestamp numerico che indica il momento in cui il certificato è stato emesso.
- "exp" (expiration time): Un timestamp numerico che definisce la data e l'ora di scadenza del certificato.
- "type": "AccreditationCertificate", che identifica lo scopo di questo JWT.

Gli header del JWT indicano l'algoritmo di firma utilizzato, RS256 e il tipo di token, JWT appunto. Il certificato viene firmato digitalmente dalla chiave privata dell'Accreditation Authority. Questa firma è fondamentale perché consente a qualsiasi parte di verificare l'autenticità del certificato utilizzando la chiave pubblica dell'Authority (la cui validità è risolvibile tramite il DID dell'Authority sul DID Registry). In questo modo, il certificato di accreditamento fornisce una prova verificabile e non

ripudiabile che l'università associata a quel DID è stata riconosciuta come un'entità legittima e fidata all'interno dell'ecosistema.

Gestione delle Chiavi e Infrastruttura di Fiducia

Per garantire l'autenticità e la fiducia tra gli attori in un contesto decentralizzato, il sistema deve superare i limiti delle PKI tradizionali basate su una singola Certificate Authority (CA) centralizzata, la quale potrebbe portare ad un singolo punto di fallimento (SPoF). Infatti le informazioni di identità e le chiavi pubbliche degli attori (sotto forma di DID Document) saranno registrate su uno Smart Contract ospitato sulla blockchain, il quale fungerà da registro pubblico, immutabile e distribuito delle identità, eliminando la dipendenza da una CA centralizzata. Gli Enti di Accreditamento svolgeranno un ruolo cruciale nella fase iniziale e avranno il compito di verificare l'identità legale e di registrare e autenticare i DID (e i relativi DID Document) delle università sul registro decentralizzato. Questo processo garantisce che solo enti riconoscitori e fidati possano emettere credenziali all'interno del sistema. Questo meccanismo permette anche a qualsiasi verificatore di interrogare la blockchain per ottenere in modo sicuro e verificabile la chiave pubblica dell'università. Tutto ciò consente di garantire l'autenticità della firma senza la necessità di intermediari centralizzati o contatti diretti tra le università.

W3C Verifiable Credentials (VCs)

Il cuore del sistema di certificazione accademica è basato sulle Verifiable Credentials (VCs), un modello di standardizzazione del World Wide Web Consortium (W3C). Le VCs rappresentano un paradigma per l'emissione, la presentazione e la verifica di attestazioni digitali. In questo modo il sistema potrà permettere ad un Emittente (Università Ospitante) di rilasciare credenziali ad un Titolare (Studente) che può poi presentarle ad un Verificatore (Università di Origine). Le VCs garantiscono che le informazioni siano autentiche e rafforzano significativamente l'affidabilità delle attestazioni accademiche. La loro adozione garantisce non solo la validità crittografica, ma anche la loro interoperabilità. La divulgazione selettiva, pur non essendo una specifica crittografica del modello dati W3C VC, è una capacità fondamentale supportata dall'architettura delle VCs per promuovere la privacy.

Decentralized Identifiers (DID) e DID Methods

Per elevare la robustezza e l'interoperabilità della soluzione, verranno integrati i Decentralized Identifiers (DIDs), uno standard del W3C per la gestione dell'identità digitale decentralizzata. I DIDs consistono in identificatori globalmente unici e persistenti che consentono agli attori di avere un controllo diretto e autonomo sulla propria identità digitale. Ogni DID è associato ad un DID Document (DID Doc), ossia un documento JSON pubblico che contiene informazioni essenziali per interagire con l'identità, incluse le chiavi pubbliche crittografiche utilizzate per la firma e la cifratura dal proprietario del DID. L'interazione con i DIDs avviene tramite un DID Method, che definisce le regole specifiche per la creazione e l'aggiornamento dei DIDs su una specifica infrastruttura decentralizzata (in questo caso la blockchain). L'adozione dei DID permette di standardizzare meglio e migliorare la gestione delle identità rispetto ad un semplice registro di chiavi pubbliche.

Struttura del DID Document

La sua struttura è standardizzata per garantire interoperabilità tra i diversi sistemi che adottano il paradigma dei DID. I campi principali del DID Document sono:

- @context: Specifica il contesto JSON-LD, che definisce i termini e le loro interpretazioni all'interno del documento. Questo campo assicura che il DID Document sia interpretabile correttamente secondo gli standard W3C.
- id: Questo campo contiene il DID stesso, un URI che identifica univocamente l'entità a cui il DID Document si riferisce (es. did:example:123456789abcdefghi).
- verificationMethod: Una lista di metodi di verifica, che tipicamente sono coppie di chiavi crittografiche (pubbliche) o altri meccanismi utilizzabili per autenticare e verificare firme. Ogni verificationMethod include:
    - id: Un URI che identifica univocamente il metodo di verifica all'interno del DID Document.
    - type: Il tipo di metodo crittografico (es. RsaVerificationKey2018).
    - controller: Il DID dell'entità che controlla questo metodo di verifica.
    - Il materiale della chiave pubblica, spesso codificato in formati standard come publicKeyPem (PEM encoding).
- authentication: Una lista di riferimenti ai verificationMethod che possono essere utilizzati per autenticare il controllore del DID. Questo permette di verificare che chiunque agisca per conto del DID sia effettivamente il suo controllore.
- assertionMethod: Una lista di riferimenti ai verificationMethod che possono essere usati per creare asserzioni o firme legate al DID (come la firma di credenziali o presentazioni).
- created: indica quando il Document è stato creato.

DApp

Le DApp (Applicazioni Decentralizzate) rappresentano l'interfaccia utente attraverso la quale gli attori interagiscono con il sistema. A differenza delle applicazioni tradizionali, le DApp eseguono la loro logica su una blockchain fornendo maggiore controllo all'utente e riducendo le dipendenze da server centralizzati. In questo contesto, gli studenti interagiscono con una DApp che fungerà da wallet digitale e permetterà loro di:

- Ricevere e conservare in modo sicuro le credenziali emesse dall'università.
- Gestire e visualizzare le credenziali accademiche digitali
- Generare le Presentazioni Verificabili (tramite divulgazione selettiva) e firmarle prima di inviarle ai verificatori.
- Interagire con lo smart contract di revoca per verificare lo stato delle credenziali ricevute.

L'utilizzo della DApp garantisce che gli studenti mantengano il pieno controllo delle proprie credenziali e dei propri dati privati.

Crittografia Ibrida

Questa combinazione di crittografia simmetrica e asimmetrica ci permette di garantire la confidenzialità delle informazioni sensibili durante la trasmissione. Facendo uso di un algoritmo AES (Advanced Encryption Standard) con modalità CTR (Counter) che trasforma il cifrario a blocchi in un cifrario a flusso, la cifratura di grandi volumi di dati risulta molto efficiente per lo scambio di informazioni riguardanti esami e attività e permette l'esecuzione in parallelo. Per proteggere la chiave simmetrica temporanea, generata ad ogni sessione di comunicazione tramite un algoritmo di generazione delle chiavi (Gen_{TempK}), verrà impiegato RSA OAEP (RSA Optimal Asymmetric Encryption Padding), il quale offre una sicurezza in più contro gli attacchi e garantisce che solo il destinatario previsto possa accedere alle informazioni cifrate.

Firme Digitali

Le firme digitali garantiscono le proprietà di integrità, autenticazione e sono state preferite alla soluzione della HMAC proprio per garantire anche la proprietà di non ripudio. In questo modo ogni credenziale e ogni presentazione selettiva viene sigillata digitalmente. Ciò garantisce di rilevare immediatamente un possibile tentativo di manomissione e che l'emittente o il presentatore non possano negare di aver compiuto una determinata azione. Verrà usato SHA-256 (Secure Hash Algorithm 256) per generare l'impronta digitale, o digest, dei dati delle credenziali e delle presentazioni selettive. La sua proprietà di resistenza alle collisioni garantisce che qualsiasi minima alterazione dei dati venga rilevata, fungendo da base per le firme digitali e per l'assicurazione dell'integrità.

Formato JWT e Merkle Tree

Per strutturare le credenziali accademiche, utilizzeremo il formato JWT (JSON Web Token), un formato standard per le credenziali accademiche che risulta compatto, facilmente trasportabile e, soprattutto, progettato per essere firmato digitalmente. Questa scelta ci permette di migliorare anche l'interoperabilità, consentendo a sistemi diversi di leggere e comprendere le credenziali senza ambiguità. Per implementare la divulgazione selettiva il payload del JWT non conterrà gli attributi in chiaro, ma la radice (root) di un Merkle Tree. Questo sarà costruito su tutti gli attributi dettagliati della credenziale e permetterà una prova crittografica ed efficiente della validità degli attributi selezionati senza rivelare l'intero set di dati. Quando uno studente desidererà di presentare solo un sottoinsieme di informazioni, genererà un Merkle Proof per gli attributi specifici richiesti. Questa prova, insieme agli attributi rilevati ed a un riferimento al JWT originale sarà inclusa in una Verifiable Presentation che verrà firmata digitalmente dallo studente stesso. Questo consentirà al verificatore di ricalcolare la Merkle Root e confrontarla con quella presente nel JWT originale verificando l'autenticità degli attributi rivelati senza dover esporre l'intera credenziale.

Protocolli di Sicurezza a livello di Trasporto

Per garantire la sicurezza delle comunicazioni lungo i canali di rete, il sistema si basa sull'utilizzo di protocolli standard come TLS (Transport Layer Security), implementato tramite HTTPS. Questo fornisce un livello fondamentale di confidenzialità, integrità e autenticazione reciproca del canale tra i diversi attori proteggendo i dati in transito da intercettazioni e manipolazioni a livello di rete.

Architettura Riassuntiva

In sintesi, questa architettura decentralizzata prevede che:

- Le università ospitanti emettono credenziali digitali firmate.
- Gli studenti le custodiscono con sicurezza graize alla DApp.
- Le università di origine possono riceverle e valutarle.

Il registro delle identità decentralizzate (DIDs) basato su smart contract e la blockchain agiscono come servizi globali e decentralizzati, garantendo la disponibilità e la tracciabilità in ogni fase del processo.

2.2. Algoritmi Utilizzati

La robustezza e l'affidabilità del sistema di gestione delle credenziali accademiche dipendono dall'adozione di specifici algoritmi che sono stati scelti per garantire le proprietà di sicurezza e privacy definite nel modello:

Algoritmo di Generazione delle Chiavi Asimmetriche (Gen_K)

Questo processo è fondamentale per la creazione delle coppie di chiavi private e pubbliche che identificano e proteggono sia le Università che gli Studenti. La sua sicurezza è garantita dall'utilizzo di un Generatore di Numeri Pseudo Casuali Crittograficamente Sicuro (CSPRNG), essenziale per produrre chiavi imprevedibili e resistenti a tentativi di indovinamento.

Algoritmo di Generazione delle Chiavi Simmetriche (Gen_{TempK})

Per ogni sessione di comunicazione che richiede cifratura, questo algoritmo produce chiavi simmetriche temporanee e uniche. Anche in questo caso , l'affidabilità si basa sull'impiego di un CSPRNG per assicurare la massima casualità e unicità di ogni chiave generata al volo.

Algoritmo di Generazione del Nonce/Numero di Sequenza

Per prevenire gli attacchi di replay sulla presentazione delle credenziali, il sistema genera un valore unico e imprevedibile, noto come nonce (Number Used Once). Anche questo algoritmo si avvale di un CSPRNG per produrre un nonce che viene incluso nella presentazione e la cui unicità è verificata dal destinatario.

Algoritmo di Hash SHA-256

Questo algoritmo di hash crittografico genera un'impronta digitale univoca e compatta di qualsiasi input. Il suo utilizzo risulta cruciale per diverse funzioni:

- La costruzione della Merkle Root
- La generazione e la verifica delle Merkle Proofs
- La preparazione dei JWT prima della firma.

La sua resistenza alle collisioni è vitale per garantire l'integrità dei dati di tutto il sistema.

Algoritmo di Calcolo del Merkle Tree

Questo processo algoritmico descrive la costruzione della struttura dati ad albero degli hash. Partendo dagli hash dei singoli attributi della credenziale (i nodi foglia), l'algoritmo combina ricorsivamente le coppie di hash dei nodi figli per generare l'hash del nodo genitore, proseguendo fino ad ottenere un unico hash finale, ossia la Merkle Root. Questa radice rappresenta un impronta digitale compatta e crittograficamente legata all'intero set di attributi della credenziale.

Algoritmo di Calcolo del Merkle Proofs

Questo processo è il cuore della divulgazione selettiva. Per ogni attributo che uno Studente decide di rivelare, l'algoritmo calcola una prova crittografica. Questa prova, unita all'attributo stesso, consente a chi verifica di confermare in modo efficiente l'appartenenza dell'attributo al set di dati originale e la sua consistenza con la Merkle Root, senza richiedere la rivelazione di tutte le altre informazioni.

Algoritmo di Verifica delle Merkle Proofs

Complementare all'algoritmo di calcolo, questo processo permette ad un verificatore di convalidare la correttezza di un attributo e della sua Merkle Proof. Utilizzando l'attributo rivelato, la Merkle Proof fornita e la Merkle Root originale, firmata dall'Università Emittente, l'algoritmo ricostruisce e confronta la radice, verificando crittograficamente che l'attributo sia parte della credenziale originale e garantendo integrità e autenticità nella divulgazione selettiva.

Algoritmo RSA

Questo algoritmo asimmetrico è fondamentale per la firma digitale e la verifica della firma digitale. Le università lo usano per firmare le credenziali emesse, mentre gli Studenti lo impiegano per firmare le presentazioni. Ciò garantisce l'autenticazione dell'origine, l'integrità del dato firmato e il non-ripudio dell'azione. RSA viene usato anche per la cifratura asimmetrica delle chiavi di sessione.

Algoritmo RSA OAEP (Optimal Asymmetric Encryption Padding)

Questo schema di padding in combinazione con l'algoritmo RSA migliora significativamente la sicurezza di RSA contro specifici attacchi, assicurando che la chiave simmetrica di sessione sia protetta in modo robusto durante la trasmissione e possa essere decifrata solo dal destinatario.

Algoritmo AES (Advanced Encryption Standard) in Modalità CTR (Counter)

Questo algoritmo di cifratura simmetrica è impiegato per la cifratura e decifratura efficiente di grandi volumi di dati sensibili, in particolare il payload delle presentazioni verificabili. La modalità CTR trasforma AES in un cifrario a flusso, consentendo un'esecuzione parallela e un'elevata velocità, elementi chiave per garantire confidenzialità delle informazioni scambiate.

2.3. Architettura delle Credenziali Digitali

Il sistema gestisce tre tipologie principali di strutture dati, ciascuna con un ruolo specifico.

Credenziale Accademica Verificabile (VC)

Questa struttura rappresenta l'attestazione ufficiale e firmata rilasciata dall'Università Ospitante. Il suo scopo è fornire una prova crittografica dell'esistenza e dell'integrità dell'intero set di dati accademici dello studente, senza però rivelarli direttamente. E' un token JWT.

- Header: specifica i metadati crittografici
    - alg: l'algoritmo di firma utilizzato (RS256).
    - typ: il tipo di token, JWT.
- Payload: contiene le claims della credenziale
    - jti: un ID univoco per la credenziale, che garantisce che ogni attestazione sia distinta.
    - iss: l'identità dell'Università che rilascia la credenziale (issuer), rappresentata dal suo identificativo decentralizzato (DID).
    - sub: l'identità dello studente (subject), anch'essa rappresentata da un DID.
    - acc: il DID dell'Ente che ha accreditato l'università
    - iat: data di emissione della credenziale.
    - exp: data di scadenza della credenziale.
    - type: un identificatore che specifica il tipo di credenziale, nel nostro caso è ErasmusCredential.
    - credentialStatus: un riferimento al registro decentralizzato per controllare se la credenziale è stata invalidata.
        - id: endpoint della revocation list.
        - type: il tipo di revocation list.
    - merkleRoot: l'impronta digitale crittografica di tutto il set dettagliato dei dati accademici. Questo campo è cruciale in quanto riassume in modo univoco e non ripudiabile le informazioni, fungendo da base per la divulgazione selettiva.
- Signature: la firma digitale. Viene ottenuta codificando in Base64url separatamente Header e Payload, che vengono poi unite: signingInput = encodedHeader + "." + encodedPayload. Su questo signingInput viene applicata la firma con RS256.

Ruolo nel Sistema: Questa struttura è il perno della fiducia. Garantisce l'autenticità dell'emittente, l'integrità dei dati attestati e la gestione della revocabilità. La Merkle Root è il fondamento tecnico che abilita la privacy dello studente.

Dati Accademici Dettagliati dello Studente

Questa è la struttura dati completa e granulare di tutte le informazioni accademiche. E' un semplice set di dati grezzi in formato JSON. Non è firmata, ma viene consegnata in modo confidenziale allo studente, che ne diventa l'unico custode. È da questo set di dati che viene calcolata la Merkle Root inserita nella VC.

- studentInfo, relativo alle informazioni anagrafiche dello studente
    - name
    - surname
    - studentId, la matricola
    - birthdate

        - nationality
        - email
        - degreeCourse, il corso di laurea
        - courseDuration, la durata del corso di laurea
        - homeUniversity, l'università di appartenenza
            - name
            - code
            - country
- erasmusInfo, contiene le informazioni che riguardano strettamente il periodo Erasmus dello studente
    - hostUniversity, l'università ospitante
        - name
        - code
        - country
    - erasmusStartDate, l'inizio del periodo di Erasmus
    - erasmusEndDate, la fine del periodo di Erasmus
    - learningAgreement, contiene le info relative al learning agreement stipulato
        - period, il periodo in quale si svolge l'Erasmus
        - agreedCourses, i corsi previsti dall'agreement
            - courseName
            - courseCode
            - ects, i crediti del corso
            - status, indica se è stato superato o meno
            - grade, voto di superamento
            - honor, la lode
            - completionDate, data di superamento
        - agreedCredits, i crediti previsti dall'agreement
        - completedCredits, i crediti effettivamente ottenuti
    - languageCertificates, eventuali certificati linguistici acquisiti
        - language, la lingua in questione
        - level, il livello della certificazione
        - certification, il nome della certificazione (es. IELTS)
        - certificationScore, il punteggio della certificazione
    - otherActivities, eventuali attività extra svolte (workshop, seminari…)
        - title, nome dell'attività
        - provider, chi ha fornito l'attività
        - hours, durata dell'attività
        - completionDate, data di completamento dell'attività

Ruolo nel Sistema: La sua granularità è ciò che rende possibile la divulgazione selettiva, permettendo allo studente di scegliere con precisione quali informazioni condividere.

Presentazione Accademica Verificabile (VP)

Questa struttura è un JWT, creata e firmata dallo studente al momento della condivisione. Contiene solo gli attributi scelti e le "prove" crittografiche (Merkle Proofs) che li collegano alla Credenziale Accademica Verificabile originale.

- Header:

    - alg: l'algoritmo di firma (RS256).
    - typ: tipo di token, JWT.
- Payload:
    - jti: l'ID univoco per la presentazione.
    - iss: il DID dello studente che crea e firma la presentazione.
    - aud: (Audience) il DID del Verificatore, a cui la presentazione è destinata.
    - iat: data di emissione della presentazione.
    - exp: data di scadenza della presentazione.
    - nonce: un valore univoco, per prevenire replay attack.
    - verifiableCredential, ovvero la VC originale
    - studentData: il sottoinsieme di dati che lo studente ha scelto di condividere.
        - name
        - surname
        - ecc…
    - merkleProofs: i dati che, uniti agli hash delle informazioni divulgate, permettono al verificatore di ricalcolare la Merkle Root originale e confermare così la validità dei dati presentati.
- Signature: la firma digitale, creata e apposta dallo studente utilizzando la sua chiave privata. Il calcolo tecnico è identico a quello della VC: si codificano in Base64url l'Header e il Payload della VP, si uniscono con un punto (signingInput = encodedHeader + "." + encodedPayload), e su questo signingInput viene applicata la firma con l'algoritmo RS256.

Ruolo nel Sistema: Questa struttura è il veicolo della divulgazione selettiva. Permette una verifica da parte di terzi senza la necessità di interrogare l'emittente originale e senza esporre dati non necessari, garantendo al contempo autenticità e non ripudio della presentazione stessa tramite la firma dello studente.

2.4. Processo di Rilascio delle Credenziali

Il rilascio delle credenziali accademiche digitali può essere visto come il punto di partenza del sistema che si intende realizzare. Consiste in un'operazione critica che stabilisce la fiducia e l'autenticità di ogni attestazione. Questo processo descrive le azioni intraprese dall'Università Ospitante, agendo come Emittente, per creare e consegnare una credenziale digitale verificabile ad uno studente che ne diventa il titolare. Questo processo può essere diviso nei seguenti passaggi:

Certificazione dell'Università Ospitante

Innanzitutto, qualsiasi università che intenda rilasciare o verificare credenziali, deve essere riconosciuta da tutte le parti del sistema come affidabile. Per permettere ciò, l'università inoltra ad un Ente di Accreditamento riconosciuto la richiesta di essere certificata come entità fidata. L'Ente, una volta effettuate le verifiche necessarie accredita l'università rilasciando un certificato JWT. Esso contiene il DID dell'università, dell'Ente, informazioni su quando è stato rilasciato, la data di scadenza e il tipo di certificato. Viene firmato tramite RS256.

Richiesta di Emissione e Verifica Preliminare

Il processo vero e proprio inizia quando uno studente completa il suo programma Erasmus, accompagnato dal superamento di uno o più esami, o il ricevimento di un qualche tipo di attestazione. In seguito avanzerà una richiesta formale all'Università Ospitante per l'ottenimento delle credenziali. Prima di procedere con la sua emissione, tale Università si occuperà di svolgere una rigorosa verifica interna per confermare l'accuratezza e la legittimità di tutte le informazioni accademiche dello studente. Solo dopo questa fase di validazione interna si procede con la digitalizzazione e l'emissione della credenziale.

Preparazione e Strutturazione della Credenziale (Payload JWT)

Il processo continua con la raccolta e l'organizzazione strutturata delle diverse informazioni pertinenti e convalidate. Per fare ciò, si fa uso di un JWT (JSON Web Token), che funge da "contenitore" standardizzato per i dati della credenziale. Il payload del JWT verrà poi attentamente compilato, in linea con ciò che è stato definito nel Capitolo precedente. Il JWT non conterrà direttamente gli attributi sensibili in chiaro, ma sarà invece calcolata la radice del Merkle Tree, costituito su tutti gli attributi della credenziale e sarà inserita nel payload. Poi insieme al JWT verranno mandati anche gli attributi per permettere alla DApp dello Studente di creare la finestra per la selezione dei dati. Questo passaggio è fondamentale per il funzionamento della divulgazione selettiva e per il processo di conservazione (descritti nei capitoli successivi).

Generazione della Firma Digitale

Dopo aver strutturato la credenziale (con la Merkle Root nel payload), il processo prevede di garantire l'autenticità e l'integrità della credenziale con l'aggiunta della Firma Digitale dell'Università Ospitante sull'intero JWT. Questo processo avviene in più fasi in modo da massimizzare la sicurezza:

- Calcolo dell'Hash (o digest): L'Università Ospitante calcola un Hash crittografico dell'intero JWT, includendo sia l'header che il payload con tutti i dati della credenziale, utilizzando una funzione di hash sicura come SHA-256. Questo produce un'impronta digitale unica e compatta della credenziale che permette di controllare la più minima alterazione dei dati.
- Applicazione della Firma: Il digest viene poi firmato utilizzando la chiave privata di firma dell'Università Ospitante. Per questa operazione, si impiega RSA. Questi schemi di firma

incorporano casualità e padding, offrendo una maggiore sicurezza contro la falsificazione e garantendo che la firma sia univocamente legata alla credenziale e all'emittente.

- Assemblaggio definitivo della Credenziale JWT: La firma digitale prodotta viene quindi allegata al JWT, formando la credenziale accademica digitale completa. Tipicamente il JWT prevede una struttura compatta e standardizzata composta da tre parti separate da punto (header.payload.signature) che rendono la credenziale pronta per la trasmissione.

Consegna Sicura della Credenziale

Il processo termina con la credenziale JWT firmata e completa che viene consegnata in modo sicuro allo studente. Il canale di consegna deve garantire la confidenzialità e l'integrità durante il trasporto, anche se il JWT è già firmato, avvalendosi di protocolli di sicurezza a livello di trasporto come TLS/HTTPS. La credenziale verrà recapitata al wallet digitale dello studente, che la riceverà e la conserverà in modo sicuro. La ricezione e la conservazione della credenziale nel wallet dello studente segna il completamento del processo di emissione e trasferimento della credenziale.

2.5. Processo di Conservazione delle Credenziali

Questa fase inizia nel momento in cui lo studente riceve la Credenziale Accademica Verificabile e il set completo dei suoi dati accademici dettagliati dall'Università Ospitante. Il wallet digitale dello studente (DApp) svolge un ruolo cruciale in questo processo, garantendo l'integrità e la fiducia nei dati ricevuti. Tale processo può essere diviso nelle seguenti fasi:

Consegna Sicura della Credenziale

L'Università Ospitante invia allo studente due componenti essenziali:

- Il Verifiable Credential JWT firmato
- Il set completo dei dati accademici dello studente (in formato JSON).

Accesso alla DApp

Per permettere l'accesso alla DApp solo ed esclusivamente allo studente, entità autorizzata ad usufruirne, è richiesto l'inserimento dei campi email e password. Ai fini dimostrativi dell'esecuzione del progetto, si presuppone che lo studente abbia ricevuto email e password antecedentemente; queste credenziali sono consultabili nel file credential.json

Ricezione e Verifica della VC

Una volta che il pacchetto di dati viene ricevuto nel wallet digitale dello studente, il primo passo è la validazione della Verifiable Credential. Lo studente innanzitutto verifica che chi ha mandato la VC sia un'università accreditata, munita di certificato valido rilasciato da un Ente di Accreditamento, dopodichè procede alla verifica della firma digitale dell'Università Ospitante apposta sul VC JWT assicurando che la credenziale provenga effettivamente dall'emittente legittimo e che non sia stata manomessa in alcun modo durante il transito.

Ricalcolo e Confronto del Merkle Root

Subito dopo la verifica della firma del VC JWT, lo studente esegue un controllo cruciale per l'integrità dei dati completi ricevuti. Partendo dal set dettagliato dei dati accademici, il wallet ricalcola la Merkle Root e la confronta con la Merkle Root presente nel payload del VC JWT che è stato firmato dall'Università Ospitante. Questo passaggio è vitale, in quanto se le due Merkle Root non corrispondo, significa che i dati dettagliati ricevuti non sono gli stessi che l'Università ha attestato.

Accettazione e Archiviazione Sicura

Solo se la firma del VC JWT è valida e la Merkle Root ricalcolata corrisponde a quella nel payload, la credenziale viene considerata valida e integra dal titolare. A questo punto, il wallet dello studente archivia la credenziale in modo sicuro con meccanismi di cifratura per proteggere la privacy dello studente anche in fase di conservazione. Questo processo di verifica attiva da parte dello studente è fondamentale per assicurarsi che i dati completi in suo possesso siano esattamente quelli attestati e firmati dall'Università, rafforzando la fiducia nell'intero sistema decentralizzato e dando allo studente il pieno controllo sui propri dati accademici.

2.6. Processo di Divulgazione Selettiva

Il processo di divulgazione selettiva è il fulcro della garanzia di privacy dello Studente, consentendogli di esercitare un controllo granulare sulle proprie informazioni accademiche. Questa funzionalità permette al Titolare di rivelare solo un sottoinsieme degli attributi contenuti in una Verifiable Credential, senza dover esporre l'intera credenziale, reso possibile grazie all'uso dei Merkle Tree, come descritto già in precedenza. Il processo può essere diviso nelle seguenti fasi:

Selezione degli Attributi

Lo Studente, attraverso l'interfaccia intuitiva della sua DApp, ha la facoltà di visionare tutti gli attributi dettagliati della propria VC e, a seconda delle specifiche richieste dal Verificatore (Università di Origine) o a propria descrizione, lo Studente seleziona con precisione quali informazioni desidera condividere. Questa fase è cruciale, perché solo gli attributi esplicitamente scelti verranno inclusi nella Presentazione Verificabile.

Calcolo delle Merkle Proof

Una volta che lo Studente ha selezionato gli attributi da divulgare, la DApp calcola automaticamente una Merkle Proof per ciascuno degli attributi scelti. Una Merkle Proof consiste in una prova crittografica che dimostra l'inclusione di un determinato attributo all'interno del Merkle Tree originale della credenziale. Questo processo si basa sugli algoritmi di Calcolo delle Merkle Proofs e dell'Algoritmo di Hash SHA-256.

Generazione della Verifiable Presentation (VP) per la Divulgazione

Le informazioni selezionate dallo Studente, insieme alle relative Merkle Proofs e ad un riferimento alla VC originale, vengono aggregate in una nuova struttura, ossia la Verifiable Presentation, anch'essa serializzata in formato JWT, e contiene esclusivamente gli attributi scelti e le sue prove crittografiche, oltre all'identificativo del Titolare (Studente) sotto forma di DID. La VP include anche un nonce generato in modo casuale e unico attraverso l'algoritmo apposito in modo da prevenire attacchi di replay.

Firma della Verifiable Presentation (VP)

Per garantire l'autenticità e l'integrità della VP, lo Studente firma digitalmente l'intera Verifiable Presentation. Questo processo utilizza la chiave privata di firma dello Studente, associata al suo DID, e l'algoritmo RSA per generare la firma digitale. In questo modo il sigillo crittografico attesta che la VP proviene legittimamente dal titolare della credenziale e che le informazioni non sono state alterate dallo Studente dopo la selezione e la firma.

2.7. Processo di Presentazione delle Credenziali

Il processo di presentazione descrive come uno Studente dimostri le proprie qualifiche all'Università di Origine ed eserciti il proprio controllo sui propri dati accademici. Il flusso di processo viene progettato in modo efficiente, sicuro e rispettoso della privacy attraverso diverse fasi:

Inizio della Presentazione

Il ciclo di presentazione inizia quando lo studente decide autonomamente di presentare le proprie credenziali per segnalare il completamento del programma Erasmus.
L'obiettivo, per l'Università di Origine, è ottenere una prova verificabile e affidabile delle qualifiche accademiche dello studente.

Preparazione della Presentazione

Lo studente decide di presentare le proprie credenziali e utilizzando l'interfaccia della sua DApp ha la capacità di selezionare in modo granulare solo le informazioni specifiche che desidera condividere, esercitando la funzionalità di divulgazione selettiva (descritta nel Capitolo precedente "Processo di Divulgazione Selettiva"). Le informazioni selezionate, insieme alla relativa prova crittografica (Merkle Proof) e ad un riferimento alla credenziale originale, vengono strutturate in un nuovo oggetto, accompagnato da nonce, Verifiable Presentation, che consiste in un JWT contenente esclusivamente le informazioni scelte e l'identificativo del Titolare (Studente) sotto forma di DID.

Firma dello Studente

Per garantire l'autenticità e l'integrità di questa presentazione selettiva, lo studente firma digitalmente questo nuovo oggetto dati utilizzando la propria chiave privata di firma, associata al suo DID. Questo sigillo crittografico permette di attestare che la presentazione proviene legittimamente dal titolare della credenziale e che le informazioni divulgate non sono state alterate dallo studente dopo la selezione.

Presentazione all'Università di Origine

L'oggetto dati firmato dallo studente viene quindi trasmesso all'Università di Origine e sebbene l'integrità, l'autenticità e il non ripudio siano garantiti dalla firma digitale, per garantire anche la massima confidenzialità del canale di trasmissione, la presentazione verrà ulteriormente cifrata utilizzando la crittografia ibrida definita nel sistema. In questo caso, l'intero oggetto firmato dallo studente verrebbe cifrato con una chiave simmetrica, che a sua volta viene cifrata con la chiave pubblica dell'Università di Origine. Questo assicura che solo quest'ultima possa accedere al contenuto della presentazione.

2.8. Processo di Verifica delle Credenziali

Il processo di verifica è la fase in cui l'Università di Origine accerta la validità e l'autenticità delle credenziali presentate dallo studente, senza dover interrogare direttamente l'Università Ospitante per ogni verifica. Questo processo prevede le seguenti fasi:

Ricezione e Decifratura della Presentazione

L'Università di Origine riceve il pacchetto di presentazione dello studente. Il primo passo è decifrare il pacchetto e quindi l'Università utilizza la propria chiave privata di crittografia per decifrare la chiave simmetrica temporanea. Successivamente, impiega questa chiave per decifrare il testo cifrato usando l'algoritmo AES. Questo processo ricostruisce l'oggetto dati originale (la VP), contenente le informazioni che lo studente ha scelto di divulgare.

Verifica dello stato di Revoca

L'Università di Origine esegue un controllo dello stato di revoca della credenziale originale. Utilizzando il riferimento alla credenziale originale, l'Università di Origine interroga la blockchain pubblica. Questo processo, descritto nel dettaglio nel Capitolo successivo "Processo di Revoca", permette di determinare se la credenziale sia stata invalidata dopo la sua emissione. Se la credenziale risulta invalidata e quindi presente nella lista di revoca, la presentazione viene immediatamente rifiutata. Questo garantisce che solo credenziali attualmente valide vengano accettate.

Verifica del Nonce

Successivamente, l'università di origine estrae il nonce in essa contenuto e attraverso un registro persistente dei nonce già elaborati controlla se è stato già ricevuto. Se già presente, la presentazione viene rifiutata, in quanto ciò indicherebbe un tentativo di replay attack. Se il nonce è nuovo, viene aggiunto al registro e il processo di verifica prosegue. Questo meccanismo è cruciale per prevenire che un attaccante possa riutilizzare una VP valida, anche se firmata correttamente, per ottenere un riconoscimento multiplo o altre azioni non autorizzate.

Verifica della Firma Digitale dello Studente

Una volta decifrato, l'Università di Origine procede alla verifica della presentazione selettiva. Il primo controllo è sulla firma digitale dello studente utilizzando la chiave pubblica di firma di quest'ultimo, ottenuta risolvendo il DID dello Studente tramite il DID Resolver che interroga la blockchain pubblica. L'università verifica che la firma sull'oggetto dati selettivo sia valida. Questo passo è cruciale per confermare che la presentazione proviene effettivamente dallo studente che la sta presentando e che le informazioni non siano state alterate in seguito alla selezione e alla firma dello studente.

Verifica della Firma Digitale dell'Emittente Originale e del Certificato

Dopo aver autenticato lo studente, l'Università di Origine deve ora accertare l'autenticità e l'integrità delle informazioni accademiche e la loro provenienza dall'emittente originale. La Verifiable Presentation contiene:

- Gli attributi scelti dallo studente
- Le Merkle Proof corrispondenti a tali attributi
- Un riferimento alla credenziale originale

- L’identificativo dell'Università Ospitante, sotto forma di DID dell’Emittente.

L’Università di Origine utilizza tale DID per risolvere il DID Document e ottenere la chiave pubblica di firma dell’Università Ospitante dalla blockchain. Dopo di che vengono verificati la validità della firma digitale originale posta dall’Università Ospitante, e la presenza e la validità del certificato rilasciato dall’Ente di Accreditamento. L’Università di Origine ricalcola poi la Merkle Root a partire dagli attributi divulgati e dalle Merkle Proof. Se quella calcolata corrisponde a quella contenuta nel payload del JWT originale, allora gli attributi specifici divulgati sono autentici e fanno parte della credenziale originale. Questa fase conferma che le informazioni accademiche divulgate sono autentiche e non sono state manomesse dal momento della loro emissione. Tutto ciò permette di creare un ponte di fiducia tra le due diverse Università senza che queste debbano contattarsi in modo diretto per ogni verifica.

Validazione e Riconoscimento Finale

Solo dopo che tutte le diverse verifiche hanno avuto esito positivo, l’Università di Origine considera la credenziale presentata valida e affidabile. A questo punto può procedere al riconoscimento ufficiale delle informazioni fornite, snellendo significativamente i processi burocratici e facilitando la mobilità internazionale.

2.9. Processo di Revoca

Il processo di revoca garantisce l'integrità e l'affidabilità continua delle credenziali accademiche digitali nel tempo, consentendo all'Università Ospitante di invalidare, in caso di necessità, una credenziale emessa. Questo meccanismo è fondamentale per la fiducia nel sistema, poiché impedisce l'uso di credenziali obsolete o non più valide. Inoltre il processo di revoca avviene in modo decentralizzato e permanente tramite la blockchain pubblica. Può essere diviso nelle seguenti fasi:

Richiesta di Revoca

Il processo di Revoca ha inizio quando l'Università Ospitante ha la necessità di invalidare una credenziale precedentemente emessa. Questa decisione può derivare da varie motivazioni, come la rettifica di errori, comportamenti scorretti o altre condizioni simili. Questa decisione di revoca prevede un processo interno all'Università, che culmina nell'identificazione della credenziale da invalidare. Una volta identificata la credenziale da revocare, l'Università Ospitante prepara una transazione di revoca contenente l'identificativo univoco della credenziale da invalidare. Tale identificativo deve necessariamente essere lo stesso utilizzato nel payload della VC originale.

Firma e Invio alla Blockchain

L'Università Ospitante firma digitalmente la transazione di revoca utilizzando la propria chiave privata, associata al suo DID. Questa firma garantisce che la richiesta di revoca provenga dall'Emittente legittimo della credenziale e non da terze parti non autorizzate. Una volta firmata, la transazione viene inviata allo Smart Contract di revoca ospitato sulla blockchain pubblica.

Registrazione Immutabile sulla Blockchain

Quando la transazione di revoca raggiunge lo Smart Contract, esso convalida la firma e la provenienza dell'Università Ospitante. Successivamente, lo Smart Contract registra l'identificativo univoco della credenziale in un registro immutabile sulla blockchain. Questo atto rende la revoca permanente e irreversibile. L'identificativo della credenziale viene così aggiunto ad una "lista di Revoca" decentralizzata e consultabile pubblicamente.

Verifica dello Stato di Revoca

Qualsiasi Verificatore, nel momento in cui riceve una Verifiable Presentation (VP) da uno Studente, include tra i suoi passaggi di verifica un controllo dello stato di revoca della credenziale originale. Utilizzando l'identificativo della credenziale originale e il DID dell'Emittente forniti nella VP, il verificatore interroga direttamente lo Smart Contract di revoca sulla blockchain pubblica. Se l'identificativo della credenziale è presente nel registro di revoca, la credenziale viene considerata non più valida e la presentazione viene rifiutata. Questo passaggio finale assicura che solo credenziali attive e valide siano accettate dal sistema.

3. WP3

3.1. Contesto per l'Analisi di Sicurezza

Nel presente Work Package 3 verrà eseguita un'analisi della sicurezza sulla soluzione per la gestione delle credenziali accademiche (descritta nel WP2) rispetto al modello del sistema (descritto nel WP1). Per fornire un contesto chiaro all'analisi che seguirà, risulta utile riassumere sinteticamente gli elementi chiave stabiliti nel WP1. In questo modo ricordiamo gli attori coinvolti (Studente Università di origine/Ospitante, Enti di Accreditamento), le funzionalità principali richieste (emissione, conservazione, presentazione, divulgazione selettiva, verifica, revoca), le minacce identificate nel threat model (dallo studente malevole al verificatore malintenzionato) e le proprietà di sicurezza e privacy che il sistema deve garantire (Confidenzialità, Integrità, Decentralizzazione, Privacy dello Studente, ecc.). Questa ricapitolazione serve da base per valutare come le scelte progettuali della nostra soluzione affrontano efficacemente le vulnerabilità e soddisfano i requisiti di resilienza.

3.2. Analisi degli Attacchi

Una volta definita l'architettura del sistema e i suoi processi operativi nel WP2, è cruciale esaminare come la soluzione si comporti di fronte agli attacchi potenziali identificati nel threat model del WP1. Di seguito verrà analizzata sinteticamente le diverse tipologie di attaccanti, valutando le loro capacità e gli obiettivi in relazione alla nostra progettazione. Per ogni scenario di attacco, verrà descritta l'azione dell'attaccante e come il nostro meccanismo, basato su DLT, DIDs, VCs, Merkle Trees e Crittografia Avanzata, mitiga o previene con successo la compromissione del sistema.

Studente Malevolo

Lo studente malevolo, come già definito nel paragrafo "Attori" nel WP1, consiste in un attore interno che dispone di credenziali legittime e tenta di modificarle a fini illeciti o per ottenere vantaggi illegittimi (aggiunta di esami non sostenuti, o modifica dei voti). La nostra soluzione è stata progettata per contrastare efficacemente questi tentativi di manomissione grazie a diversi meccanismi e processi di verifica:

- Integrità della Credenziale Accademica Verificabile (VC): la VC è un JWT la cui integrità è garantita da una firma digitale apposta dall'Università stessa. Qualsiasi alterazione invaliderebbe la firma rendendo la manomissione palese durante il processo di verifica.
- Validazione tramite Merkle Tree e Merkle Proofs: Le informazioni accademiche dettagliate sono strutturate in un Merkle Tree, la cui radice è inclusa e firmata all'interno del JWT della VC. Se lo studente modifica i dati dettagliati in suo possesso, durante il ricalcolo della Merkle Root questa non corrisponderà a quella firmata, rendendo rilevabile l'alterazione.
- Meccanismo di Revoca: In scenari in cui una falsificazione venga scoperta successivamente all'emissione, l'Università Ospitante può revocare la credenziale sulla Blockchain, invalidandone l'uso futuro.

In conclusione il sistema mitiga efficacemente gli attacchi di questo tipo, garantendo validità e autenticità delle credenziali.

Eavesdropper

L'Eavesdropper consiste in un attore passivo che si limita ad osservare ed intercettare le comunicazioni tra gli attori del sistema e ottenere dati sensibili (credenziali complete o informazioni sul traffico di rete). La nostra soluzione è stata progettata per garantire la confidenzialità dei dati tramite l'impiego di protocolli e meccanismi crittografici robusti:

- Protocolli di Sicurezza a livello di Trasporto (TLS/HTTPS): tutte le comunicazioni tra gli attori avvengono su canali protetti da TLS/HTTPS. L'utilizzo di questi protocolli permettono di garantire la cifratura del traffico a livello di trasporto, rendendo estremamente difficile l'intercettazione e lettura dei dati sensibili.
- Crittografia Ibrida per la Presentazione Verificabile: La confidenzialità del payload della Presentazione Verificabile (VP) è ulteriormente protetta con l'utilizzo di un algoritmo di cifratura simmetrica con chiave temporanea. Tale chiave viene a sua volta cifrata con la chiave pubblica del destinatario. In questo modo solo il destinatario può accedere agli attributi rilevanti, anche se il canale di trasporto fosse compromesso ad un livello inferiore.
- Merkle Tree e Divulgazione Selettiva: La natura stessa della divulgazione selettiva contribuisce alla privacy. Anche se l'attaccante riuscisse a dedurre la presenza di una comunicazione, non sarebbe in grado di accedere a tutte le informazioni contenute nella credenziale completa.

Pertanto, gli attacchi di tipo eavesdropping sono efficacemente contrastati, salvaguardando la privacy degli utenti.

Man in the Middle

Il Man in the Middle consiste in un attore attivo che si intromette nella comunicazione tra due parti, potendo intercettare, modificare o immettere informazioni in transito. Una sua caratteristica specifica è il Replay Attack, consistente nel ripetere l'invio di un messaggio utilizzando una presentazione legittima precedentemente registrata. La nostra soluzione proposta permette di contrastare queste tipologie di attacchi utilizzando una combinazione di firme digitali, integrità dei dati e meccanismi anti-replay:

- Firme Digitali: La Credenziale Verificabile, come già detto, è firmata digitalmente dall'emittente utilizzando la sua chiave privata. Qualsiasi alterazione del contenuto della VC invaliderebbe la firma, permettendo la rilevazione della manomissione. Analogamente avviene lo stesso per la Presentazione Verificabile.
- Merkle Tree e Merkle Proofs: La strutturazione dei dati in Merkle Tree e l'uso di Merkle Proofs assicurano che ogni singolo attributo presentato sia crittograficamente legato alla merkle root originale. Il ricalcolo della Merkle Root da parte del verificatore rileverebbe molto facilmente l'alterazione o la manomissione dei dati sensibili.
- Crittografia Ibrida: Come già detto per l'eavesdropper, la cifratura ibrida realizzata per la comunicazione tra Studente ed Università impedisce al Man in the Middle di leggere o estrarre informazioni sensibili dal payload.
- Nonce per Prevenire Replay Attack: Ogni Presentazione Verificabile (VP) include un valore unico e imprevedibile generato per ogni sessione. L'università di Origine mantiene un registro dei nonce già elaborati. Se si riceve una VP con un nonce già visto, la presentazione viene rifiutata impedendo ad un attaccante di registrare una VP valida e ripresentarla per ottenere informazioni.

Di conseguenza, l'attacco Man in the Middle viene correttamente mitigato tramite l'autenticazione e l'integrità fornite dalla soluzione proposta.

Attaccante DoS

L'Attaccante DoS mira a rendere non disponibili o inaccessibili i servizi del sisteme inviando un numero elevato di richieste per causare rallentamenti, sovraccarichi o interruzioni. La mitigazione a questo attacco è principalmente data dalla sua architettura decentralizzata e dalla distribuzione dei carichi di verifica:

- Decentralizzazione della Blockchain Pubblica: I servizi più critici come il registro delle revoche e il registro delle identità centralizzate sono ospitati su una blockchain pubblica, un sistema distribuito e altamente resistente agli attacchi DoS diretti ad un singolo nodo o server, in quanto non esiste un singolo punto di fallimento.
- Verifica Decentralizzata: Il processo di verifica delle credenziali non richiede un interrogazione diretta e continua all'Università emittente. Le università che fanno da destinatario possono autonomamente verificare la validità delle firme e lo stato di revoca interrogando la blockchain pubblica.
- Efficienza degli algoritmi: Le scelte degli algoritmi efficienti per la cifratura e la generazione e verifica di Merkle Proofs e firme digitali riducono il carico computazionale per ogni operazione rendendo il sistema più resistente ad un elevato numero di richieste.

Tutte queste caratteristiche permettono di rendere il sistema resiliente a tentativi DoS, mantenendone la disponibilità.

Impersonificatore

L'Impersonificatore consiste in un attore che tenta di assumere l'identità di un attore legittimo per svolgere azioni non autorizzate. Questa tipologia di attacco si basa sul furto o sulla compromissione delle credenziali di autenticazione. La soluzione proposta garantisce protezione da questo attacco tramite la forte associazione tra le identità digitali decentralizzate (DID) e le coppie di chiavi crittografiche, abbinata a processi di verifica rigorosi:

- Decentralized Identifiers e controllo della chiave privata: Ogni attore del sistema possiede un proprio DID unico e controlla la chiave privata ad esso associata. Senza il possesso della chiave, un impersonificare non può generare firme valide e, di conseguenza, non può spacciarsi per l'attore legittimo.
- Verifica della Firma Digitale basata su DID: Ogni volta che una VC o una VP viene presentata, il processo di verifica include la convalida della firma digitale. La chiave pubblica viene recuperata risolvendo il loro DID sulla blockchain pubblica. Se un impersonificare tenta di firmare con una chiave non valida o non associata al DID dichiarato, la verifica fallisce.
- Nonce: Anche se un impersonificatore riuscisse in qualche modo ad ottenere una VP firmata in modo valido, il meccanismo del nonce ne impedisce il riutilizzo. Ogni VP valida può essere usata una sola volta per la verifica, rendendo inutile la ripresentazione.

In questo modo l'impersonificazione viene efficacemente mitigata.

Verificatore Malintenzionato

Il verificatore malintenzionato consiste in un attore autorizzato a ricevere credenziali che, però, tenta di abusare di tali informazioni per violare la privacy o effettuare tracciamenti. Questo può includere tentativi di forzare la divulgazione di un numero eccessivo di attributi o di tracciare e profilare lo studente basandosi sulle prestazioni. La protezione da questa tipologia di attacco consiste in un pilastro fondamentale del sistema e viene garantito tramite una combinazione di processi e meccanismi appositi:

- Divulgazione Selettiva: Questo è il meccanismo che permette di proteggere la privacy. Lo studente detiene il controllo su quali informazioni desidera condividere. Le presentazioni verificabili (VP) presentano solo gli attributi scelti dallo studente e le relative Merkle Proofs. Non c'è alcun modo per il verificatore di derivare o richiedere l'intero set di dati in quanto mai esposti nella VP.
- Assenza di un Registro centrale delle presentazioni: Il sistema non prevede un database centrale dove tutte le presentazioni sono registrate ed indicizzate. Questo impedisce al verificatore di correlare facilmente diverse presentazioni di uno stesso studente nel tempo per scopi di profilazione non autorizzate.
- DApp come Wallet di Controllo: La DApp agisce come intermediario di fiducia per lo studente, permettendogli di visualizzare, selezionare e gestire le proprie credenziali in un ambiente controllato.

Pertanto, il rischio legato a questo attacco viene contenuto e monitorato.

3.3. Analisi di Soddisfacimento delle Proprietà

Dopo aver analizzato la resistenza del sistema agli attacchi specifici, risulta utile valutare anche il soddisfacimento delle proprietà desiderate delineate nel WP1. Per ciascuna proprietà chiave verrà esaminato come le scelte architetturali e gli algoritmi implementati contribuiscono a garantirne la realizzazione. L'obiettivo è dimostrare che la soluzione non solo mitiga le minacce, ma opera in modo conforme ai principi di sicurezza e privacy predefiniti.

Confidenzialità

La confidenzialità dei dati, ossia la garanzia che le informazioni siano accessibili solo a soggetti autorizzati, è assicurata da diversi livelli presenti nella soluzione proposta:

- Crittografia Ibrida: Quando si vogliono presentare delle informazioni sensibili, l'intero payload viene cifrato usando una chiave simmetrica che a sua volta viene cifrata con la chiave pubblica del destinatario. Tutto ciò permette la decifrazione solo al destinatario proteggendo da qualsiasi intercettazione non autorizzata durante il transito.
- Protocollo di Trasporto Sicuri (TLS/HTTPS): tutte le comunicazioni avvengono su canali protetti fornendo un robusto strato di cifratura a livello di trasporto.
- Divulgazione Selettiva: La confidenzialità è rafforzata dal principio di divulgazione selettiva. Lo studente decide quali attributi desidera condividere e nel farlo l'intera credenziale o informazioni non pertinenti non vengono mai esposte al verificatore.
- Conservazione Locale: I dati accademici vengono salvati dallo studente grazie alla DApp minimizzando la fuga di dati e massimizzando il controllo diretto dell'utente.

Sotto Proprietà della Confidenzialità

- C.1: I dati completi delle credenziali accademiche devono essere accessibili solo allo studente che ne è titolare e, in modo controllato, alle parti autorizzate (università) che richiedono una specifica divulgazione.
    - Questo è garantito dall'approccio basato su Merkle Tree per la struttura delle credenziali e dalla crittografia ibrida. La credenziale completa è conservata cifrata sul dispositivo dello studente. Solo lo studente, in possesso della chiave privata, può decifrarla. Per la condivisione con enti autorizzati viene utilizzato il meccanismo di divulgazione selettiva che consente allo studente di esporre solo gli attributi specifici richiesti, senza rivelare l'intera credenziale.
- C.2: Le comunicazioni tra lo studente e le università devono essere protette da possibili intercettazioni non autorizzate.
    - La protezione delle comunicazioni è assicurata tramite l'uso della crittografia ibrida. Quando dati sensibili vengono trasmessi, l'intero payload è cifrato usando una chiave simmetrica, che a sua volta è cifrata con la chiave pubblica del destinatario. Questo garantisce che solo il destinatario previsto possa decifrare e leggere il contenuto, rendendo le intercettazioni inefficaci.
- C.3: Il meccanismo di divulgazione selettiva deve impedire la rivelazione di dati non richiesti, anche se la credenziale completa contiene tali informazioni.
    - L'implementazione della divulgazione selettiva tramite i Merkle Proofs soddisfa questa sotto proprietà. Lo studente costruisce una prova crittografica che include solo gli attributi selezionati, senza rivelare la struttura completa del Merkle Tree o gli altri

attributi non scelti. Questo assicura che solo le informazioni esplicitamente desiderate siano esposte.

- C.4: Un verificatore malintenzionato non deve poter forzare la divulgazione di attributi aggiuntivi rispetto a quelli esplicitamente scelti dallo studente.
    - Il controllo sulla divulgazione selettiva rimane completamente nelle mani dello studente.

Integrità

L'integrità dei dati, ovvero la garanzia che le informazioni non siano state alterate in modo non autorizzato, è un pilastro del sistema, implementata tramite firme digitali e strutture dati crittografiche:

- Firme Digitali: Le credenziali generate sono firmate digitalmente. La combinazione di algoritmi di firma robusti come RSA e funzioni di hash come SHA-256 permette di creare un'impronta digitale unica del contenuto..
- Merkle Tree e Merkle Proofs: Questa struttura dati è fondamentale per garantire integrità con l'utilizzo della divulgazione selettiva. Quando lo studente presenta un sottoinsieme degli attributi, vengono generati i merkle proofs di questi ultimi e vengono inviati insieme al merkle tree originale. Una volta arrivati a destinazione la Merkle Root viene ricalcolata e confrontata con quella originale. Se non corrispondono, significa che i dati sono stati alterati o che non appartengono alla stessa credenziale originale.
- Blockchain: La natura immutabile della blockchain in cui vengono registrati i DID e le transizioni di revoca garantisce l'integrità di queste informazioni fondamentali. Una volta registrato sulla blockchain il dato non può essere modificato o cancellato senza essere rilevato dall'intera rete.

Sotto Proprietà dell'Integrità

- I.1: Le credenziali accademiche, una volta emesse e firmate digitalmente dall'Università Ospitante, non devono poter essere falsificate, manomesse o alterate.
    - Questo è garantito dall'uso di firme digitali RSA apposte dall'Università Ospitante sulla Merkle Root della credenziale. Qualsiasi tentativo di alterazione della credenziale invaliderà la firma digitale, rendendo l'alterazione immediatamente rilevabile.
- I.2: Qualsiasi alterazione avvenuta sulla credenziale deve essere rivelabile dal verificatore.
    - Grazie alla firma digitale dell'Università Ospitante sulla Merkle Root, il verificatore, al momento della ricezione di una presentazione, può ricalcolare la Merkle Root dagli attributi presentati e confrontarla con quella firmata. Qualsiasi discordanza indica un'alterazione e invalida la credenziale.
- I.3: Le informazioni presentate in modo selettivo dallo studente devono mantenere la loro integrità e non devono poter essere alterate in transito.
    - Le Merkle Proofs, utilizzate per la divulgazione selettiva, sono intrinsecamente robuste contro le alterazioni. La prova stessa contiene gli hash necessari per ricostruire la Merkle Root originale. Se le informazioni presentate o la prova stessa vengono alterate durante la trasmissione, il processo di verifica della Merkle Proof fallirà, rivelando la manomissione.
- I.4: Il processo di verifica deve confermare l'integrità delle informazioni presentate rispetto alle credenziali originali firmate.

    - Il verificatore, utilizzando la Merkle Root firmata dall'Università Ospitante e la Merkle Proof fornita dallo studente, può matematicamente convalidare che gli attributi presentati corrispondono a quelli originali e firmati, senza dover accedere all'intera credenziale. Questo processo crittografico garantisce l'integrità.

Autenticazione

L'autenticazione, la capacità di verificare l'identità di un utente, è centrale per stabilire la fiducia nel sistema e si basa sull'infrastruttura di identità decentralizzate:

- DID e Chiavi Crittografiche: Ogni attore del sistema possiede un proprio DID unico e controlla la corrispondenza con le coppie di chiavi crittografiche. L'autenticazione è implicitamente realizzata attraverso la prova di possesso della chiave privata usata per la firma.
- Trust Root degli Enti di Accreditamento: Gli Enti di Accreditamento svolgono un ruolo cruciale per la registrazione dei DIDs delle Università sulla blockchain dopo averne verificato l'identità legale. Questo stabilisce una catena di fiducia che permette alle Università di Origine di fidarsi dei DID e delle chiavi pubbliche associate alle Università Ospitante.

Sotto Proprietà dell'Autenticazione

- A.1: Solo gli utenti legittimi e autenticati possono accedere ed interagire con i servizi offerti dal sistema.
    - L'accesso e l'interazione con i servizi sono mediati dall'uso di Decentralized Identifiers e dalle relative chiavi crittografiche. Ogni attore possiede un DID e una coppia di chiavi privata/pubblica. L'autenticazione avviene tramite la prova di possesso della chiave privata associata al DID, garantendo che solo l'entità legittima possa agire per conto del proprio DID.
- A.2: L'identità dello studente che presenta le credenziali deve essere verificabile e non impersonificabile.
    - Lo studente firma digitalmente la sua presentazione utilizzando la chiave privata associata al suo DID. Il verificatore può quindi convalidare questa firma usando la chiave pubblica dello studente, risolta tramite il suo DID sulla blockchain. Questo processo garantisce l'autenticità dello studente e previene l'impersonificazione.
- A.3: L'identità dell'università deve essere verificabile durante il processo di verifica delle credenziali, assicurando che l'emittente sia chi dichiara di essere.
    - Allo stesso modo, l'Università Ospitante firma le credenziali con la propria chiave privata associata al suo DID. Il verificatore, tramite la risoluzione del DID dell'università, può recuperare la sua chiave pubblica e verificare la firma della credenziale, garantendo l'autenticità dell'emittente.
- A.4: Le chiavi delle università e degli enti devono essere autenticate e gestite in modo affidabile per prevenire possibili attacchi.
    - La gestione delle chiavi è strettamente legata alla gestione dei DID. I DID e le relative chiavi pubbliche sono registrati sulla blockchain, che ne garantisce l'immutabilità e la disponibilità. Le chiavi private sono mantenute in modo sicuro dalle rispettive entità.

Non-Ripudio

La proprietà di non-ripudio garantisce che un'entità che ha eseguito un azione non possa successivamente negare di averla compiuta. Tale proprietà è intrinsecamente garantita dall'uso delle firme digitali nel sistema:

- Firme digitali: Le firme digitali sono un attestazione crittografica dell'autenticità e provenienza del messaggio trasmesso. Poiché la chiave privata è controllata solo dal mittente, e la firma è verificabile da chiunque abbia la sua chiave pubblica, chi ha trasmesso il messaggio non può disconoscere la sua emissione.
- Immutabilità della Blockchain: Le registrazioni dei DID e degli stati di revoca sulla blockchain pubblica sono immutabili. Questo significa che sono permanenti e non possono essere alterati o ripudiati una volta registrati.

Sotto Proprietà del Non-Ripudio

- NR.1: L'università Ospitante non può negare di aver rilasciato una credenziale valida.
    - L'Università Ospitante firma digitalmente la Merkle Root della credenziale con la propria chiave privata. Questa firma è crittograficamente legata all'identità dell'università (tramite il suo DID) e alla credenziale stessa. Poiché la validità della firma è verificabile pubblicamente e l'identità dell'università è risolvibile tramite il suo DID registrato su blockchain, l'università non può disconoscere l'emissione.
- NR.2: Lo studente non può negare di aver presentato un sottoinsieme specifico di informazioni dalla sua credenziale in modo da evitare che uno studente malevolo o un impersonificare possano rinnegare un'azione fraudolenta svolta.
    - Quando lo studente presenta un sottoinsieme di informazioni, firma digitalmente la presentazione con la sua chiave privata. Questa firma, insieme alla Merkle Proof che attesta l'autenticità degli attributi presentati rispetto alla credenziale originale, crea un legame crittografico innegabile tra lo studente (tramite il suo DID) e la specifica presentazione. Questo impedisce allo studente di negare l'azione.
- NR.3: Le azioni di revoca di una credenziale devono essere attribuibili all'entità che le ha eseguite.
    - Le azioni di revoca sono registrate sulla blockchain. Ogni operazione di revoca è associata al DID dell'entità che ha richiesto la revoca. Questo garantisce che l'azione di revoca sia tracciabile e attribuibile in modo inequivocabile all'entità responsabile.

Revocabilità

La revocabilità è la capacità di invalidare una credenziale precedentemente emessa, un aspetto cruciale per mantenere l'affidabilità del sistema in caso di errori o frodi. Il sistema implementa appositamente un meccanismo di revoca decentralizzato:

- Registro di Revoca su Blockchain: Il sistema prevede uno Smart Contract dedicato sulla blockchain pubblica che funge da registro di revoca immutabile.
- Processo di Richiesta di Revoca: L'emittente originale della credenziale è l'unica entità autorizzata a richiedere la revoca. Dopo un processo decisionale interno, l'emittente genera e firma digitalmente una transizione di revoca contenente l'identificativo univoco della credenziale da invalidare. La transizione viene poi inviata allo Smart Contract corrispondente.
- Registrazione immutabile: Una volta che la transizione di revoca è convalidata e registrata dallo Smart Contract, l'identificativo della credenziale viene aggiunto permanentemente alla

lista di revoca decentralizzata della blockchain. Questa registrazione è irreversibile e pubblicamente verificabile.

Sotto Proprietà della Revocabilità

- R.1: Le università emittenti devono avere la capacità di invalidare credenziali specifiche in caso di errore, scadenze o comportamenti illeciti.
    - Il sistema include un meccanismo per cui l'Università Ospitante, in qualità di emittente, può aggiornare lo stato di una credenziale a "revocata" su un registro di revoca decentralizzato. Questo processo consente di invalidare credenziali specifiche in base a criteri predefiniti o a discrezione dell'emittente.
- R.2: Lo stato di revoca di una credenziale deve essere pubblico, in modo da essere facilmente verificabile da qualsiasi parte interessata.
    - Il registro di revoca è implementato su una blockchain pubblica. Questo significa che qualsiasi verificatore o parte interessata può interrogare la blockchain per controllare lo stato di revoca di una credenziale specifica, garantendo trasparenza e accessibilità universale allo stato di validità.
- R.3: La revoca deve essere irreversibile e deve essere possibile aggiornare lo stato in tempo utile per prevenire l'uso di credenziali invalidate.
    - Una volta che una credenziale è registrata come revocata sulla blockchain, tale stato è immutabile e non può essere annullato, soddisfacendo il requisito di irreversibilità. L'aggiornamento dello stato avviene in tempo reale sulla blockchain, garantendo che le credenziali invalidate siano prontamente riconosciute come tali da tutti i verificatori.
- R.4: Il meccanismo di revoca non deve compromettere la privacy dello studente rivelando informazioni non pertinenti al di là dello stato di validità della credenziale.
    - Il registro di revoca sulla blockchain è progettato per contenere solo un identificativo univoco della credenziale e il suo stato di revoca. Non vengono esposte informazioni personali dello studente o dettagli della credenziale che non siano strettamente necessari per determinare la sua validità.

Trasparenza

La trasparenza si riferisce alla visibilità e alla comprensibilità del funzionamento del sistema per tutte le parti interessate. Questa viene garantita tramite standard aperti e un'infrastruttura pubblica:

- Standard Aperti: L'intero sistema è costruito su standard aperti e interoperabili definiti da organismi come W3C. Questo assicura che i protocolli e i formati siano pubblicamente documentati e ispezionabili facilitandone la comprensione da terze parti indipendenti.
- Blockchain Pubblica: L'uso della blockchain per il registro di revoca e il salvataggio dei DID fornisce un elevato grado di trasparenza. Tutte le transizioni e gli strati di questi registri sono pseudonimi ma pubblicamente consumabili, immutabili e verificabili. Questo permette a chiunque di ispezionare l'emissione e lo stato di validità delle credenziali.
- Assenza di autorità Centrali nascoste: La decentralizzazione del sistema elimina la necessità di intermediari centrali aumentando la trasparenza complessiva delle operazioni.

Sotto Proprietà della Trasparenza

- T.1: I meccanismi e i protocolli per l'emissione, la presentazione, le verifica e la revoca delle credenziali devono essere basati su standard aperti e pubblicamente noti, consentendo a tutti gli attori di comprendere e verificare il funzionamento del sistema.
    - Il sistema si basa sull'adozione di standard aperti e interoperabili come i Decentralized Identifiers, Verifiable Credentials, Merkle Trees e protocolli crittografici standard (RSA). L'uso di una blockchain pubblica per la gestione dei DID e dei registri di revoca rende i meccanismi sottostanti trasparenti e verificabili da chiunque.
- T.2: La validità e l'autenticità delle credenziali devono essere verificabili da terze parti senza richiedere l'interazione diretta dell'Università Ospitante per ogni verifica.
    - Grazie alla firma digitale dell'Università sulla credenziale e alla risoluzione dei DID tramite la blockchain, un verificatore può autonomamente convalidare l'autenticità e la validità di una credenziale presentata. Non è richiesta alcuna comunicazione diretta o sincrona con l'Università Ospitante per ogni verifica.
- T.3: Le procedure e le politiche relative alla gestione delle credenziali devono essere chiare, documentate e comprensibili a tutti gli attori.
    - Il sistema, basato su standard aperti e protocolli noti, facilita la creazione di procedure chiare e documentate. La natura intrinsecamente trasparente della blockchain contribuisce a rendere i processi di emissione e revoca comprensibili.

Privacy dello Studente

La privacy dello studente è un requisito fondamentale e un principio cardine del nostro design e viene assicurata da meccanismi specifici che pongono lo studente al centro del controllo dei propri dati:

- Divulgazione Selettiva: Questo è il meccanismo più potente per la proteione della privacy dello Studente. Invece di essere costretto a condividere l'intera credenziale, lo Studente può scegliere in modo granulare solo gli attributi specifici e necessari da rivelare.
- DID e Controllo Autonomo: I DIDs offrono allo Studente un identità digitale auto-sovrana in quanto non dipendente da provider di identità centralizzate che possono tracciare o monetizzare i suoi dati.
- Conservazione Locale: Le credenziali complete e i dati accademici dettagliati sono conservati direttamente e in forma cifrata nel wallet digitale/DApp dello studente. In questo modo si riduce di molto il rischio di esposizione a violazioni della privacy su larga scala.
- Assenza di un Registro Centrale delle Presentazioni: Il sistema non mantiene un database centrale che traccia tutte le presentazioni effettuate dallo studente. Ogni presentazione è un evento indipendente, impedendo la correlazione o la profilazione dello studente con interazioni precedenti.

Sotto Proprietà della Privacy dello Studente

- P.1: Lo studente deve avere il controllo granulare sulla divulgazione delle proprie informazioni accademiche, in modo da poter scegliere solo un sottoinsieme strettamente necessario di informazioni da condividere.
    - Questo è raggiunto attraverso l'implementazione della divulgazione selettiva basata sui Merkle Trees. Lo studente può specificare esattamente quali attributi della sua

credenziale desidera rivelare, generando una Merkle Proof che valida solo quegli attributi, mantenendo gli altri privati.

- P.2: La presentazione selettiva non deve rivelare implicitamente altri attributi non scelti dallo studente, proteggendo da tecniche di correlazione.
    - La natura delle Merkle Proofs assicura che la presentazione di un sottoinsieme di attributi non fornisca informazioni aggiuntive sugli attributi non scelti. Gli hash dei nodi non rivelati sono inclusi solo come elementi di prova e non possono essere decifrati per rivelare i dati sottostanti.
- P.3: Il sistema deve proteggere da tecniche di profilazione o fingerprinting basate sui dati divulgati.
    - La divulgazione selettiva limita la quantità di dati che un singolo verificatore può raccogliere, rendendo più difficile la profilazione completa dello studente.

Decentralizzazione

La decentralizzazione è un principio architetturale fondamentale che elimina la dipendenza da autorità centrali, aumentando la resistenza alla censura e l'autonomia degli attori:

- DLT e Blockchain Pubblica: L'architettura del sistema si basa sulla blockchain pubblica che per sua natura è decentralizzata e distribuita su una rete di nodi, eliminando un singolo punto di fallimento per i servizi critici come la risoluzione dei DID e la gestione della revoca.
- DID: I DIDs sono identificatori che non vengono emessi o controllati da autorità centrali. La loro registrazione e risoluzione avviene tramite blockchain garantendo che nessun singolo ente possa controllare o censurare le identità digitali degli attori.
- Wallet Digitale/DApp: Gli studenti gestiscono le proprie credenziali tramite un wallet digitale/DApp custodito nei propri dispositivi. Questo riduce la dipendenza da servizi di custodia centralizzati e consente agli studenti di avere il pieno controllo sui propri dati accademici.

Sotto Proprietà della Decentralizzazione

- D.1: Il sistema deve operare senza un singolo punto di fallimento o un'unica autorità centrale di controllo per l'emissione, la gestione e la verifica delle credenziali.
    - L'architettura del sistema si basa su DIDs e blockchain. Non esiste un'unica entità che controlla l'emissione, la gestione o la verifica. Le università emettono credenziali autonomamente, gli studenti le gestiscono localmente e i verificatori possono convalidare indipendentemente.
- D.2: La verifica delle credenziali non deve dipendere dall'interrogazione diretta e continua dell'Università Ospitante.
    - Come menzionato in T.2, la verifica si basa sulla firma digitale dell'Università Ospitante sulla credenziale e sulla risoluzione dei DID e dei registri di revoca sulla blockchain. Questo elimina la necessità di interrogare l'Università Ospitante per ogni verifica, rendendo il processo scalabile e robusto.
- D.3: Il sistema deve resistere ad attacchi mirati a singole entità, garantendo la disponibilità complessiva del servizio di verifica.
    - La natura distribuita della blockchain e la decentralizzazione delle operazioni significano che un attacco a una singola università o a un singolo verificatore non compromette l'intero sistema. Il servizio di verifica rimane disponibile finché la rete blockchain è operativa.

- D.4: La gestione delle chiavi crittografiche e delle identità deve essere distribuita tra i diversi attori coinvolti.
    - Ogni attore (università, studente, ente) gestisce autonomamente le proprie chiavi private associate al proprio DID. Le chiavi pubbliche e i DID sono registrati sulla blockchain, che è un registro distribuito, garantendo una gestione distribuita delle identità e delle chiavi.

Efficienza

L'efficienza si riferisce all'ottimizzazione delle risorse necessarie per il funzionamento del sistema. Il design del sistema integra diverse scelte che ne migliorano l'efficienza:

- Divulgazione Selettiva: Invece di trasmettere e verificare l'intera credenziale originale ogni volta, lo Studente invia solo un sottoinsieme degli attributi con le relative Merkle Proofs. Questo riduce significativamente la quantità di dati da trasmettere e il carico computazionale per la verifica rispetto a sistemi che richiederebbero l'accesso all'intera credenziale.
- Verifica Decentralizzata e asincrona: La capacità dei verificatori di interrogare direttamente la blockchain per la risoluzione dei DID e lo stato di revoca elimina la necessità di comunicazioni sincrone e potenzialmente lente ottimizzando i tempi di verifica.
- Riduzione della Burocrazia e dei Tempi di Processo: A livello di processo, il sistema snellisce significativamente le procedure amministrative tradizionali per la condivisione e la verifica delle credenziali (il quale era l'obiettivo del progetto).

Sotto Proprietà dell'Efficienza

- E.1: L'emissione delle credenziali deve essere efficiente in termini di tempo e risorse computazionali consentendo un rilascio rapido e scalabile.
    - Il processo di emissione che prevede la generazione di una Merkle Tree e la firma digitale da parte dell'Università, in combinazione con l'uso di formati standard (come JWT) per le credenziali contribuisce a un'emissione rapida.
- E.2: La conservazione delle credenziali sul dispositivo dello studente deve minimizzare il consumo di memoria, rendendo il sistema utilizzabile anche su dispositivi con risorse limitate.
    - Il formato JWT, combinato con l'organizzazione dei dati tramite Merkle Tree, consente una rappresentazione compatta delle credenziali. Questo minimizza lo spazio di archiviazione richiesto sul dispositivo dello studente.
- E.3: La presentazione selettiva delle credenziali deve essere rapida e deve minimizzare il traffico di rete.
    - La generazione di Merkle Proofs per la divulgazione selettiva è un'operazione computazionalmente leggera. La chiave è che vengono trasmesse solo le informazioni essenziali (gli attributi scelti e gli hash necessari per la prova), riducendo drasticamente il traffico di rete rispetto alla trasmissione dell'intera credenziale.
- E.4: La verifica delle credenziali ricevute deve essere veloce ed efficiente in termini di risorse computazionali.
    - La verifica di una Merkle Proof è un'operazione efficiente, che richiede un numero logaritmico di operazioni rispetto al numero totale di attributi. Inoltre, la risoluzione dei DID e la verifica dello stato di revoca tramite la blockchain sono processi ottimizzati per la velocità, eliminando la necessità di interazioni lente e sincrone con l'emittente.

Valutazione e Confronto

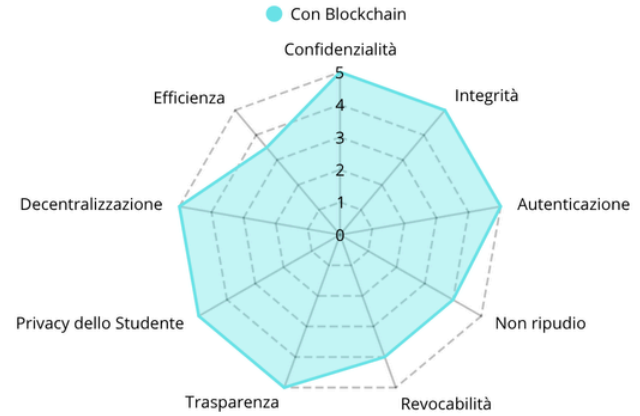
Per comprendere meglio l'impatto delle diverse tecnologie implementate nel sistema, è stata condotta una valutazione comparativa delle proprietà chiave tra la soluzione completa e una versione semplificata priva di blockchain. Questo confronto evidenzia come la rimozione di componenti critici influisca sul soddisfacimento delle caratteristiche fondamentali di sicurezza, privacy e decentralizzazione.

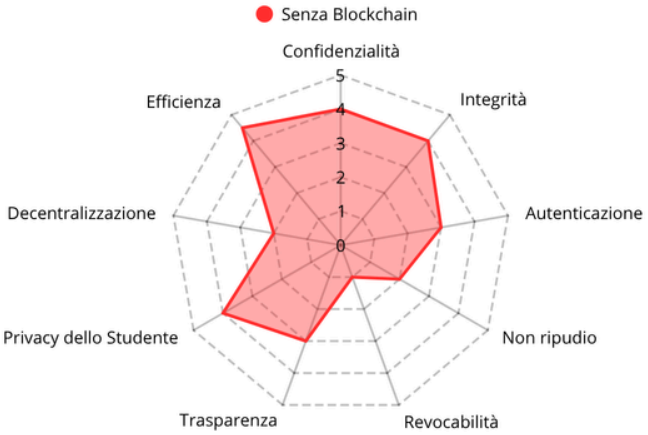
Proprietà	Valutazione (da 1 a 5)		Note
	Sistema Completo	Sistema Semplificato senza Blockchain	
Confidenzialità	5	4	La crittografia e la divulgazione selettiva sono garantite, ma senza blockchain la sicurezza cala.
Integrità	5	4	Firme digitali garantiscono l'integrità dei dati, senza blockchain manca immutabilità dei DID e delle revoche.
Autenticazione	5	3	DID esistono ma senza blockchain la fiducia nella catena di identità è più debole.
Non Ripudio	4	2	Blockchain garantisce non ripudio per l'emissione dei DID e la revoca.
Revocabilità	4	1	Registro di Revoca decentralizzato scompare, compromettendo affidabilità e trasparenza della revoca
Trasparenza	5	3	Blockchain offre trasparenza per DID e revoche, senza di essa la visibilità si riduce, ma non scompare del tutto.
Privacy dello Studente	5	4	Privacy protetta da divulgazione selettiva e wallet, ma senza blockchain è minore la garanzia di non tracciabilità.
Decentralizzazione	5	2	Senza blockchain, il sistema diventa più centralizzato.
Efficienza	3.5	4.5	Senza blockchain il sistema potrebbe essere più leggero e meno costoso, ma a discapito della sicurezza e decentralizzazione.

Dal confronto emerge chiaramente che il sistema completo, che integra la blockchain per la gestione dei DID e la revocabilità delle credenziali, offre un livello significativamente più elevato di sicurezza, decentralizzazione e trasparenza rispetto alla versione semplificata senza blockchain. Sebbene il sistema semplificato migliori leggermente l'efficienza e riduca la complessità operativa, questi vantaggi vengono pagati con una marcata perdita di affidabilità nelle funzioni chiave di autenticazione, revoca e non-ripudio. Pertanto, per scenari in cui la sicurezza, la fiducia distribuita e la tutela della privacy sono prioritarie, il sistema completo rappresenta la soluzione preferibile. La

versione semplificata potrebbe invece essere valutata in contesti con requisiti meno stringenti o per prototipi dove la semplicità e i costi ridotti sono più importanti.

Di seguito, sono mostrati i grafici a radar delle prestazioni delle due differenti soluzioni:





3.4. Criticità e Potenziali Miglioramenti

Sebbene la soluzione proposta sia stata progettata con un forte focus su sicurezza, privacy e decentralizzazione, è importante riconoscere che nessun sistema è immune da criticità o margini di miglioramento. Di seguito sono state identificati i potenziali limiti o le sfide che potrebbero emergere nell'implementazione o nell'adozione del sistema:

- Scalabilità della Blockchain: L'utilizzo di una blockchain pubblica può comportare costi di transazione variabili e potenzialmente elevati, soprattutto in scenari di alto volume come la registrazione di DID o la revoca delle credenziali. La scalabilità potrebbe inoltre diventare un fattore limitante in caso di un numero esponenziale di transazioni. Un possibile miglioramento quindi potrebbe essere l'adozione di una blockchain con architetture più scalabili e con costi di transazioni minori, ma mantenendo al contempo le proprietà di sicurezza e decentralizzazione.
- Gestione delle Chiavi Private da parte degli Utenti: Lo studente ha il pieno controllo sulla propria chiave privata e sebbene questo sia un vantaggio per la sua privacy, pone una grande responsabilità sull'utente. La perdita della chiave o una sua compromissione comporterebbe la perdita di accesso e controllo sulle proprie credenziali digitali. Una possibile soluzione è l'aggiunta si meccanismi user-friendly per il recupero delle chiavi che non vadano a compromettere la decentralizzazione.
- Complessità del sistema: L'uso combinato di DID, VC, Merkle Tree, crittografia ibrida e Blockchain introduce una complessità tecnologica significativa. Per enti meno digitalizzati o studenti non esperti, l'interazione con wallet, presentazioni verificabili o la gestione delle chiavi risulta difficile. Un possibile miglioramento consiste nella progettazione di interfacce utenti semplificate e sistemi assistiti per studenti e personale universitario.
- Problemi con la logica di Revoca: Sebbene il meccanismo di revoca avvenga sulla blockchain e in maniera decentralizzata, la decisione di revoca è un processo decisionale che avviene al di fuori di tutto ciò e si tratta di un processo interno all'università. Ciò significa che la fiducia ultima della tempestività e correttezza della revoca dipende dalla capacità di quest'ultima di agire correttamente. Un possibile miglioramento consiste nell'enfatizzare la necessità di policy chiare e procedure interne robuste per l'Università.

3.5. Conclusioni

L’obiettivo delle analisi svolte è stato valutare la robustezza del sistema rispetto alle principali minacce identificate, nonché verificare il grado di soddisfacimento delle proprietà di sicurezza e privacy richieste. Dall’analisi degli attaccanti è emersa una chiara capacità del sistema di contrastare efficacemente comportamenti malevoli, sia da parte di utenti interni, sia da attaccanti esterni. Infatti le tecnologie utilizzate forniscono una solo insieme di strumenti per garantire le proprietà identificate nel WP1. L’analisi del soddisfacimento delle proprietà ha mostrato come la soluzione riesca a garantire un elevato livello di confidenzialità, privacy, decentralizzazione e revocabilità pur evidenziando alcuni limiti in termini di efficienza e semplicità d’uso per utenti non esperti.
Il sistema si dimostra abbastanza scalabile e verificabile da mantenere un'elevata coerenza con i principi di trasparenza e fiducia distribuita. Infine, per quanto riguarda le criticità, sono stati individuati margini di ottimizzazione suggerendo possibili direzioni future per l’evoluzione del sistema. In conclusione la soluzione proposta rappresenta un modello sicuro e decentralizzato per la gestione delle credenziali accademiche digitali capace di tutelare lo studente e semplificare i processi burocratici tra le istituzioni. L’adozione del sistema porterebbe ad un aumento della fiducia, una riduzione dei tempi di verifica e un rafforzamento nell’interoperabilità tra Università.

4. WP4

4.1. Ambiente di sviluppo

Per l'implementazione di quanto descritto nel WP2 abbiamo realizzato un prototipo di esecuzione in Python, sull'IDE PyCharm. In virtù di ciò, funzionalità e parti del sistema quali la blockchain usata, il registro dei DID, il registro delle revoche, la parte di interazione con l'utente (DApp) sono implementati mediante classi Python. Per permettere la collaborazione e lo sviluppo di questo prototipo è stato creato un repository GitHub, accessibile mediante il link: https://github.com/Pepi2002/APS-Project.git .

4.2. Actors

Per simulare l'utilizzo del sistema da parte di entità quali università e studenti, sono state definite classi che ne implementassero le funzionalità principali e le interazioni con il sistema. Queste entità sono i nostri attori, ed estendono quindi il comportamento della classe Actor che è di seguito illustrata.

Actor

La classe Actor rappresenta un'entità generica che interagisce con identità decentralizzate (DID) e registri. Ogni attore è dotato di una coppia di chiavi crittografiche (pubblica e privata) per firmare e verificare messaggi, e di un DID che funge da identificatore unico nella blockchain. Questa classe gestisce la generazione e la gestione delle chiavi, la creazione del documento DID associato e l'interazione con i registri DID e di revoca. Le funzionalità principali sono:

- Al momento della creazione, un Actor genera automaticamente una coppia di chiavi RSA (privata e pubblica), un DID unico e un DID Document associato. Viene anche inizializzato un oggetto HybridCrypto per operazioni crittografiche ibride e vengono passati i riferimenti a un DIDRegistry e un RevocationRegistry per interagire con i rispettivi registri.
- Generazione chiavi:
    - generate_private_key: Crea una nuova chiave privata RSA con una dimensione di 2048 bit.
- Esportazione chiavi PEM (get_private_key_pem, get_public_key_pem): Queste funzioni permettono di ottenere la chiave privata e la chiave pubblica rispettivamente in formato PEM (Privacy-Enhanced Mail), un formato standard per la codifica di chiavi crittografiche.
- Gestione DID Document:
    - generate_did_document: Costruisce il DID Document, una struttura di dati che descrive il DID, le sue chiavi pubbliche e i relativi metodi di verifica, in conformità con lo standard W3C.

AccreditationAuthority

La classe AccreditationAuthority estende la classe base Actor, ereditando tutte le sue funzionalità relative alla gestione dei DID e alle operazioni crittografiche. Rappresenta un Ente in grado di generare e firmare certificati di accreditamento richiesti dalle università. La funzionalità principale è:

- Generazione di certificati di accreditamento (generate_accreditation_certificate): permette all'Autorità di Accreditamento di emettere un certificato JWT per un DID specifico.
    - Il certificato include un payload con informazioni essenziali: il DID del soggetto a cui è destinato il certificato (sub), il DID dell'autorità emittente (iss), l'ora di emissione (iat), l'ora di scadenza (exp) e il tipo di certificato (type).
    - Il JWT viene firmato digitalmente utilizzando la chiave privata dell'Autorità di Accreditamento e l'algoritmo RS256. Questa firma crittografica garantisce che chiunque possa verificare l'autenticità e l'integrità del certificato utilizzando la chiave pubblica dell'Autorità.

Issuer

Estende la classe base Actor. Un Issuer è un'università accreditata che emette credenziali verificabili, fungendo da fonte affidabile che attesta informazioni specifiche su un holder (titolare). Questa classe

gestisce quindi la creazione delle VC e la trasmissione sicura dei messaggi. Le funzionalità principali sono:

- Creazione di credenziali verificabili (create_verifiable_credential): Questa è la funzione centrale dell'Issuer. Permette di generare una Verifiable Credential (VC) firmata dall'Issuer stesso.
    - La funzione richiede il DID del titolare (holder_did), il DID dell'autorità di accreditamento (accreditation_did), la Merkle Root dei dati accademici (merkle_root) e la validità desiderata in giorni (exp_days). Il risultato è un JWT che rappresenta la VC.
- Trasmissione di messaggi (transmit): consente all'Issuer di inviare messaggi cifrati a un destinatario.
    - Utilizza il DID del destinatario per recuperare la sua chiave pubblica tramite il did_registry.
    - Il messaggio viene cifrato utilizzando la crittografia ibrida, combinando crittografia simmetrica e asimmetrica per efficienza e sicurezza
- Revoca della credenziale (revoke_credential): permette all'Issuer di revocare una Verifiable Credential precedentemente emessa.
    - Prende in input il JWT della Verifiable Credential.
    - Decodifica il JWT per estrarre l'ID univoco della credenziale.
    - Invia l'ID della credenziale al Revocation Registry, che si occupa di registrare l'evento di revoca sulla blockchain.

Student

Estende la classe base Actor. Uno studente è l'entità che riceve, verifica, archivia e presenta credenziali. Questa classe include quindi logiche per decifrare pacchetti di dati, validare la catena di fiducia delle credenziali, e creare presentazioni verificabili per condividere selettivamente le proprie informazioni. Le funzionalità principali sono:

- All'inizializzazione, uno studente, come un attore generico, genera la propria coppia di chiavi RSA, il proprio DID e il relativo DID Document. Oltre a ciò, riceve in input un’istanza di StudentDApp, che simula un'applicazione decentralizzata per la gestione delle credenziali dello studente.
- Decifrazione pacchetto (decrypting_package): decifra un pacchetto di dati crittografato ricevuto, che contiene gli attributi divulgati e il JWT della Credenziale Verificabile.
    - Utilizza la crittografia ibrida per decifrare il pacchetto con la chiave privata dello studente.
    - Estrae gli attributi divulgati (disclosed_attributes) e il JWT della Verifiable Credential dal payload decifrato.
- Controllo revoca (is_revoked): consulta il registro di revoca per vedere se una credenziale è stata revocata.
- Verifica del certificato di accreditamento (verify_accreditation_certificate): prima di fidarsi di una credenziale, lo studente verifica che l'issuer della VC sia stato accreditato da un'autorità riconosciuta.
    - Recupera il DID dell'issuer e dell'autorità di accreditamento dal payload della VC.
    - Ottiene la chiave pubblica dell'autorità di accreditamento dal DID Registry.
    - Recupera il certificato di accreditamento dell'issuer dal DID Registry.

    - Decodifica il certificato JWT usando la chiave pubblica dell'autorità e verifica che sia valido e che accrediti l'issuer corretto.
- Verifica della firma della credenziale (verify_credential_signature): verifica che la Verifiable Credential non sia stata manomessa e sia stata effettivamente firmata dall'issuer.
    - Decodifica il JWT della VC utilizzando la chiave pubblica dell'issuer. Se la firma non corrisponde, la verifica fallisce.
- Verifica della Merkle root dalla VC (verify_merkle_root_from_vc): per garantire l'integrità degli attributi divulgati, questa funzione verifica che la Merkle root calcolata dagli attributi divulgati corrisponda alla Merkle root inclusa nella VC.
    - Estrae la Merkle root dalla VC (decodificandola senza verificare la firma, poiché la firma è già stata verificata).
    - Costruisce un Merkle Tree con gli attributi che lo studente ha deciso di rivelare.
    - Calcola la Merkle root da questo albero e la confronta con quella presente nella VC.
- Verifica credenziale (verify_credential):
    - Decifra il pacchetto e recupera gli attributi e il JWT della VC.
    - Decodifica il payload della VC.
    - Esegue la verifica del certificato di accreditamento dell'issuer.
    - Recupera la chiave pubblica dell'issuer e verifica la firma della VC.
    - Esegue la verifica della Merkle root.
    - Controlla lo stato di revoca della credenziale.
    - Se tutte le verifiche passano, la credenziale è considerata valida.
- Archiviazione credenziale in DApp (store_credential_in_dapp): archivia gli attributi divulgati e il JWT della VC nell'applicazione decentralizzata dello studente.
- Creazione di presentazione verificabile (create_verifiable_presentation): consente allo studente di generare una Verifiable Presentation per un verificatore, divulgando selettivamente solo una parte dei propri attributi.
    - Recupera la credenziale completa dalla StudentDApp.
    - Permette allo studente di selezionare quali attributi divulgare.
    - Calcola le Merkle proofs per gli attributi selezionati rispetto al Merkle Tree completo della VC. Confronta il valore recuperato dal calcolo interno dell'albero Merkle con il value che lo studente intende divulgare.
    - Se questi valori non corrispondono, viene stampato un avviso, indicando una potenziale discrepanza o un problema con l'integrità dei dati *prima* che la proof venga inclusa. Questo aggiunge un ulteriore livello di robustezza e rilevamento degli errori.
    - Costruisce un payload per la VP che include il DID del titolare, il DID del verificatore, i dati divulgati, le Merkle proofs e un nonce (per prevenire attacchi di replay).
    - Utilizza CredentialUtils per generare il JWT della VP.
- Trasmissione (transmit): metodo per inviare messaggi cifrati ad altri attori.
    - Ottiene la chiave pubblica del destinatario dal DID Registry.
    - Cifra il messaggio usando la crittografia ibrida.
    - Restituisce il messaggio cifrato.

Verifier

Estende la classe base Actor ed è progettata per il ruolo cruciale di verificare l'autenticità e l'integrità delle VP ricevute. Si assicura che le informazioni presentate siano valide, non siano state manomesse e provengano da fonti affidabili. Le funzionalità principali sono:

- Come gli altri attori, genera la propria coppia di chiavi RSA, il proprio DID e il DID Document. Viene anche inizializzato un nonce_registry, un set utilizzato per tenere traccia dei nonce già visti, prevenendo così replay attacks.
- Decifratura presentazione (decrypt_presentation): decifra il pacchetto di dati crittografato ricevuto, che contiene una Verifiable Presentation.
    - Utilizza la crittografia ibrida per decifrare il pacchetto con la chiave privata del Verifier.
    - Una volta decifrato, estrae il JWT della VP e, dal suo payload, recupera gli attributi divulgati (studentData).
- Controllo revoca (check_revocation): verifica se la VC è stata revocata dall'issuer.
    - Estrae l'ID della credenziale (jti) dal payload della VC.
    - Interroga il Revocation Registry per controllare lo stato di revoca della credenziale.
- Verifica nonce (verify_nonce): controllo di sicurezza fondamentale per prevenire attacchi di replay.
    - Verifica che il nonce sia presente nel payload della VP.
    - Controlla se il nonce è già stato utilizzato in precedenza (cercandolo nel nonce_registry). Se sì, segnala un possibile replay attack.
    - Se il nonce è nuovo e valido, lo aggiunge al registro per futuri controlli.
- Verifica firma dello studente (verify_holder_signature): assicura che la Verifiable Presentation sia stata effettivamente firmata dallo studente che la sta presentando.
    - Decodifica il JWT della VP per ottenere il DID dello studente (holder_did).
    - Recupera la chiave pubblica dello studente dal DID Registry.
    - Verifica la firma del JWT della VP utilizzando la chiave pubblica dello studente e l'algoritmo RS256, assicurandosi anche che il destinatario (audience) sia il DID del Verifier stesso.
- Verifica firma dell'issuer e del certificato (verify_issuer_signature): verifica la catena di fiducia fino all'issuer della VC e all'Autorità di Accreditamento che ha accreditato l'issuer.
    - Decodifica il JWT della VC per ottenere il DID dell'issuer e il DID dell'autorità di accreditamento.
    - Recupera la chiave pubblica dell'autorità di accreditamento e il certificato di accreditamento dell'issuer dal DID Registry.
    - Decodifica e valida il certificato di accreditamento usando la chiave pubblica dell'autorità, assicurandosi che il certificato accrediti correttamente l'issuer.
    - Recupera la chiave pubblica dell'issuer e verifica la firma del JWT della VC con essa.
- Verifica Merkle root dalla VP (verify_merkle_root_from_vp): garantisce l'integrità e la validità dei dati che lo studente ha scelto di divulgare nella presentazione.
    - Estrae gli attributi divulgati (studentData) e le Merkle proofs dal payload della VP.
    - Recupera la Merkle root attesa dal payload della VC.
    - Per ogni Merkle proof fornita, utilizza il metodo MerkleTree.verify_merkle_proof per confermare che l'attributo divulgato è effettivamente incluso nella VC originale e che la proof è valida.

    - Appiattisce prima i disclosed_attributes per creare un dizionario. Poi, per ogni path (chiave dell'attributo) e la sua proof trovati in merkle_proofs, verifica attivamente se quel path esiste all'interno dei dati flat_disclosed.
    - Ricostruisce l'hash della foglia per ogni attributo divulgato.
- Verifica presentazione (verify_presentation): è il metodo principale, che orchestra l'intero processo di verifica di una Verifiable Presentation.
    - Decifra il pacchetto ricevuto per ottenere il JWT della VP e gli attributi divulgati.
    - Esegue la verifica del nonce.
    - Verifica la firma dello studente sulla VP.
    - Esegue la verifica completa della firma dell'issuer sulla VC e del relativo certificato di accreditamento.
    - Verifica le Merkle proofs per gli attributi divulgati rispetto alla Merkle root della VC.
    - Esegue il controllo dello stato di revoca della credenziale.
    - Se tutti i passaggi di verifica hanno successo, la presentazione è considerata valida e il metodo restituisce True insieme agli attributi divulgati e al JWT della VP. In caso contrario, restituisce False.

4.3. Blockchain e Smart Contract

Per simulare l'uso della blockchain sono state definite delle classi che simulano il suo funzionamento e gli smart contract che servono per il salvataggio dei DID e la funzionalità di revoca. Di seguito sono riportate delle descrizioni riguardanti le classi che hanno permesso tale simulazione.

Block

La classe Block rappresenta un'unità fondamentale di una blockchain. Ogni blocco contiene un insieme di transazioni, un riferimento al blocco precedente, e viene convalidato attraverso un processo di "proof of work". Questa struttura garantisce l'immutabilità e l'integrità della catena di blocchi. Le funzionalità principali sono:

- Un blocco viene inizializzato con un indice, una lista di transazioni, l'hash del blocco precedente e un nonce. Il timestamp del blocco viene impostato automaticamente al momento della sua creazione, e il suo hash viene calcolato immediatamente.
- Calcolo dell'hash (calculate_hash): genera l'hash SHA-256 del blocco. L'hash è calcolato serializzando tutti gli attributi del blocco (indice, timestamp, transazioni, hash precedente, nonce) in una stringa JSON e poi applicando la funzione di hash. Questo assicura che qualsiasi minima modifica al contenuto del blocco cambierà il suo hash, rendendo le manomissioni facilmente rilevabili.
- Mining del blocco (mine_block): questo metodo implementa il "Proof of Work". Il blocco cerca un nonce tale che il suo hash inizi con un certo numero di zeri (definito dalla difficoltà). Questo processo è computazionalmente intenso e serve a convalidare il blocco e a mantenere la sicurezza della blockchain. Il nonce viene incrementato ripetutamente finché non viene trovato un hash che soddisfa la condizione di difficoltà.

Blockchain

La classe Blockchain rappresenta la catena di blocchi vera e propria e gestisce tutte le operazioni relative alla sua costruzione, validazione e interazione, inclusi i registri per i DID e per le revoche. È il cuore del sistema, che garantisce la decentralizzazione e la sicurezza delle informazioni. Le funzionalità principali sono:

- Alla sua creazione, la blockchain viene inizializzata con una catena vuota di blocchi, un elenco di transazioni in attesa di essere incluse in un blocco, una ricompensa per il mining e un livello di difficoltà per il proof of work. Viene immediatamente creato il primo blocco della catena, noto come "blocco genesis".
- Creazione del blocco genesis (create_genesis_block): genera il primissimo blocco della blockchain. È un blocco speciale con indice 0 e un hash precedente "0", che viene minato e aggiunto alla catena.
- Ottenere l'ultimo blocco (get_latest_block).
- Aggiungere una transazione (add_transaction): aggiunge una nuova transazione all'elenco delle pending_transactions. Queste transazioni rimarranno in attesa fino a quando non verrà minato un nuovo blocco.
- Mining delle transazioni in sospeso (mine_pending_transactions): Questo è il processo chiave per aggiungere nuovi blocchi alla blockchain.
    - Inizia aggiungendo una transazione di ricompensa per il miner alla lista delle transazioni in sospeso.

    - Crea un nuovo blocco includendo tutte le pending_transactions, l'indice successivo e l'hash dell'ultimo blocco della catena.
    - Esegue il mining del nuovo blocco, risolvendo il "Proof of Work".
    - Una volta minato, il blocco viene aggiunto alla catena e l'elenco delle pending_transactions viene svuotato.
- Validazione della catena (is_chain_valid): verifica l'integrità dell'intera blockchain.
    - Scorre tutti i blocchi della catena.
    - Per ogni blocco, verifica che il suo hash corrente sia stato calcolato correttamente.
    - Controlla che l'hash del blocco precedente memorizzato nel blocco corrente corrisponda effettivamente all'hash del blocco precedente nella catena.
    - Se una di queste condizioni non è soddisfatta, la catena è considerata invalida.
- Ottenere transazioni per tipo (get_transactions_by_type): filtra e restituisce tutte le transazioni di un tipo specifico che sono state incluse nei blocchi della blockchain. Questo è utile per interrogare i registri come il DID Registry o il Revocation Registry.

DIDRegistry

La classe DIDRegistry gestisce la registrazione e il recupero dei Decentralized Identifiers e dei loro DID Documents sulla blockchain. Agisce come un'interfaccia tra le entità e la blockchain sottostante, garantendo che le informazioni relative alle identità siano immutabili e verificabili. Questa classe permette di registrare nuovi DID con o senza certificati di accreditamento e di recuperare i dati associati in modo affidabile. Le funzionalità principali sono:

- La classe viene inizializzata con un riferimento a un oggetto Blockchain. Questo collegamento è essenziale perché il DIDRegistry opera registrando e recuperando informazioni direttamente da essa.
- Registrazione DID accreditati (save_accredited_did): permette di registrare un DID e il suo DID Document sulla blockchain, includendo anche un certificato di accreditamento JWT.
    - Crea una transazione di tipo "DID_REGISTRATION" che contiene il DID, il DID Document completo, il certificato di accreditamento, un hash del DID Document e un timestamp.
    - Aggiunge questa transazione alla lista delle transazioni in sospeso della blockchain, in attesa di essere inclusa in un blocco.
- Registrazione DID (save_did): simile alla funzione precedente, ma specifica per la registrazione di un DID e del suo DID Document senza un certificato di accreditamento.
    - Costruisce una transazione di tipo "DID_REGISTRATION" con il DID, il DID Document, l'hash del documento e il timestamp.
    - La transazione viene poi aggiunta alla blockchain.
- Recupero DID Document (get_did_document): permette di recuperare il DID Document più recente associato a un DID specifico, cercandolo tra le transazioni registrate sulla blockchain.
- Recupero chiave pubblica (get_public_key): permette di ottenere la chiave pubblica in formato PEM associata a un DID.
- Recupero certificato (get_certificate): recupera il certificato JWT di accreditamento associato a un DID, se questo è stato registrato.

RevocationRegistry

La classe RevocationRegistry è dedicata alla gestione della revoca delle credenziali. Il suo ruolo principale è garantire che le credenziali, una volta emesse, possano essere marcate come revocate in

modo immutabile e verificabile da chiunque consulti la blockchain. Questo è cruciale per la sicurezza e la validità a lungo termine delle credenziali. Le funzionalità principali sono:

- La classe viene inizializzata con un riferimento a un oggetto Blockchain. Questo le permette di interagire direttamente con il registro distribuito per pubblicare e verificare lo stato di revoca delle credenziali.
- Revoca della credenziale (revoke_credential): consente di revocare una specifica credenziale identificata dal suo ID.
    - Se la credenziale non è già revocata, crea un record di revoca che include l'ID della credenziale e un timestamp.
    - Costruisce una transazione di tipo "CREDENTIAL_REVOCATION" che contiene l'ID della credenziale, un hash del record di revoca e un timestamp.
    - Aggiunge questa transazione alla lista delle transazioni in sospeso della blockchain, rendendo la revoca permanente una volta che la transazione è inclusa in un blocco.
- Verifica stato di revoca (is_revoked): controlla se una data credenziale è stata revocata.
    - Cerca tra tutte le transazioni di tipo "CREDENTIAL_REVOCATION" presenti nella blockchain.
    - Se trova una transazione con l'ID della credenziale specificato, significa che la credenziale è stata revocata.

4.4. Other Technologies

Per simulare l'uso di tecnologie come la crittografia ibrida, il generatore di numeri casuali e la DApp da far utilizzare allo studente sono state create altrettante classi da utilizzare nel codice. Di seguito sono riportate le descrizioni con le funzionalità principali per le quali sono state create.

HybridCrypto

La classe HybridCrypto implementa un meccanismo di cifratura ibrida, combinando i vantaggi della crittografia asimmetrica (per la sicurezza dello scambio di chiavi) e della crittografia simmetrica (per l'efficienza nella cifratura di grandi quantità di dati). Questo approccio garantisce sia la confidenzialità che l'efficienza nella trasmissione dei messaggi. Le funzionalità principali sono:

- All'atto della creazione, un oggetto HybridCrypto inizializza un generatore di numeri casuali crittograficamente sicuro (CSPRNGGenerator). Questo generatore è fondamentale per creare chiavi di sessione e Initialization Vector imprevedibili, elementi essenziali per la sicurezza delle operazioni di cifratura.
- Cifratura (encrypt): cifra un messaggio in plaintext per un destinatario, utilizzando la sua chiave pubblica.
    - Generazione chiave simmetrica: inizialmente, viene generata una chiave simmetrica usa e getta (chiave di sessione) e un Initialization Vector utilizzando il CSPRNG.
    - Cifratura simmetrica: il messaggio in plaintext viene cifrato usando l'algoritmo AES in modalità CTR con la chiave di sessione e l'IV.
    - Cifratura chiave simmetrica: La chiave di sessione appena generata viene poi cifrata utilizzando la chiave pubblica RSA del destinatario. Viene impiegato il padding OAEP con SHA256 per garantire la sicurezza della cifratura asimmetrica.
    - Struttura del pacchetto cifrato: Il pacchetto finale cifrato è composto dalla lunghezza della chiave simmetrica cifrata, la chiave simmetrica cifrata, l'IV e i dati cifrati. Questa struttura permette al destinatario di decifrare correttamente il messaggio.
- Decifratura (decrypt): decifra un pacchetto di dati cifrato, utilizzando la chiave privata del destinatario.
    - Estrazione componenti: analizza il pacchetto cifrato ricevuto, estraendo la lunghezza della chiave simmetrica cifrata, la chiave simmetrica cifrata, l'IV e i dati cifrati.
    - Decifratura chiave simmetrica: utilizzando la propria chiave privata RSA, il destinatario decifra la chiave di sessione che era stata cifrata asimmetricamente. Anche qui viene usato il padding OAEP con SHA256 per la decifratura.
    - Decifratura dati: con la chiave di sessione e l'IV recuperati, i dati cifrati vengono decifrati usando l'algoritmo AES in modalità CTR, rivelando il messaggio originale in testo semplice.

MerkleTree

La classe MerkleTree implementa la struttura dati di un albero di Merkle, che è fondamentale per verificare l'integrità e l'autenticità dei dati in modo efficiente. Questo albero permette di riassumere un grande insieme di dati in un singolo hash (la Merkle root), e di dimostrare l'inclusione di specifici elementi senza rivelare l'intero dataset. È ampiamente utilizzato nelle blockchain e nei sistemi di verifica di credenziali per garantire che le informazioni non siano state alterate. Le funzionalità principali sono:

- L’albero viene creato a partire da un dizionario di dati. Durante l'inizializzazione, la classe appiattisce questi dati, calcola gli hash per ciascun elemento per formare le foglie, e quindi costruisce l'intero albero fino a ottenere la Merkle root.
- Ottenere la Merkle Root (get_merkle_root).
- Hash dei dati (hash_data): prende un singolo dato e ne calcola l'hash SHA-256.
- Appiattimento dei dati (flatten_data): trasforma strutture di dati annidate in una lista piatta di coppie (chiave_path, valore_atomico). Il chiave_path è una stringa che indica la posizione originale del dato all'interno della struttura nidificata (es., "studentInfo.name")..
- Costruzione delle foglie (build_leaves): prende i dati appiattiti e, per ogni coppia (path, value), crea un hash combinando il percorso e il valore.
- Costruzione dell'albero (build_tree): prende una lista di hash di foglie e costruisce progressivamente i livelli superiori dell'albero di Merkle. Ogni nodo genitore è l'hash della concatenazione dei suoi due figli. Se un livello ha un numero dispari di nodi, l'ultimo nodo viene duplicato per garantire che ogni nodo abbia una coppia. Questo processo continua iterativamente fino a quando non si arriva a un singolo hash, che è la Merkle root.
- Calcolo della Merkle Root (calculate_merkle_root): restituisce semplicemente l'hash del nodo all'ultimo livello dell'albero.
- Ottenere la Merkle Proof (get_proof): dato l'indice di una foglia, questo metodo genera la Merkle Proof per quella foglia.
- Calcolo della Merkle Proof per attributo (calculate_merkle_proof): semplifica la generazione della Merkle Proof per un attributo specifico dato il suo attribute_key (il suo percorso all'interno dei dati originali, ad esempio "studentInfo.name"). Trova l'indice della foglia corrispondente, estrae il valore e quindi invoca get_proof per ottenere il percorso della proof. Restituisce il valore dell'attributo e la sua Merkle Proof.
- Verifica della Proof (verify_proof): consente a chiunque di l'inclusione di un dato in un albero Merkle, a condizione che si abbia l'hash della foglia, la Merkle Proof e la Merkle root attesa. Il metodo ricostruisce il percorso di hashing dalla foglia alla radice utilizzando gli hash forniti nella proof e confronta l'hash finale calcolato con la Merkle root attesa. Se coincidono, la proof è valida e il dato è autentico.

CSPRNGGenerator

La classe CSPRNGGenerator fornisce metodi per generare dati casuali crittograficamente sicuri. È fondamentale per tutte le operazioni che richiedono imprevedibilità e sicurezza, come la creazione di chiavi di sessione, nonce e altro materiale segreto necessario per protocolli crittografici robusti. Le funzionalità principali sono:

- Generazione di materiale per chiavi (generate_key_material): genera un numero specificato di byte casuali crittograficamente sicuri. È ideale per la creazione di chiavi simmetriche o altri dati binari che devono essere imprevedibili e difficili da indovinare da un attaccante.
- Generazione di un nonce (generate_nonce): produce una stringa casuale. I nonce (number used once) sono valori casuali impiegati una singola volta in una comunicazione crittografica per prevenire attacchi di replay e garantire l'unicità di un'operazione.
- Generazione di una chiave di sessione (generate_session_key): genera una chiave di sessione di 32 byte. Le chiavi di sessione sono chiavi simmetriche temporanee usate per cifrare e decifrare i dati durante una singola sessione di comunicazione, offrendo un alto livello di sicurezza per lo scambio di informazioni.

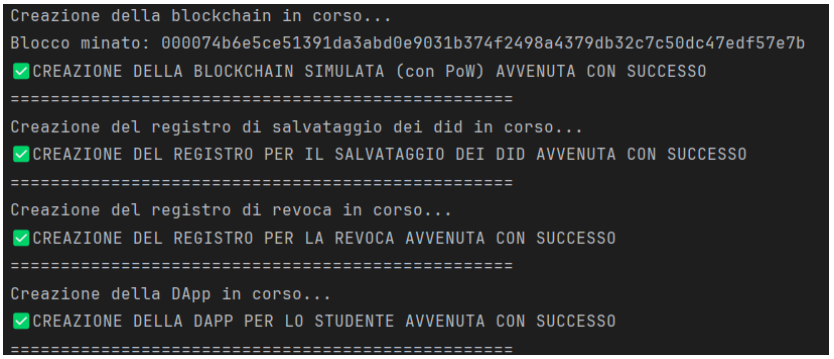
StudentDApp

La classe StudentDApp simula un'applicazione decentralizzata che uno studente usa per gestire le proprie credenziali. Il suo ruolo principale è quello di fornire un'interfaccia per archiviare una credenziale e, in particolare, permettere allo studente, dopo essersi autenticato, di selezionare quali specifici attributi desidera rivelare quando crea una presentazione verificabile. Le funzionalità principali sono:

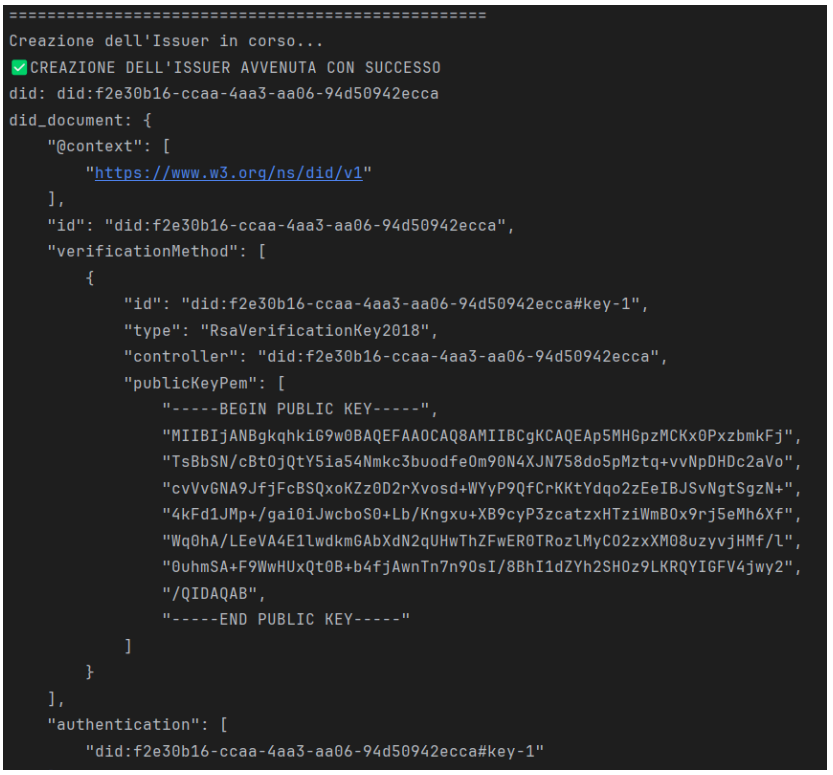
- Autenticazione utente (authenticate_user): permette allo studente di effettuare il login, inserendo email e password.
- Memorizzazione della credenziale (store_credential): permette allo studente di salvare la propria credenziale verificabile all'interno della DApp. La credenziale viene memorizzata come un dizionario che include gli attributi divulgati e il JWT della credenziale stessa.
- Recupero della credenziale (get_credential): restituisce la credenziale attualmente memorizzata nella DApp.
- Selezione degli attributi (select_attributes): permette allo studente di scegliere quali specifiche informazioni della propria credenziale desidera condividere.
    - Viene sempre incluso automaticamente il blocco "studentInfo" (informazioni di base dello studente).
    - Vengono presentate opzioni per la selezione di attributi relativi alle informazioni Erasmus (es. date, corsi, certificati linguistici, altre attività).
    - Lo studente interagisce tramite input testuale, inserendo i numeri corrispondenti agli elementi che desidera includere nella presentazione.

4.5. Esempio di esecuzione

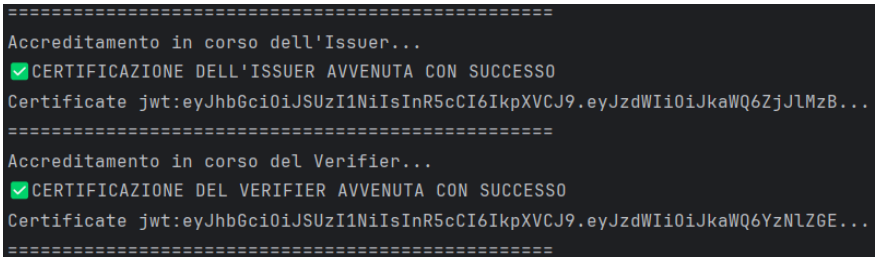
Adesso verrà descritto un esempio di esecuzione del codice fornito dal main. Per prima cosa bisogna inizializzare la blockchain e la DApp:



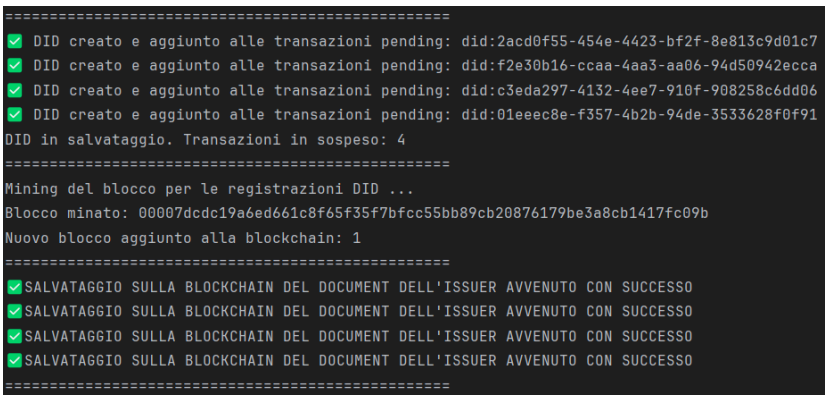
Dopo di che vi è bisogno di creare i diversi attori con i rispettivi DID document:



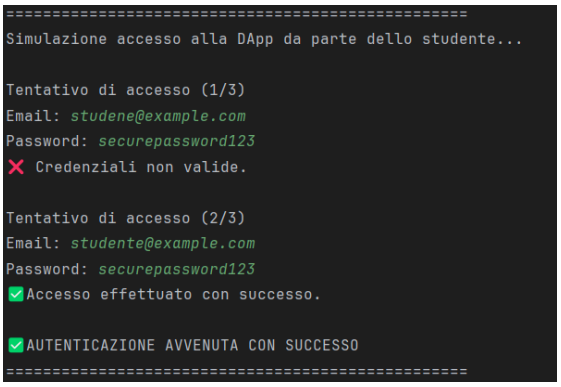
In questa immagine l'output viene tagliato in quanto troppo lungo. Inoltre la stampa avviene allo stesso modo per gli altri attori. Poi avviene l'accreditamento dell'issuer e del verifier tramite l'ente di accreditamento:



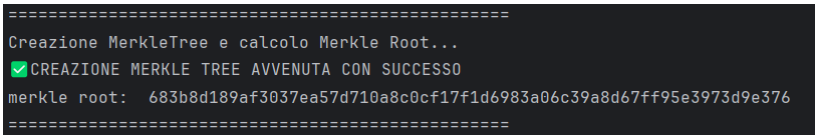
Successivamente si salvano i DID con i DID document ed eventuali certificati nella blockchain:



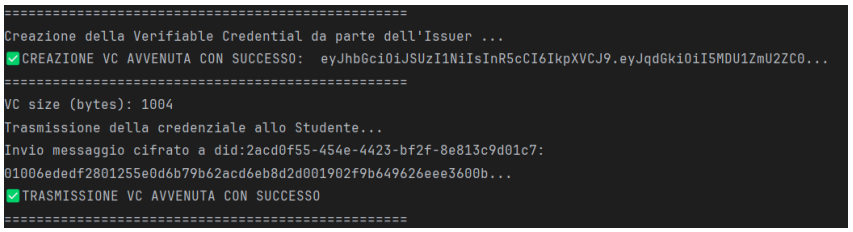
Infine lo studente si autentica per accedere alla DApp:



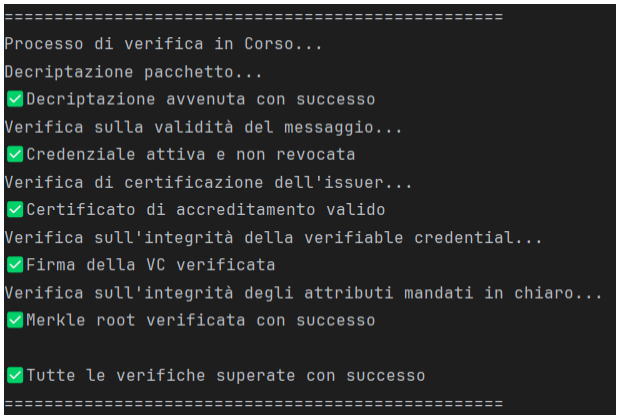
Con quest'ultima operazione può essere considerata conclusa la fase di settaggio. Dopo di che viene caricato un Mock dei dati accademici di uno studente ed è possibile iniziare la simulazione di una comunicazione standard. Per prima cosa si calcolano i merkle tree e merkle root dei dati:



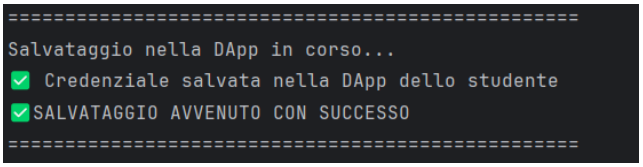
Dopo di che si crea e si trasmette la Verifiable Credential e gli attributi in chiaro allo Studente:



Lo Studente, una volta ricevute le informazioni, fa partire il processo di verifica:

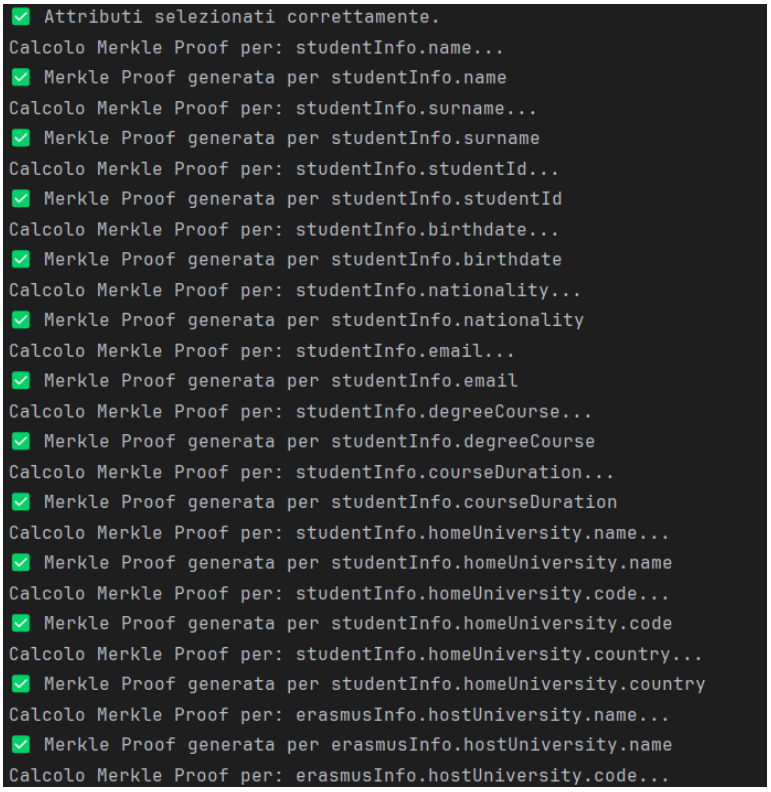


Finito il processo di verifica vengono mostrate a video le informazioni ricevute e vengono salvate nella DApp:

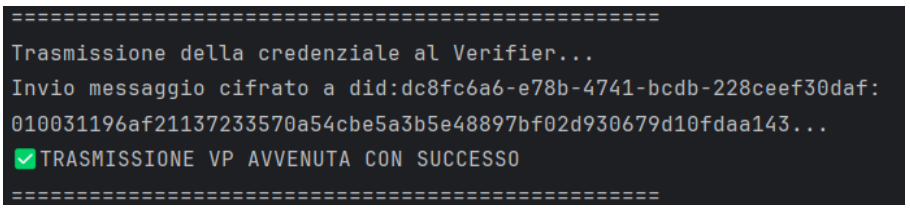


In questo momento parte il processo di divulgazione selettiva in cui lo studente può decidere quali informazioni fornire al destinatario (verifier):

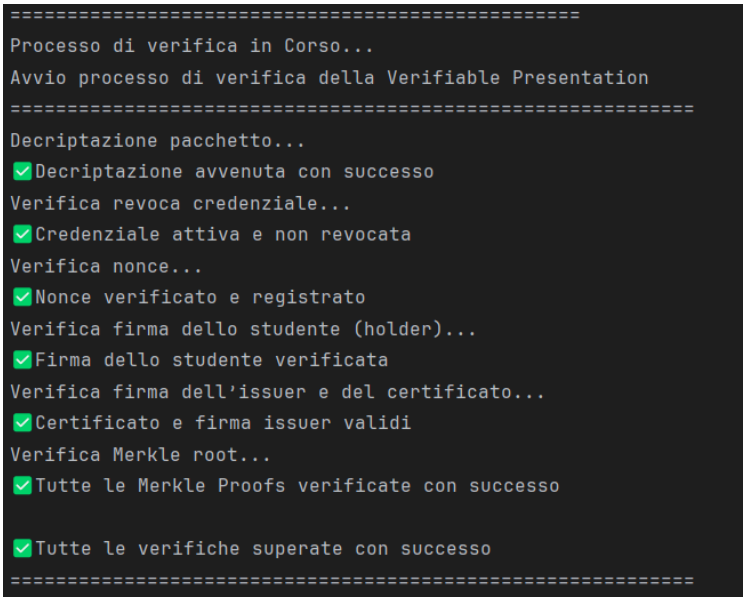
Dopo aver selezionato gli attributi parte il calcolo delle merkle proof per ognuno di esso:



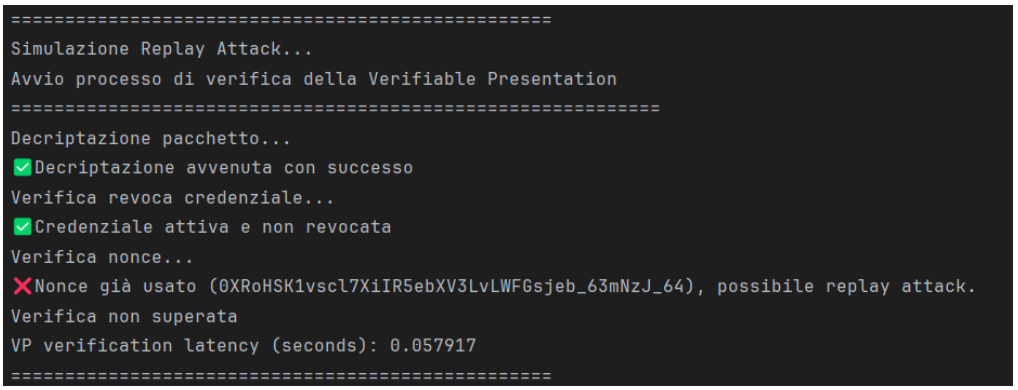
Essendo troppo lungo anche questa immagine non mostrerà l'intero output. Le informazioni da inviare vengono mostrate a video (attributi selezionati e Verifiable Presentation) e lo studente termina le sue attività trasmettendo tali informazioni al verifier:



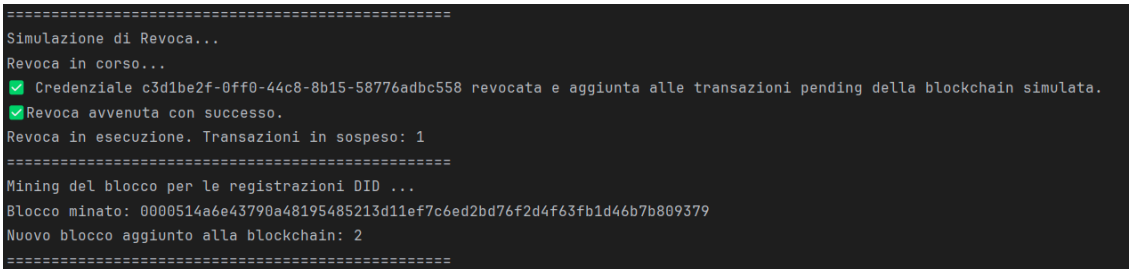
Il verifier, una volta ricevute, fa partire il processo di verifica:



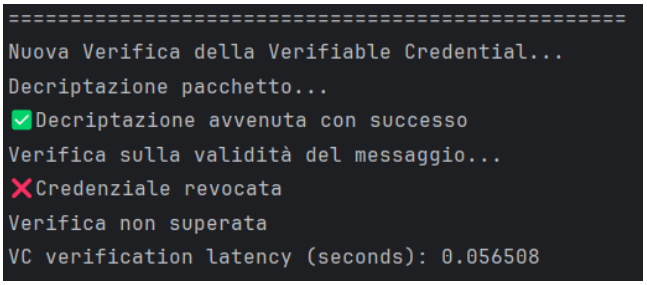
Infine vengono mostrare a video le informazioni ricevute facendo terminare la simulazione di una comunicazione standard. Dopo di che viene fatta partire la simulazione di un replay attack che viene fermato grazie all'uso del nonce:



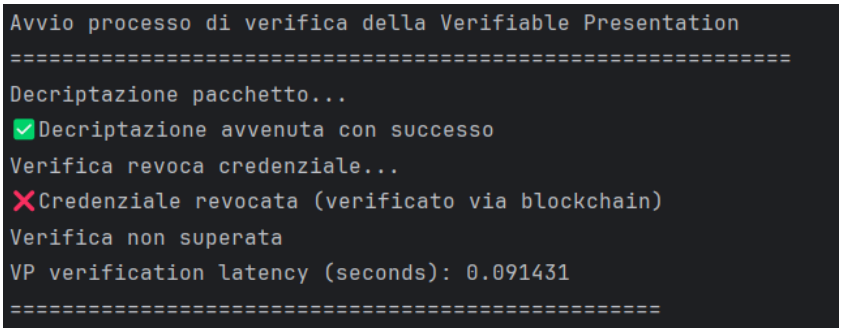
Infine viene simulata una comunicazione con revoca della credenziale. Prima avviene la revoca:



Poi avviene la trasmissione della vc dall'issuer allo studente e la verifica da parte di quest'ultimo:



Infine la creazione di una Verifiable Presentation partendo dalla Verifiable Credential revocata e mostrato a video come anche l'issuer non considera la VP valida perchè derivante da una VC revocata:



4.6. Analisi delle prestazioni

Una volta eseguito un esempio di uso del sistema, possiamo effettuare dei commenti circa la dimensione delle credenziali, delle presentazioni e sulla latenza di verifica.
Considerando il caso in cui lo studente selezioni tutte le informazioni, tranne quelle riguardanti certificazioni linguistiche ed altre attività extra, otteniamo:

VC size (bytes): 1004

- Risulta una credenziale verificabile di circa 1KB, estremamente compatta. Questo è in linea con l'obiettivo E.2 (La conservazione delle credenziali sul dispositivo dello studente deve minimizzare il consumo di memoria). Un formato così leggero è ideale per dispositivi con risorse limitate e minimizza l'impatto sullo storage.

VC verification latency (secondi): 0.067833 (circa 67 ms)

- Una latenza di verifica della Verifiable Credential di circa 67 millisecondi è molto buona, indica che il processo di ricezione e validazione della credenziale da parte dello studente è quasi istantaneo.

VP size (bytes): 20390 (circa 20.4 KB)

- La Verifiable Presentation mostra una dimensione di circa 20.4 KB nel caso in cui vengano divulgati quasi tutti gli attributi della credenziale originale. Questo aumento di dimensione rispetto alla VC è atteso e riflette la necessità di includere tutti gli hash intermedi della Merkle Proof per convalidare un numero elevato di attributi. Nonostante l'incremento, 20.4 KB rappresenta comunque una dimensione contenuta per un payload che garantisce crittograficamente la selettività e l'integrità. Questo dato dimostra l'efficacia del meccanismo di divulgazione selettiva basato su Merkle Tree nel minimizzare il traffico di rete, supportando l'obiettivo E.3 (La presentazione selettiva delle credenziali deve essere rapida e deve minimizzare il traffico di rete).

VP verification latency (secondi): 0.068898 (circa 69 ms)

- Nonostante l'aumento della dimensione della VP dovuto alla selezione di un numero maggiore di attributi, la latenza di verifica è rimasta notevolmente costante, attestandosi intorno ai 69 millisecondi. Questo dato è cruciale, poiché dimostra che l'impatto computazionale della verifica delle Merkle Proofs (la cui complessità è logaritmica rispetto al numero di attributi) è marginale anche nel caso di divulgazione quasi completa degli attributi. Tale efficienza nel tempo di verifica conferma pienamente il raggiungimento dell'obiettivo E.4 (La verifica delle credenziali ricevute deve essere veloce ed efficiente in termini di risorse computazionali), garantendo un'esperienza utente fluida e una rapida convalida delle presentazioni.

I risultati ottenuti evidenziano che il sistema proposto garantisce elevate prestazioni in termini di compattezza delle credenziali e rapidità dei processi di presentazione e verifica.