UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

ED ELETTRICA E MATEMATICA APPLICATA

Algoritmi e Protocolli per la Sicurezza

Project Work - WP1

Gruppo 26

Membri:

Antonio Carbone	a.carbone88@studenti.unisa.it	0622702566	WP 1-3
Enrico Cavuoto	e.cavuoto1@studenti.unisa.it	0622702565	WP 2-4

WP1 – Modello

1.1. Introduzione

Il progetto ha come obiettivo la realizzazione di una architettura per il rilascio e la diffusione di credenziali universitarie contenenti informazioni riguardanti attività svolte da uno studente universitario in Erasmus.

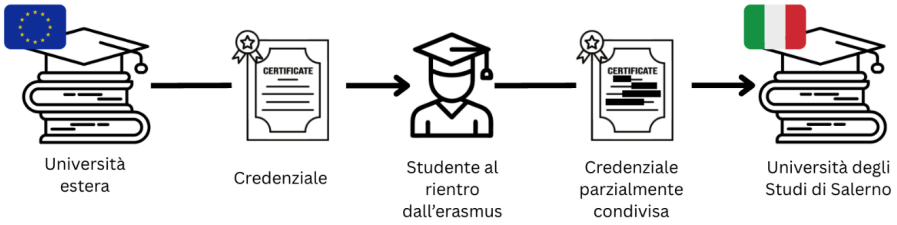
In particolar modo, si è alla ricerca di una soluzione che si discosti dalle tradizionali architetture dipendenti da una singola autorità centrale, permettendo verifiche di validità decentralizzate e divulgazione selettiva delle informazioni contenute all'interno.

Il possessore della credenziale, difatti, deve essere in grado di fornire parte delle informazioni contenute, consentendo all'ente verificatore di attestarne l'integrità e l'autenticità rispetto a quanto certificato dall'ente che le ha rilasciate.

Questo Work Package si pone l'obiettivo di:

1. Analizzare gli attori onesti del sistema
2. Individuare i threat model correlati a tutti i possibili attaccanti del sistema
3. Valutare le proprietà di sicurezza che l'architettura deve rispettare
4. Fornire una struttura alla credenziale universitaria

1.2. Attori onesti del sistema



1.2.1. Università estera

È, in questo scenario, l'organo preposto al rilascio di credenziali firmate e risulta pertanto un'entità fidata rispetto a tutti gli altri attori onesti del sistema. Quest'ultima, attraverso la generazione di tali credenziali, consente allo studente di certificare alla sua università di origine quanto svolto durante il suo soggiorno all'estero.

1.2.2. Studente

Detiene il possesso della credenziale nei suoi dispositivi, i quali possono essere di svariata natura, ad es. Smartphone, PC, Tablet, Wallet Hardware.

Può generare una credenziale contenente un set ridotto delle informazioni contenute in essa, condividendo, a qualunque università o ad altri enti, esclusivamente quanto a lui necessario.

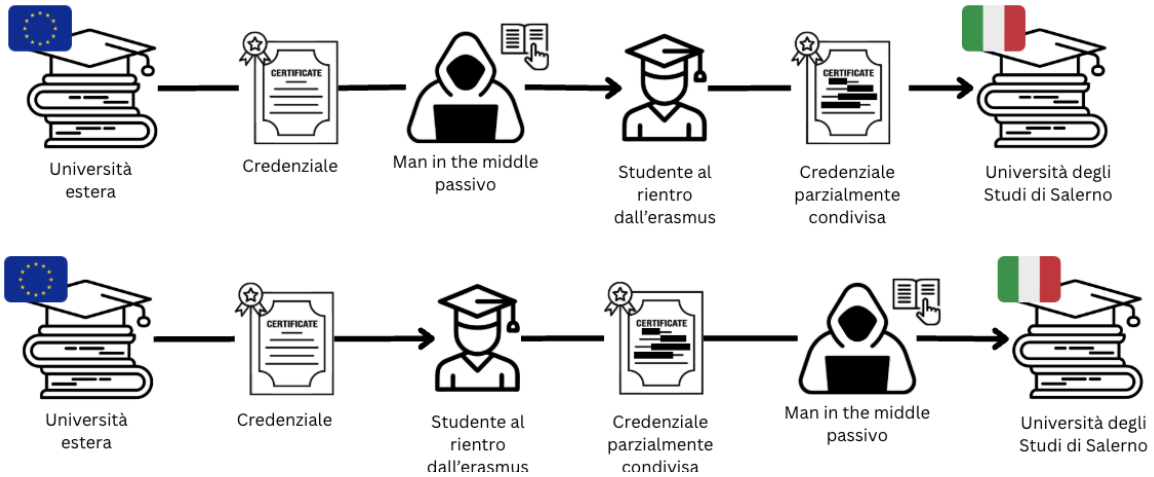
1.2.3. Università di origine

Ha la possibilità di verificare la validità di una credenziale fornita dallo studente al rientro dall'esperienza Erasmus, garantendo che essa non risulti revocata, certificandone l'autenticità e l'integrità dei dati, seppur parziali, contenuti all'interno di quest'ultima.

Si richiede che la validità sia verificata in maniera decentralizzata, evitando di interrogare l'ente certificatore per effettuare questa operazione.

1.3. Possibili attaccanti del sistema

1.3.1. Man in the middle passivo

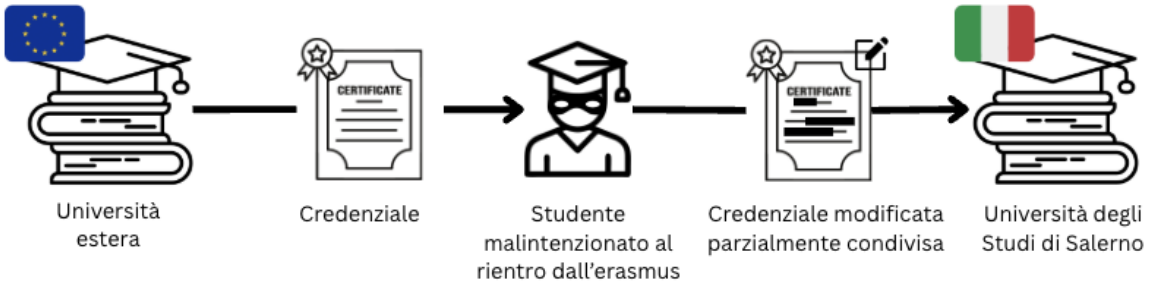


Utente terzo che vuole leggere le info del certificato, questo può avvenire sia intercettando la comunicazione della credenziale completa dall'università estera allo studente, oppure quando quest'ultimo condivide parzialmente la credenziale al rientro dal periodo di Erasmus.

Risorse e capacità dell'attaccante

L'attaccante dispone di una potenza di calcolo limitata, ma ha accesso ai canali di comunicazione utilizzati dalle parti oneste.

1.3.2. Utente certificato malintenzionato

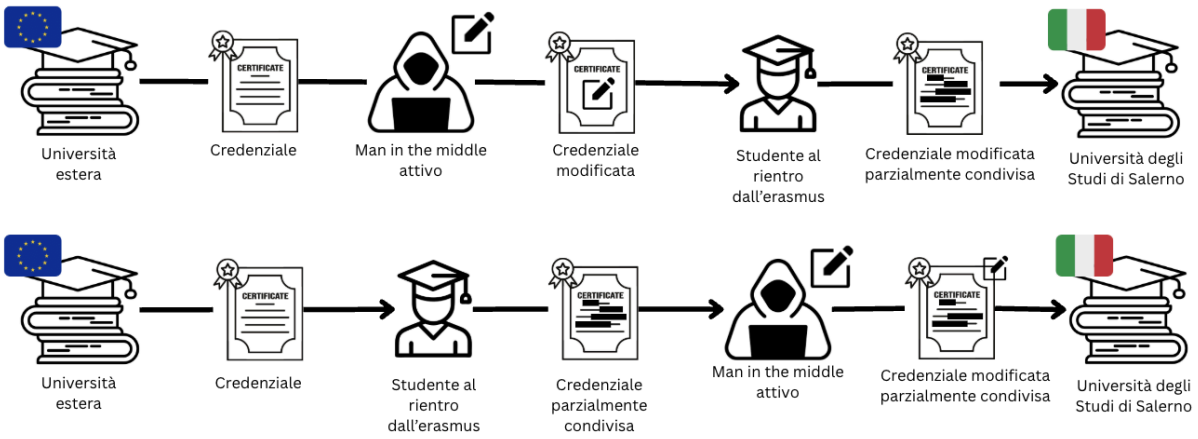


Utente possessore della credenziale intenzionato a modificarne il contenuto in modo tale da trarne vantaggio. Per esempio, sostituendo i voti ottenuti in Erasmus con voti più alti così da alzare la media al rientro nell'università italiana.

Risorse e capacità dell'attaccante

L'attaccante dispone di una potenza di calcolo limitata ma ha pieno accesso alla propria credenziale universitaria, ai canali di comunicazione e la sua identità è riconosciuta da tutte le parti oneste coinvolte.

1.3.3. Man in the middle attivo

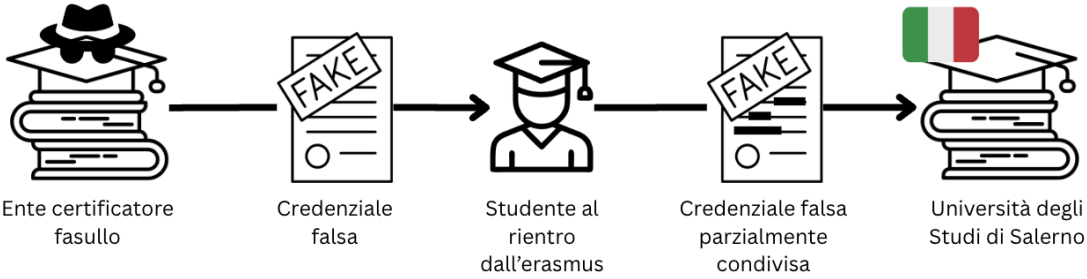


Utente terzo che vuole modificare le informazioni del contratto intercettato durante la comunicazione tra l'utente possessore delle credenziali e l'altra parte onesta. Questo può accadere sia in favore dell'utente certificato che in sfavore. Per esempio, l'utente terzo può essere un amico dell'utente possessore delle credenziali intenzionato ad aiutarlo alzandogli i voti ottenuti in Erasmus, oppure può essere un malintenzionato che vuole abbassarli o modificare dati sensibili in modo tale da creare problemi all'utente certificato.

Risorse e capacità dell'attaccante

Risorse computazionali limitate ma ha accesso ai canali di condivisione della credenziale.

1.3.4. Ente certificatore fasullo



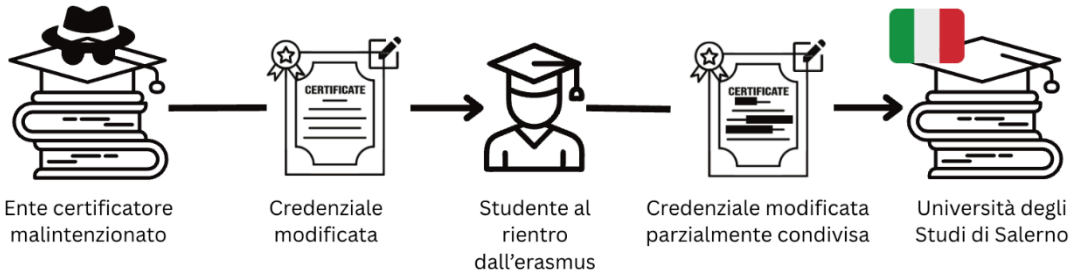
Ente certificatore, all'apparenza affidabile, che rilascia credenziali con informazioni false che vengono però riconosciute dagli enti di validazione.

Ad esempio, un ente malevolo al quale lo studente si rivolge per acquistare una credenziale che attesti un'esperienza Erasmus mai avvenuta.

Risorse e capacità dell'attaccante

Dispone di risorse computazionali elevate e ha la possibilità di registrare delle credenziali totalmente conformi all'iter di verifica.

1.3.5. Ente certificatore malintenzionato



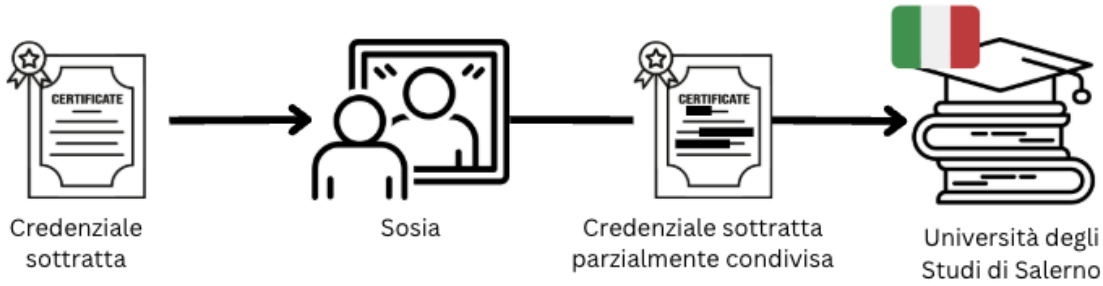
In questo caso è un ente certificatore autorizzato ed inizialmente fidato che agisce in maniera fraudolenta rilasciando credenziali modificate con informazioni non veritiere.

Ad esempio, un'università che rilascia certificati modificati in modo da favorire i suoi studenti.

Risorse e capacità dell'attaccante

Dispone di risorse computazionali elevate ed ha la possibilità di registrare delle credenziali valide.

1.3.6. Sosia



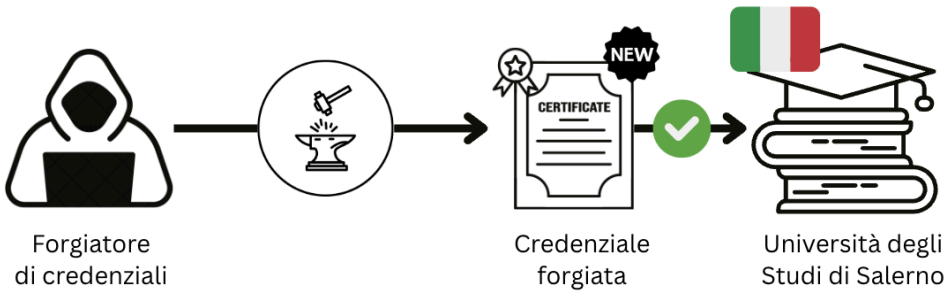
Utente terzo che vuole fingersi lo studente possessore della credenziale presentando il suo certificato, precedentemente ottenuto in qualche maniera.

Ad esempio, lo studente Mario Rossi potrebbe tentare di utilizzare la credenziale parziale contenente l'esito di un esame svolto da un suo collega, con lo scopo di ottenerne la convalida.

Risorse e capacità dell'attaccante

Dispone di risorse computazionali minime ma ha accesso ad una credenziale sottratta ad un altro studente.

1.3.7. Forgiatore di credenziali



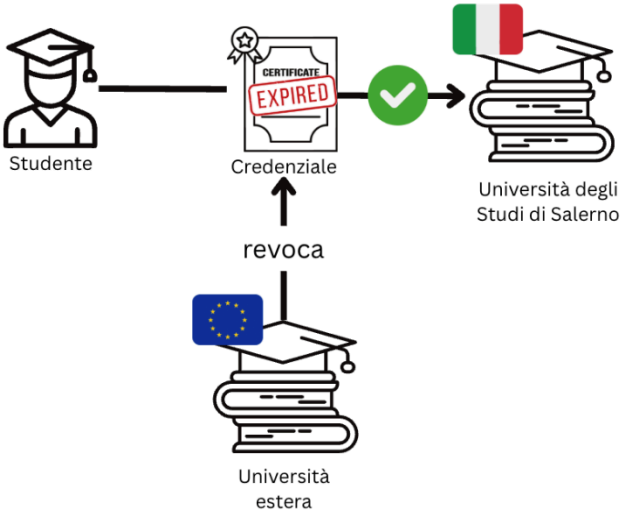
Utente terzo che vuole falsificare autonomamente una credenziale fasulla che certifica un'esperienza di studio all'estero mai avvenuta al fine di trarne beneficio nella sua carriera universitaria.

Ad esempio, lo studente Mario Rossi potrebbe tentare di generare una credenziale a partire dalle specifiche.

Risorse e capacità dell'attaccante

Dispone di risorse computazionali minime ma ha accesso ad una credenziale sottratta ad un altro studente.

1.3.8. Revoca inefficace



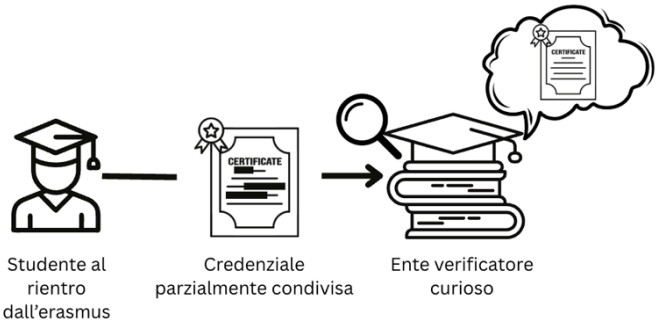
Una credenziale revocata dall'emittente potrebbe passare i controlli applicati da un ente verificatore, a causa di un meccanismo di revoca lento, inefficace o inaffidabile.

Tale vulnerabilità permetterebbe ad un attaccante di continuare a trarre beneficio, per un periodo successivo alla revoca, da una credenziale non più valida.

Risorse e capacità dell'attaccante

L'attaccante non dispone di risorse computazionali considerevoli, l'attacco può avere successo o meno a seconda di fattori che non sono da lui direttamente controllabili.

1.3.9. Ente verificatore curioso



Un ente verificatore curioso è un soggetto terzo inizialmente fidato che tenta di ricavare informazioni ulteriori rispetto a quelle che gli vengono comunicate.

Questo può avvenire attraverso:

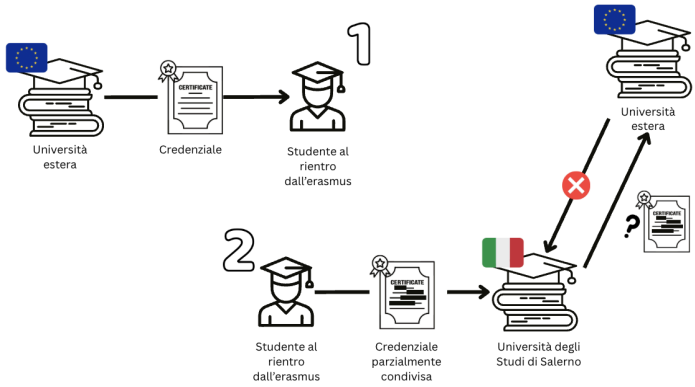
- Analisi statistiche sulla credenziale fornita.
- Attacchi di forza bruta su specifiche informazioni divulgate selettivamente, sfruttando conoscenze pregresse.

Ad esempio, un attaccante potrebbe analizzare le informazioni condivise per cercare di ottenere informazioni su di un esame superato di cui conosce il nome e, potenzialmente, i CFU ma che lo studente non ha divulgato.

Risorse e capacità dell'attaccante

Dispone di risorse computazionali medio-alte e ha accesso ad una credenziale parzialmente condivisa ottenuta in maniera legittima.

1.3.10. Ente certificatore rinnegante

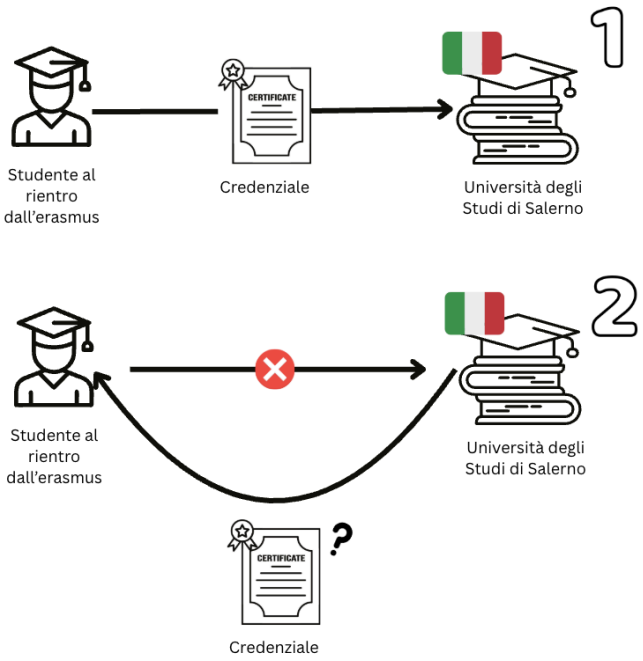


Un ente certificatore rinnegante è un ente in grado di registrare una credenziale ma che, nel momento in cui uno studente presenta tale documento ad un ente verificatore, ripudia di essere l'autore di tale documento.

Risorse e capacità dell'attaccante

Dispone di risorse computazionali medio-alte e ha modo di registrare credenziali valide.

1.3.11. Possessore di credenziale rinnegante



Un possessore di una credenziale rinnegante è, per l'appunto, una persona a cui è stata rilasciata una credenziale che al momento della sua condivisione ne ripudia il possesso o il contenuto.

Risorse e capacità dell'attaccante

Dispone di risorse computazionali basse, ma possiede una credenziale valida.

1.4. Proprietà

1.4.1. Integrità dei dati

ID	Proprietà
I.1.	Le informazioni certificate attraverso la credenziale non devono poter essere modificabili da nessuna parte onesta o meno.

1.4.2. Confidenzialità dei dati

ID	Proprietà
C.1.	Le credenziali devono essere condivise tra le parti oneste garantendo la confidenzialità
C.2.	Lo storage della credenziale da parte dello studente deve avvenire in maniera sicura

1.4.3. Autenticazione

ID	Proprietà
A.1.	Le informazioni contenute all'interno delle credenziali rilasciate dall'ente fidato devono essere autentiche ed essa deve poter essere verificata da chi riceve tali informazioni.
A.2.	Le identità delle parti coinvolte nello scambio di informazioni (credenziali totali o parziali) devono essere verificabili.

1.4.4. Non ripudio

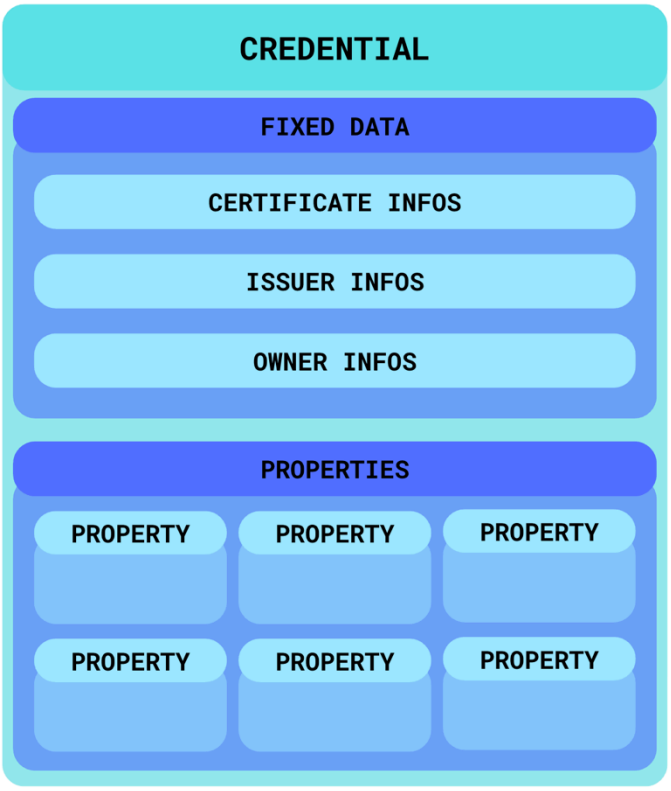
ID	Proprietà
N.R.1.	L'ente certificatore non può ripudiare una credenziale dopo averla rilasciata.
N.R.2	Uno studente non può ripudiare di aver inviato una credenziale o affermare che la credenziale da lui inviata avesse al suo interno informazioni diverse

1.4.5. Efficienza

ID	Proprietà
E.1.	Il meccanismo di storage delle credenziali deve essere efficiente, consentendo l'utilizzo di hardware-wallet o dispositivi con risorse computazionali limitate.
E.2.	Il meccanismo di verifica delle credenziali deve essere efficiente e decentralizzato.

1.5. Il formato della credenziale

Si è scelto di implementare la credenziale con uno schema che risulta condiviso tra la credenziale completa rilasciata dall'università ospitante e la credenziale parziale che può essere generata al proprietario. Il formato è il seguente:



1.5.1. Struttura della credenziale

La credenziale può essere suddivisa in 2 sezioni principali, la prima comprende:

- tutti i campi che sono relativi all'ente che ha emesso la certificazione
- data di emissione e l'identificativo univoco del documento
- informazioni sul possessore del certificato

Nella seconda sezione, denominata 'properties', sono presenti tutte le informazioni che devono essere inserite nel certificato, comprendendo ad esempio gli esami di profitto svolti, le attività extra, il luogo di residenza all'estero ma anche informazioni riguardanti eventuali compensi economici che lo studente ha ricevuto durante la sua esperienza Erasmus.

Una credenziale parziale derivata da quest'ultima può contenere un numero inferiore di properties ma deve necessariamente contenere la restante parte delle informazioni della parte fissa.

WP2 – Soluzione

2.1. Introduzione

Il focus di questo progetto, come anche specificato dai requisiti, è sulla struttura di una credenziale, sul suo meccanismo di condivisione parziale e sull'architettura che permette di verificarne la validità o effettuarne la revoca decentralizzata.

La blockchain Ethereum sarà la spina dorsale di questa architettura, rappresentando la tecnologia su cui rilasci e revoche di credenziali ed identificativi saranno basati.

Il Merkle Tree, invece, sarà la struttura dati utilizzata per l'hashing delle credenziali e per la gestione della logica di condivisione parziale.

In questo work package, quindi, la soluzione verrà illustrata spiegando, prima, l'architettura del sistema, e poi seguendo quelle che sono le fasi che vanno dall'immatricolazione presso l'università di origine fino alla condivisione parziale o la possibile revoca della credenziale carriera, rilasciata dall'università estera presso cui è stato svolto l'Erasmus.

2.2. L'Architettura

2.2.1. La Spina Dorsale

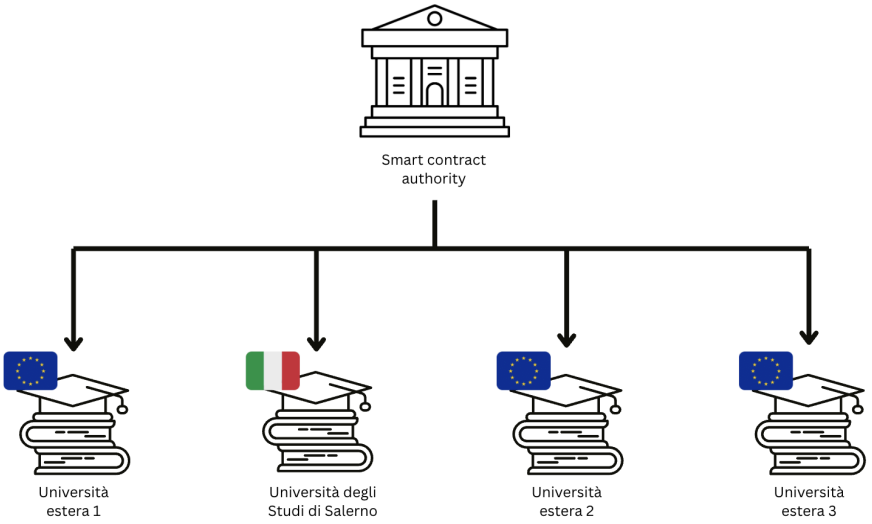
Come appena anticipato, la spina dorsale dell'intera architettura è la blockchain Ethereum.

La soluzione si propone di utilizzarla sia per la registrazione degli studenti da parte delle università di origine, sia per la registrazione delle credenziali carriera che vengono rilasciate a fine Erasmus.

Nella proposta ideata, si assume l'utilizzo della blockchain pubblica di Ethereum, considerando i costi delle transazioni come parte del costo di utilizzo attivo del sistema, dovendo pagare esclusivamente per operazioni di registrazione e modifica e non per la semplice consultazione.

Il funzionamento del sistema sarebbe del tutto equivalente, ma con una diversa gestione del costo di utilizzo, se si decidesse di utilizzare una blockchain privata tra le istituzioni partecipanti.

2.2.1. La Gerarchia



Prima di imbattersi nel funzionamento dell'intera architettura bisogna chiarire la posizione delle figure coinvolte nel rilascio delle credenziali e nella registrazione degli studenti.

L'assunzione alla base di questo meccanismo è che un'autorità centrale fidata si occupa di certificare l'operato delle università partecipanti al progetto Erasmus, dando loro il permesso di registrare utenti e credenziali su blockchain.

Infatti è l'autorità centrale che crea, in primis, uno smart contract per la registrazione delle università rendendole, quindi, entità fidate. Le università, inoltre, hanno accesso a questo smart contract unicamente per la verifica dell'id di altre università della rete.

`UID:<smart_contract_authority>:<id_univoco>`

Inoltre, rendendo un'università fidata, la Smart Contract Authority, da anche il permesso di creare un proprio smart contract per la registrazione degli studenti ed un proprio smart contract per la registrazione delle credenziali rilasciate.

Registrazione degli Studenti

L'identificazione degli studenti è una struttura basata su identificativi registrati sulla blockchain ed associati alla chiave pubblica che ogni studente presenta in fase di registrazione.

La registrazione di queste coppie ID : chiave pubblica avviene attraverso uno smart contract registrato dall'università di origine e propriamente comunicato all' autorità centrale.

`SID:<università_di_origine>:<id_univoco>`

Registrazione delle Credenziali Carriera Erasmus

Allo stesso modo, le credenziali che le università rilasciano agli studenti al termine del periodo in Erasmus, vengono registrate sulla blockchain e collegate ad un id attraverso un altro smart contract registrato dall'università che si occupa del rilascio della data credenziale associata al CID

`CID:<università_ospitante>:<id_univoco>`

2.2.3. Struttura degli Smart Contract

Smart Contract Authority

Come illustrato nella gerarchia, l'autorità centrale è l'ente preposto all'accreditamento delle università all'interno del sistema di rilascio credenziali.

Sulla Blockchain, la Smart Contract Authority registra uno smart contract il cui indirizzo viene assunto noto a tutte le parti oneste del sistema. Il contratto consente la gestione delle università aderenti al sistema,attraverso le seguenti operazioni:

Operazione	Descrizione	Modifier
registraUniversita(uid, publicKey, isRevoked, sidContractAddress, cidContractAddress)	Il metodo, eseguibile solo dalla Smart Contract Authority, consente l'accreditamento di una nuova università all'interno del sistema di gestione delle credenziali.	onlyOwner
modificaInfoUniversita(uid, newPublicKey, newIsRevoked, newSidContractAddress, newCidContractAddress)	Il metodo, eseguibile solo dalla Smart Contract Authority, consente la modifica delle informazioni associate all'UID di una università già accreditata. Il metodo è utile per poter gestire eventuali modifiche a parametri come chiave pubblica e/o indirizzi degli smart contract registrati, così da poter gestire eventuali cyber attacchi nei confronti di un'università.	onlyOwner
setRevokeStatusUniversita(uid, newIsRevoked)	Il metodo, eseguibile solo dalla Smart Contract Authority, consente di revocare o di riattivare l'accreditamento al sistema di un'università.	onlyOwner
getUniversityInfo(uid)	Metodo che consente di ottenere informazioni certificate a riguardo di un UID. Con questo metodo è possibile ottenere la sua chiave pubblica e lo stato di validità.	view
verificaSid(uid, sid)	Metodo che consente di ottenere informazioni riguardo un SID registrato da una data università abilitata. Con questo metodo è possibile ottenere la sua chiave pubblica e lo stato di validità.	view
verificaCid(uid, cid)	Metodo che consente di ottenere informazioni riguardo il CID di una credenziale, consentendo di verificarne la validità.	view

SID Smart Contract

Il contratto di gestione dei SID è un contratto con una struttura comune, registrato da ogni università aderente al sistema.

Operazione	Descrizione	Modifier
registraSid(sid, publicKey, isValid)	Il metodo, eseguibile solo dall'università proprietaria dello smart contract, consente di registrare un nuovo SID, associando ad esso la chiave pubblica.	onlyOwner
modificaSid(sid, newPublicKey, newIsValid)	Il metodo, eseguibile solo dall'università proprietaria dello smart contract, consente di modificare un SID, associando ad esso nuove informazioni	onlyOwner
getInfoSid(sid)	Il metodo consente di ottenere informazioni su di un dato SID e viene invocato dallo smart contract della Smart Contract Authority.	view

CID Smart Contract

Il contratto di gestione dei CID è un contratto con struttura comune, registrato da ogni università aderente al sistema.

Operazione	Descrizione	Modifier
registraSid(cid, isValid)	Il metodo, eseguibile solo dall'università proprietaria dello smart contract, consente di registrare il CID di una nuova credenziale, associandolo con	onlyOwner
modificaSid(cid, newIsValid)	Il metodo, eseguibile solo dall'università proprietaria dello smart contract, consente di modificare la validità di un certo CID.	onlyOwner
getInfoCid(cid)	Il metodo consente di ottenere informazioni su di un dato CID e viene invocato dallo smart contract della Smart Contract Authority.	view

2.2.4. Struttura della Credenziale

Le credenziali vengono rilasciate in formato JSON e rispettano il formato anticipato nel WP1. Esse, infatti, sono dotate di una sezione fissa che contiene gli identificativi univoci della credenziale, dell'issuer e dello studente a cui essa appartiene, oltre che una data di rilascio.

Fixed Data

Campo	Valore
credentialId	id associato alla credenziale, registrato su blockchain dall'università che l'ha rilasciata
studentId	id associato allo studente, registrato su blockchain dall'università di origine
issuerId	id associato all'università che rilascia la credenziale, registrato su blockchain dalla Smart Contract Authority
issuanceDate	data in cui la credenziale è stata registrata

La sezione dinamica della credenziale comprende invece un array di proprietà, ossia informazioni che possono essere separate atomicamente nella divulgazione selettiva da parte dello studente.

Le proprietà sono discriminate in base alla loro tipologia ed ognuna di esse può contenere al suo interno un insieme diverso di informazioni, rappresentate dal campo 'data'. A questi due si aggiunge un campo 'nonce' che garantisce maggiore sicurezza e che verrà approfondita nel WP3.

Si assume che le proprietà e la loro struttura seguano uno standard definito dalla Smart Contract Authority e che, pertanto, tutte le università partecipanti siano in grado di interpretare in maniera univoca.

A scopo dimostrativo della soluzione, sono state individuate le seguenti tipologie di proprietà:

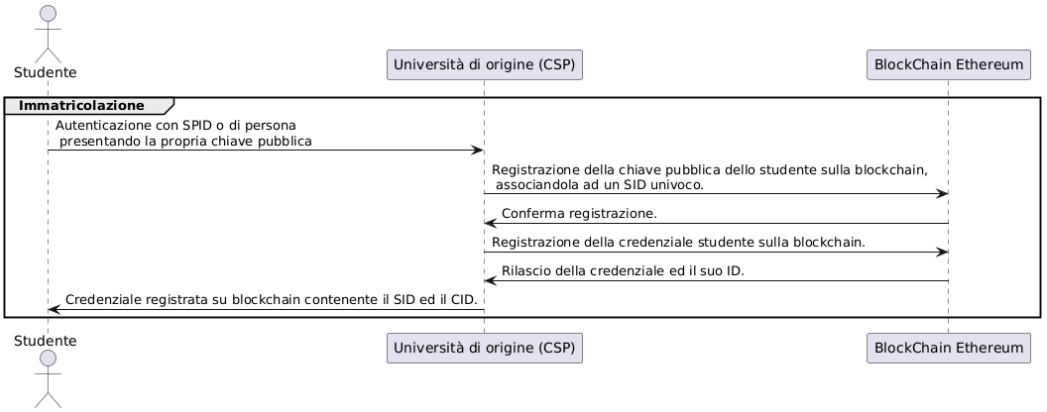
Properties

Tipologia	Dati
subjectInfo informazioni anagrafiche	**name:** nome del proprietario della credenziale **surname:** cognome del proprietario della credenziale **birthDate:** data di nascita del proprietario della credenziale **gender:** genere del proprietario della credenziale **nationality:** nazionalità del proprietario della credenziale **documentNumber:** numero di documento del proprietario della credenziale **documentIssuer:** ente che ha emesso il documento fornito in documentNumber **email:** email del proprietario della credenziale

achievedExam informazioni su esame di profitto superato	**name:** nome dell'esame sostenuto **grade:** voto **cfu:** crediti formativi **achievementData:** data di convalidazione
extraActivity attività extra svolta	**name:** nome del corso extra seguito **cfu:** crediti formativi **date:** data di convalidazione
residenceInfo informazioni sulla residenza universitaria	**type:** tipologia di residenza **address:** indirizzo
scholarship informazioni su borsa di studio	**amount:** importo della borsa di studio **unit:** valuta **payments:** numero di rate erogate

2.3. Il Funzionamento

2.3.1. Immatricolazione Presso Università di Origine



Un qualsiasi studente, in quanto tale, per intraprendere un'esperienza di Erasmus si presuppone che sia già regolarmente iscritto ad un'università.

Questo procedimento, come accennato, prevede che l'università abbia registrato la chiave pubblica dello studente sulla blockchain, associandola allo Student ID.

Ma come si fa ad essere certi dell'identità dello studente in sede di immatricolazione?

All'atto dell'immatricolazione, l'università di origine fa da Credential Service Provider (CSP) verificando l'identità dello studente. Questo può avvenire in due modi: o usando una CSP di livello superiore come SPID, o tramite un riconoscimento di persona. In entrambi i casi, l'università funge da garante dell'identità dello studente.

Questo garantisce all'università la validità delle generalità presentate dallo studente, così da soddisfare la proprietà A.2. in tutte le comunicazioni future a cui prenderà parte lo studente.

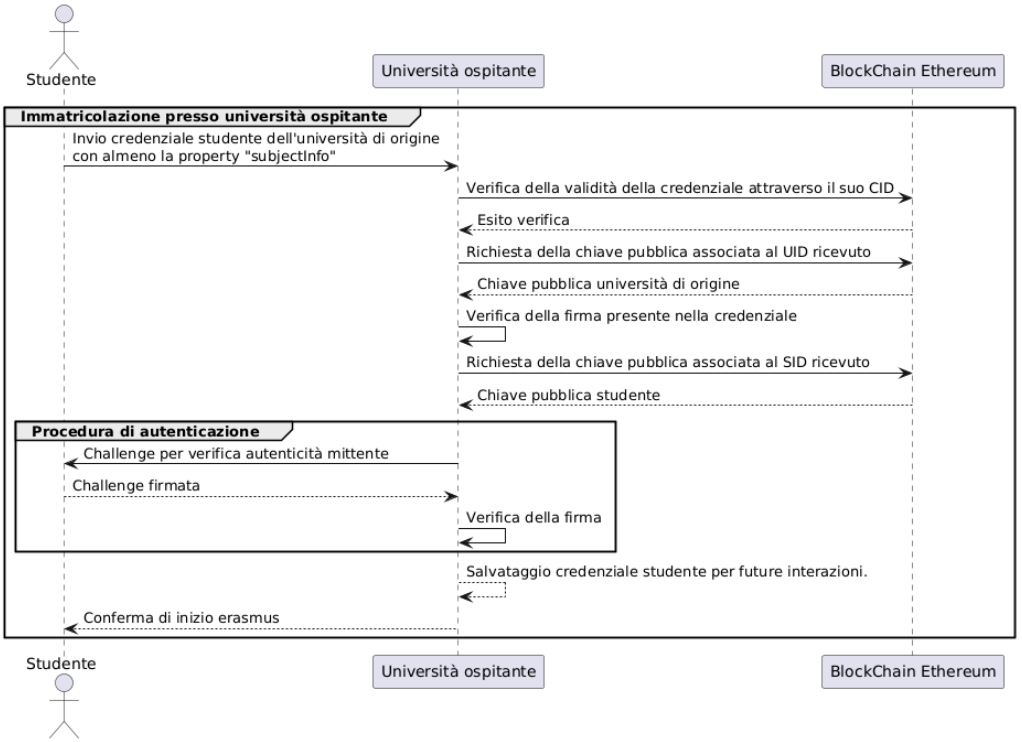
A questo punto, l'università rilascia una credenziale accademica. Questa viene registrata tramite lo smart contract delle credenziali di proprietà dell'università, permettendo allo studente di essere riconosciuto all'interno della rete universitaria. La credenziale rilasciata include lo Student ID, il Credential ID, l'ID dell'università (garantendo il non ripudio come richiesto dalla proprietà NR.1.) e i dati anagrafici dello studente.

Struttura Credenziale Accademica

Un esempio di credenziale accademica rilasciata dall'università di origine in fase di immatricolazione potrebbe essere la seguente:


```
{
   "credentialID": "CID:exampleUni:12345",
   "issuerID": "UID:authority:12345",
   "issuanceDate": "01-09-2026",
   "studentID": "SID:exampleUni1:6789",
   "properties": [
       {
           "typology": "subjectInfo",
           "data": {
               "name": "Antonio",
               "surname": "Carbone",
               "gender": "M",
               "cf": "CRBNTN02S29H163Q",
               "email": "a.carbone88@studenti.unisa.it",
               "birthDate": "29-11-2002"
           } ,
           "nonce": 48923
        }
    ],
  "issuerSignature":"ba7816bf8f01cfea414140de5dae2223b"
}
```

2.3.2. Inizio dell’esperienza Erasmus



Lo studente, a questo punto, per iniziare il proprio periodo in Erasmus deve autenticarsi anche presso l’università che lo ospiterà.

Questo procedimento ha inizio con la presentazione della credenziale accademica di cui lo studente è in possesso, in modo da attestare il suo stato di studente presso la sua università di origine. Essa viene firmata dallo studente così da garantire l’integrità delle informazioni inviate, come richiesto dalla proprietà I.1.

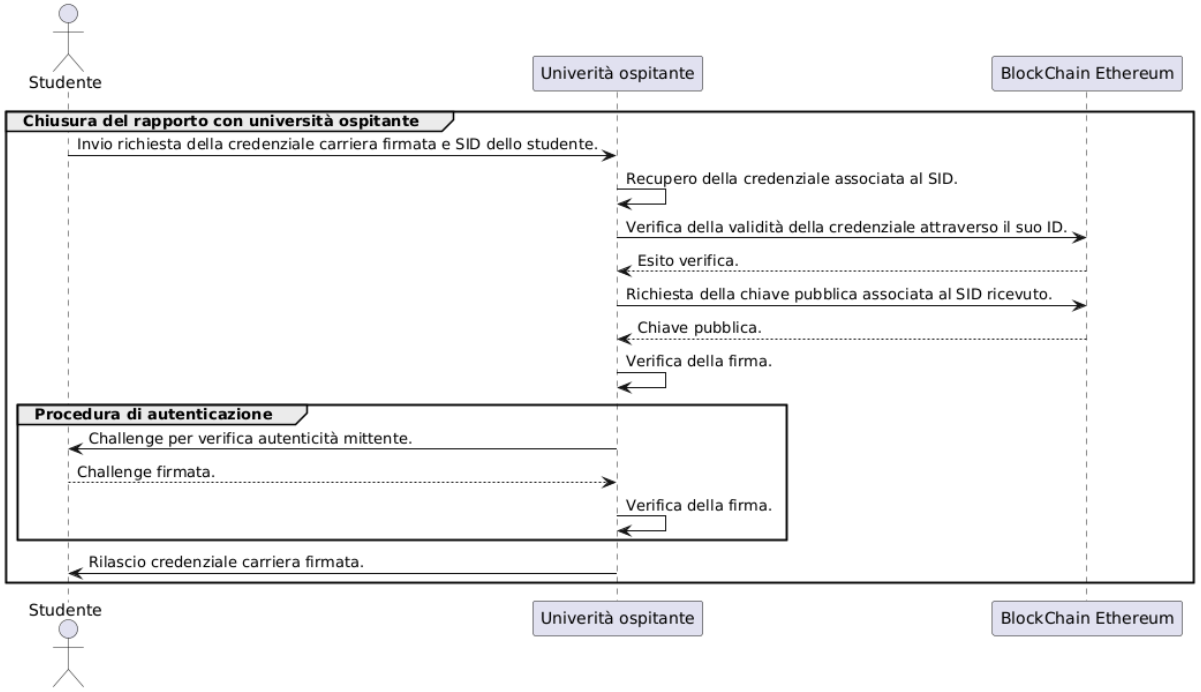
Ricevuta la credenziale, per concedere l’inizio dell’Erasmus, l’università ospitante deve verificarne il contenuto ed autenticare chi la presenta, nel rispetto delle proprietà A.1. e A.2.

Questo procedimento si compone di 3 passaggi:

1. **Verifica della validità della credenziale**
   Attraverso il CID, l’università ospitante, può verificare se la credenziale ricevuta è ancora in corso di validità. Questa verifica viene fatta attraverso una richiesta allo smart contract delle credenziali che, dato un CID, indica se la credenziale è stata o meno revocata.
2. **Verifica della firma che l’università di origine contrappone alla credenziale**
   Per eseguire questa verifica, si utilizza l’UID contenuto nella credenziale, includendolo nella richiesta rivolta allo smart contract delle entità trusted per ottenere la chiave pubblica associata ed utilizzarla per verificare la firma.
3. **Autenticazione dello studente**
   Infine, interrogando lo smart contract degli studenti utilizzando il SID ricevuto, l’università ospitante può autenticare lo studente attraverso una challenge firmata e verificata con la chiave pubblica ottenuta dalla blockchain.

Al termine di queste tre fasi, la richiesta effettuata dallo studente risulta autentica ed integra, perciò viene concesso l’inizio del periodo in erasmus presso l’università ospitante e viene salvata la credenziale dello studente per future interazioni.

2.3.3. Rilascio della Credenziale



Terminato il periodo in Erasmus, lo studente può richiedere all'università ospitante la credenziale che attesti la sua carriera.

Questa operazione viene effettuata seguendo questi 5 step:

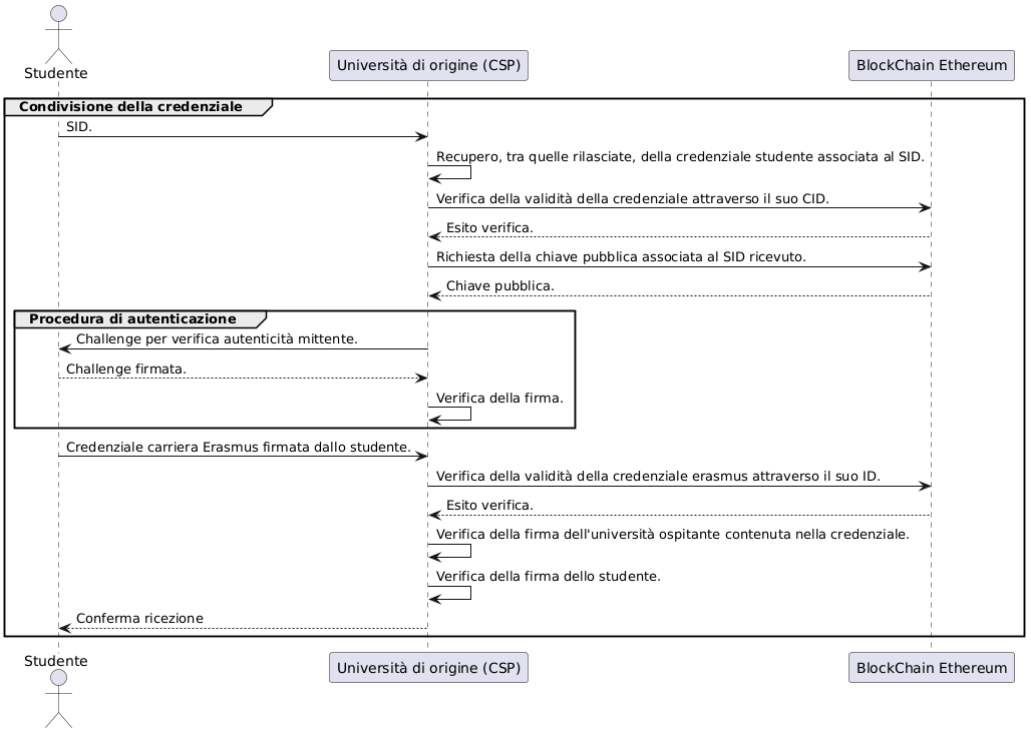
1. **Invio della richiesta di rilascio credenziale**
   Lo studente invia una richiesta della credenziale carriera, con allegato il suo SID, firmata. Questo per garantire l'integrità dell'informazione inviata come richiesto dalla proprietà I.1.
2. **Verifica validità credenziale studente**
   L'università ospitante recupera, attraverso il SID, la credenziale accademica memorizzata in fase di immatricolazione e ne verifica la validità attraverso il suo CID.
3. **Verifica della firma contrapposta alla richiesta**
   A questo punto, viene recuperata la chiave pubblica associata al SID dalla blockchain e si verifica la firma contrapposta alla richiesta.
4. **Autenticazione studente**
   Se la richiesta risulta valida, l'università ospitante procede con l'autenticazione dello studente attraverso una challenge che va firmata e verificata.
5. **Rilascio della credenziale carriera**
   Terminato il processo di verifica delle informazioni ricevute, l'università ospitante, rilascerà la credenziale carriera richiesta dallo studente contrapponendo la propria firma per garantire l'autenticità e l'integrità dell'informazione.

Struttura Credenziale Carriera

Un esempio di credenziale carriera, rilasciata dall'università ospitante al termine del periodo in Erasmus, potrebbe essere la seguente:


```
{
   "credentialID": "CID:exampleUni:12345",
   "issuerID": "UID:authority:12345",
   "issuanceDate": "01-09-2026",
   "studentID": "SID:exampleUni1:6789",
   "properties": [
       {
           "typology": "Course",
           "data": {
               "name": "Alanizee 2",
               "grade": 21,
               "cfu": 12,
               "achievementData": "12-11-2026"
           },
           "nonce": 965293
       },
       {
           "typology": "Extra Activity",
           "data": {
               "name":"Corso di robotica",
               "cfu":3,
               "date":"2010-01-01T19:23:24Z"
           },
           "nonce": 29845
       },
       {
           "typology": "Residence Info",
           "data": {
               "type":"University Residence",
               "address":"Via Roma, 54"
           },
           "nonce": 672939
       },
       {
           "typology": "Scholarship",
           "data": {
               "amount":2400,
               "unit":"Euro",
               "payments":6
           },
           "nonce": 5842439
       }
   ]
}
```

2.3.4. Condivisione della Credenziale



Lo studente, rientrato dal periodo in Erasmus, dovrà presentare la credenziale carriera ottenuta alla propria università di origine.

Questa operazione prevede le seguenti fasi:

1. **Inizio della comunicazione**
   Lo studente invia il proprio SID.
2. **Verifica validità credenziale studente**
   L'università di origine attraverso il SID, recupera tra le credenziali da essa rilasciate, quella corrispondente e ne verifica la validità facendo richiesta alla blockchain attraverso il CID.
3. **Recupero della chiave pubblica dello studente**
   Se la credenziale risulta valida, l'università recupera la chiave pubblica dello studente richiedendola alla blockchain attraverso il SID.
4. **Autenticazione studente**
   L'università di origine procede con l'autenticazione dello studente attraverso una challenge che va firmata e verificata.
5. **Comunicazione della credenziale carriera**
   Terminato il processo di verifica delle informazioni ricevute, lo studente, può condividere la propria credenziale carriera firmata.
6. **Verifica della credenziale carriera**
   Attraverso il CID della credenziale carriera, l'università interroga la blockchain per verificare la sua validità.
7. **Verifica della firma dell'università ospitante**
   Attraverso l'UID contenuto nella credenziale, l'università di origine ottiene la chiave pubblica dell'università ospitante e verifica la firma.
8. **Verifica della firma dello studente**
   Verifica la firma contrapposta, dallo studente, alla credenziale carriera appena ricevuta.

Condivisione Parziale

La credenziale, ed in particolar modo la sua struttura, è stata progettata per garantire una divulgazione selettiva delle informazioni. Per consentire un'adeguata granularità nella scelta delle informazioni da condividere, i vari campi “atomici” sono stati suddivisi in properties separate.

Ogni property, come è possibile osservare nel paragrafo 1.5, contiene un attributo che ne definisce la tipologia ed uno che ne definisce i dati variabili.

Merkle Tree

Come già anticipato, è la struttura dati di riferimento per il calcolo dell’hash della credenziale. In particolare, l’hash dell’intera credenziale si ottiene applicando l'algoritmo SHA256 nella seguente maniera:

```
1. constPart = SHA256("credentialID" + <CID…> +
           "issuerID" + <UID…> +
           "issuanceDate" + <...-...-...> +
           "studentID" + <SID…>)
2. SHA256(constPart + merkleRoot)
```

L’unica cosa che resta da definire prima di parlare del funzionamento effettivo della divulgazione selettiva è il calcolo dell’hash di una singola property, quindi delle foglie dell’albero.

Questo hash si ottiene come segue:

```
leaf = SHA256
(
   "typology" + <nome tipologia> + "data" + <data> + "name" + <nome corso>+
   "grade" + <grade num> + "cfu" + <cfu num> + "achievementData" +
   <data> + "nonce" + <nonce generato>
)
```

Divulgazione

Il possessore di una credenziale, a questo punto, può decidere di condividere solo un sottoinsieme delle sue property.

In pratica, le proprietà che si desidera rendere note vengono trasmesse in chiaro, mentre per garantire che esse non siano state manomesse, ad ogni proprietà viene associata una Merkle Proof.

Questa prova consiste in una serie di hash che, a partire dalla foglia corrispondente alla proprietà, consentono di ricostruire il percorso fino alla radice del Merkle Tree, permettendo così di verificarne l'integrità.

2.3.5. La Revoca di una Credenziale

Come richiesto dalle specifiche di progetto, la revoca è una procedura totalmente decentralizzata. A garanzia di ciò, la blockchain Ethereum viene interrogata ogni volta che l'ente che deve verificare il CID di una qualsiasi credenziale ricevuta.

Un'università che necessita di applicare la revoca di una credenziale rilasciata può usufruire delle funzioni ad hoc all'interno del proprio smart contract di rilascio credenziali. Esso consente, attraverso l'esecuzione di una funzione che imposta la validità o meno di un dato CID, di diffondere la revoca attraverso una transazione.

Quando l'ente verificatore ha la necessità di valutare una credenziale, interroga la blockchain di Ethereum facendo riferimento al contratto, a lui noto, registrato dalla Smart Contract Authority centrale.

Quest'ultimo, fa "da tramite" tra l'ente verificatore e lo smart contract che ha rilasciato la credenziale che si sta verificando.

2.3.6. La Revoca o modifica di uno Student Identifier

La revoca di un SID è una procedura che coinvolge l'università che ne ha eseguito il rilascio. Quest'ultima, attraverso la funzione di revoca del proprio SID smart contract, può annullare la validità di un SID.

Risulta possibile, inoltre, modificare la chiave pubblica associata ad un SID, attraverso un'altra opportuna funzione dello smart contract.

2.3.7. La Revoca o modifica di un University Identifier

La Smart Contract Authority può decidere di impedire l'utilizzo del sistema ad un'università, segnalando la non validità della sua chiave pubblica a chiunque la richieda, attraverso una apposita funzionalità dello smart contract.

Quando si richiedono informazioni legate ad un CID o un SID registrati da un'università il cui utilizzo del sistema è stato bloccato, allora la richiesta viene annullata, segnalando che l'ente di provenienza del documento o dell'identità non presenta un UID valido.

La Smart Contract Authority ha inoltre la possibilità di eseguire modifiche alle informazioni associate ad un UID, potendo cambiare ad esempio la chiave pubblica associata o gli indirizzi dei contratti di verifica della data università.

2.3.8. La confidenzialità del sistema

Alla base del funzionamento del sistema, si presuppone la presenza di:

- un canale di comunicazione sicuro che preservi la confidenzialità adottando la tecnologia HTTPS.
- uno storage sicuro delle credenziali e della chiave privata dello studente, tramite un applicativo che, svolgendo il ruolo di portafogli digitale delle credenziali, possa utilizzare tecniche di: autenticazione biometrica, pin o password.

WP3 - Analisi di Sicurezza

3.1. Introduzione

In questo work package verrà analizzato il livello di sicurezza garantito dalla soluzione illustrata nel work package 2.

Quest'analisi considera come riferimento per la valutazione il numero di proprietà che la soluzione riesce a soddisfare.

In particolare, ogni proprietà sarà considerata soddisfatta quando il sistema sarà resiliente rispetto a tutti i threat model che la minacciano.

Crittografia Asimmetrica

Prima, però, di procedere con l'analisi di sicurezza bisogna fare un appunto sullo scenario più estremo che potrebbe colpire l'integrità delle informazioni scambiate tra le parti oneste del sistema, ovvero il furto di una chiave privata.

Ciò rappresenta chiaramente uno stato irreversibile ed un grande vantaggio per un qualsiasi attaccante del sistema.

In questo caso, l'unica soluzione che l'architettura proposta mette a disposizione è la possibilità di aggiornare lo stato dell'identificativo legato al possessore della chiave compromessa, rendendolo invalido e rilasciandone uno nuovo legato ad una nuova coppia di chiavi generata.

3.2. Confidenzialità dei dati

Le proprietà C.1. e C.2. indicano che sono stati individuati due modi in cui la confidenzialità può essere compromessa:

1. Se la comunicazione non prevede un meccanismo per preservarla, allora il sistema fallisce nel tentativo.
2. Se le informazioni riservate non sono immagazzinate nella maniera corretta, allora non è possibile definire l'architettura proposta come confidenziale.

È già possibile definire quelle che sono le assunzioni che permettono di garantire queste due proprietà, infatti, l'adozione del protocollo di comunicazione HTTPS per le comunicazioni tra le parti del sistema e l'utilizzo di una metodologia di storage cifrato assicurano il rispetto della confidenzialità.

Di seguito vengono analizzati i due threat model individuati per questo tipo di proprietà.

3.2.1. Man in the Middle Passivo

Questo tipo di minaccia perde di senso nel momento in cui qualsiasi tentativo di eavesdropping risulta difficile e con scarse possibilità di successo.

L'utilizzo di HTTPS come protocollo di comunicazione, quindi TLS come protocollo di cifratura delle informazioni, infatti, rende sicuro lo scambio di dati tra le parti oneste del sistema.

3.2.2. Ente Verificatore Curioso

Se il rispetto delle proprietà C.1. e C.2. risolvono automaticamente il precedente threat model, per questo bisogna aggiungere un passaggio.

Infatti, in questo scenario il problema non si presenta durante la comunicazione, bensì nel momento in cui una parte apparentemente onesta riceve una credenziale i cui campi sono parzialmente divulgati.

Quello che si vuole evitare è che chi riceve una credenziale possa accedere alle informazioni nascoste, questo è possibile grazie al meccanismo delle merkle proof che permettono di divulgare in chiaro solamente ciò che il possessore della credenziale vuole condividere, il resto viene trasmesso sotto forma di hash.

Gli hash che possono interessare ad un ente verificatore curioso sono le foglie coinvolte nella prova di una contenente il campo che si sta trasmettendo in chiaro. Con questo tipo di conoscenza, un attaccante di questo tipo può fare solo inferenza a partire dalla conoscenza della struttura del campo trasmesso sotto forma di hash, ma ciò non rappresenta un problema dal momento che nelle proprietà della credenziale è stato inserito un campo ‘nonce’ che aggiunge variabilità all’hashing.

3.3. Integrità dei dati

Come anticipa la proprietà I.1., all’interno del nostro sistema, si considera soddisfatta la proprietà di integrità dei dati se nessun elemento, onesto o meno, riesce a manomettere le informazioni contenute nelle credenziali rilasciate o in generale scambiate tra le parti oneste del sistema.

Data l’elevata quantità di threat model che possono compromettere questa caratteristica e dato che ognuno di essi necessita di contromisure leggermente differenti, di seguito, li analizzeremo uno alla volta illustrando quelle che sono le decisioni prese per evitarli.

3.3.1. Utente Certificato Malintenzionato

Questo tipo di threat model, come anche altri legati a questa proprietà, punta a modificare il contenuto delle comunicazioni tra le parti oneste del sistema.

In particolare, ciò che si vuole modificare, in questo caso volontariamente, è il contenuto di una credenziale. Un apparente vantaggio in questo caso potrebbe essere che a fare l’attacco è il possessore di una credenziale, quindi una figura fidata all’interno del sistema.

Questo vantaggio è apparente perché, grazie al meccanismo delle firme con chiave privata, la modifica viene identificata allo stesso modo se a farla è il possessore o una figura terza come nel caso del Man in the Middle Attivo.

Quindi, l’università che rilascia la credenziale, firmandola, rende impossibile la modifica del contenuto dal momento che grazie al meccanismo dell’hash and sign qualsiasi alterazione del contenuto rende incompatibili la firma contrapposta alla credenziale e quella calcolata dall’algoritmo di verifica.

3.3.2. Man in the Middle Attivo

Come appena anticipato, questo threat viene completamente evitato dal meccanismo delle firme con chiave privata.

L'unica differenza dal threat model precedente è che la modifica non viene effettuata direttamente dal possessore della credenziale, bensì da un terzo che è riuscito ad intercettarne ed alterarne la condivisione. Ma come già detto questa differenza non aggiunge o toglie potere alla minaccia.

3.3.3. Ente Certificatore Fasullo

In questo caso, siamo di fronte ad un threat model che, in base alla soluzione precedentemente proposta, non può sussistere. La presenza di una gerarchia di enti fidati ci garantisce che una terza entità non riconosciuta dalla Smart Contract Authority non può registrare credenziali valide e perciò una qualsiasi credenziale da essa forgiata non verrà mai accettata da nessun ente verificatore, rendendo impossibile la circolazione di credenziali fake.

3.3.4. Ente Certificatore Malintenzionato

Se in tutti gli altri casi le minacce possono essere evitate, in questo si può solo fare in modo che non accada più. Questo perché, di fronte ad un'università certificata dalla Smart Contract Authority e capace, quindi, di rilasciare certificati verificabili dalle altre università della rete, non si può escludere a priori la possibilità che questa rilasci credenziali il cui contenuto non corrisponde alla realtà.

L'unica contromisura che il sistema progettato ha a disposizione è la revoca dello stato di università fidata a seguito di segnalazioni da parte degli studenti da essa frodati.

3.3.5. Forgiatore di Credenziali

Questo tipo di minaccia viene considerata nel momento in cui viene, giustamente, utilizzato un formato standard per le credenziali rilasciate che possa far sorgere l'idea di creare autonomamente una credenziale ed utilizzarla.

Questo threat model è stato considerato in fase di progettazione della soluzione che ne risulta, infatti, immune per due principali motivi:

1. Il meccanismo delle firme prevede che chi rilascia la credenziale deve firmarla con la propria chiave privata, quindi un qualcuno che non sia riconosciuto come fidato dalla Smart Contract Authority non può farlo.
2. La presenza di un'autorità centrale, appunto, garantisce che solo chi ha un identificativo registrato ed associato alla propria chiave pubblica può rilasciare credenziali.

3.3.6. Revoca Inefficace

La soluzione proposta, basata sull'utilizzo della tecnologia blockchain per la gestione delle revoche, rende tale evenienza estremamente rara.

L'affidabilità e la solidità di questa tecnologia assicurano infatti che tutte le università appartenenti alla rete siano tempestivamente informate dell'eventuale revoca di una credenziale.

Non si può nemmeno attribuire il problema a un malfunzionamento dell'operazione di revoca, poiché qualsiasi errore viene immediatamente notificato al soggetto che la effettua, consentendogli di ripetere l'operazione fino al buon esito.

3.4. Autenticazione

Nel work package 1, le proprietà A.1. e A.2. suggeriscono che ci sono due modi di interpretare il termine autenticazione. Il primo si riferisce alla possibilità di poter garantire e verificare l'autenticità delle informazioni che qualsiasi ente del sistema può ricevere. La seconda, invece, si riferisce al concetto più comune di autenticazione, ovvero la garanzia sull'identità delle figure coinvolte in uno scambio di informazioni.

La soluzione proposta nella sezione precedente del documento illustra diversi meccanismi che possono garantire l'autenticazione in entrambe le accezioni del termine.

Autenticazione delle parti del sistema

È possibile dimostrare la validità dell'identità delle parti coinvolte grazie a due meccanismi adoperati nella soluzione.

1. L'assunzione che, a garanzia dell'identità di uno studente in fase di immatricolazione ci sia lo SPID, permette a chiunque di potersi fidare dei dati associati allo studente dalla propria università di origine.
2. La presenza Smart Contract Authority dà, invece, la garanzia sulla bontà dell'operato delle università della rete Erasmus.
3. Infine, la blockchain Ethereum ed il meccanismo degli identificativi, rappresentano il modo attraverso il quale chiunque può verificare l'identità ed il ruolo di qualsiasi parte della rete.

Autenticazione delle informazioni

Per quanto riguarda, invece, l'autenticità delle informazioni contenute in una credenziale, può essere garantita dall'utilizzo del meccanismo di firma con chiave privata.

Infatti, basta applicare l'algoritmo di hash and sign alla credenziale o ad un'informazione qualsiasi che si vuole condividere per far sì che qualsiasi modifica venga rilevata.

3.4.1. Sosia

Le soluzioni appena descritte consentono al sistema di prevenire scenari in cui un attore malevolo, in possesso di credenziali ottenute illecitamente, riesca a spacciarsi per il legittimo titolare.

In particolare, le scelte progettuali descritte nel paragrafo sull'autenticazione delle parti, insieme all'impiego di challenge firmate, assicurano una protezione efficace contro attacchi riconducibili al threat model da noi definito 'Sosia'.

3.5. Non ripudio

Il non ripudio, nel caso in questione, sta nell'impedire a qualsiasi elemento del sistema di rinnegare qualcosa da esso condivisa o generata.

In particolare ciò può avere un forte impatto sulle credenziali rilasciate agli studenti e questo può accadere sia dal lato di chi le rilascia (università) sia dal lato di chi le possiede e le condivide (studenti).

Infatti, le proprietà NR.1. ed NR.2. vogliono esattamente che si eviti lo scenario in cui un'università che rilascia un certificato possa rinnegarlo nel momento in cui viene interpellata per la verifica, oppure che uno studente in possesso di una credenziale possa rinnegarne la condivisione o il contenuto.

Il meccanismo degli identificativi adottato dal sistema proposto impedisce il verificarsi di queste due situazioni, di seguito un'analisi dettagliata di entrambi gli scenari.

3.5.1. Ente Certificatore Rinnegante

Dal momento che ogni credenziale presenta, oltre alla firma, l'identificativo dell'università che l'ha rilasciata, quest'ultima non può negare di essere l'autrice di tale documento.

L'atto della verifica di una credenziale, secondo il funzionamento del sistema, non coinvolge direttamente l'università che l'ha generata, bensì lo smart contract che essa ha utilizzato per farlo.

3.5.2. Possessore di Credenziale Rinnegante

Praticamente allo stesso modo, uno studente possessore di una credenziale non può rinnegarne il contenuto o la condivisione dato che essa presenta il suo SID, oltre che la sua firma, a garanzia della sua identità.