Università degli Studi di Salerno

.DIEM

Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica Applicata

Corso di Laurea Magistrale in Ingegneria Informatica

Algoritmi e Protocolli per la Sicurezza
a.a. 2024/2025

Project Work
Gruppo n. 06 – AH

Cognome e Nome	Matricola	e-mail
Cirillo Francesco Pio	0622702466	f.cirillo36@studenti.unisa.it
Fasolino Alessandra	0622702465	a.fasolino35@studenti.unisa.it

WP1

Attori onesti e loro Obiettivi

Di seguito sono indicati i 3 attori onesti individuati durante il WP1 e i relativi obiettivi perseguiti durante l'uso del software prodotto.

Studente

- Ricevere credenziali certificate dalla sua Università di Origine;
- Ricevere credenziali certificate dalla sua Università Ospitante;
- Inviare selettivamente credenziali certificate alla sua Università di Origine;
- Inviare selettivamente credenziali certificate alla sua Università Ospitante.

Università Ospitante

- Creazione credenziali certificate;
- Verificare validità credenziali certificate;
- Revocare certificati emessi.

Università di Origine

- Creazione credenziali certificate;
- Verificare validità credenziali certificate;
- Revocare certificati emessi.

Funzionalità che si intende realizzare

- Invio di credenziali;
- Richiesta di credenziali;
- Check dell'identità dell'interlocutore;
- Check delle credenziali richieste;
- Check Revoca;
- Protocollo di distribuzione sicura della chiave di sessione per verificare l'autenticità.

Avversari (threat model) + Obiettivi/attacchi degli avversari

Threat Model 1: Adulterazione delle credenziali di uno studente

Obiettivo: modificare le credenziali di uno studente in modo da fornirgli, a pagamento, un aumento della media degli esami.

Impatto: incapacità di riconoscere eventuali adulterazioni nel contenuto del messaggio.

Attaccante: studente malintenzionato.

Risorse: accesso alla propria chiave privata e a tutti i propri certificati, relativi alle credenziali e all'identità.

Asset da proteggere: integrità delle credenziali.

Threat Model 2: Furto di credenziali di uno studente fingendosi tale

Obiettivo: fingersi uno studente al fine di ottenere le sue credenziali chiedendole all'università e poi mettere in atto un furto d'identità.

Impatto: divulgazione illecita di informazioni private di studenti a terze parti.

Attaccante: conoscente dello studente.

Risorse: accesso al Certificato di Identità dello studente ottenuto per mezzo di una comunicazione precedente con lo stesso.

Asset da proteggere: privacy dello studente.

Threat Model 3: Furto di credenziali di uno studente fingendosi un'università

Obiettivo: fingersi un'università per ricevere le credenziali di uno studente e mettere in atto la vendita delle credenziali dello studente.

Impatto: divulgazione illecita di informazioni private di studenti a terze parti.

Attaccante: hacker professionista.

Risorse: accesso al Certificato di Identità dell' Università ottenuto per mezzo di una comunicazione precedente con lo stesso.

Asset da proteggere: privacy dello studente.

Threat Model 4: Man in the middle durante lo scambio di chiavi

Obiettivo: ottenere la chiave di sessione

Impatto: perdita di sicurezza nello scambio di informazioni da ambo le parti oneste.

Attaccante: hacker professionista.

Risorse: accesso attivo alla comunicazione.

Asset da proteggere: chiave di sessione per la comunicazione simmetrica.

Threat Model 5: Invio di informazioni incoerenti ma valide

Obiettivo: inviare Credenziali valide e certificate ma sfruttare la divulgazione selettiva al fine di inviare informazioni incoerenti per trarne vantaggio, ad esempio condividendo il nome di un esame solo frequentato e il voto di un esame conseguito.

Impatto: rischio di convalidare un esame non sostenuto.

Attaccante: studente che vuole “rubare” un esame.

Risorse: accesso alla propria chiave privata e a tutti i propri certificati, relativi alle credenziali e all’identità.

Asset da proteggere: validità delle informazioni scambiate.

Threat Model 6: Cambio dello stato di revoca di un certificato non revocato

Obiettivo: modificare lo stato di revoca di un certificato per fare in modo da far risultare revocato un certificato valido.

Impatto: perdita di fiducia nell’università e nei certificati da essa rilasciati.

Attaccante: terza parte malintenzionata.

Risorse: essere un proponente della blockchain che ospita la lista delle revoche.

Asset da proteggere: credibilità dell’università e credibilità delle certificazioni.

Threat Model 7: Cambio dello stato di revoca di un certificato revocato

Obiettivo: modificare lo stato di revoca di un certificato per fare in modo da far risultare non revocato un certificato revocato.

Impatto: perdita di fiducia nell’università e nei certificati da essa rilasciati.

Attaccante: terza parte malintenzionata.

Risorse: essere un proponente della blockchain che ospita la lista delle revoche.

Asset da proteggere: credibilità dell’università e credibilità delle certificazioni.

Proprietà che si vorrebbe poter preservare

- confidenzialità;
- integrità;
- non ripudio;
- resilienza (robustezza);
- autenticazione.

Definizione della struttura (campi) della credenziale


```
{
    "matricola_casa": "0123456789",
    "matricola_ospitante": "0123456789",
    "nome": "Mario",
    "cognome": "Rossi",
    "email_casa": "mario.rossi@studenti.casa.it",
    "email_ospitante": "mario.rossi@etudiant.maison.fr",
    "data_di_nascita": "01/01/2000",

    "codice_corso_di_laurea": "LM-32",
    "nome_corso_di_laurea": "Magistrale in Ingegneria Informatica",
    "cfu_totali_conseguiti": 100,
    "media_voti": 28,

    "attestazioni_di_frequenza": [
      {
        "id_oggetto": "attestazioni_di_frequenza_65468481",
        "nome_corso": {
          "id_oggetto": "attestazioni_di_frequenza_65468481",
          "nome_corso": "Analisi Matematica 1"
        },
        "percentuale_frequentata": {
          "id_oggetto": "attestazioni_di_frequenza_65468481",
          "percentuale_frequentata": 87
        }
      },
      {
        "id_oggetto": "attestazioni_di_frequenza_65468259",
        "nome_corso": {
          "id_oggetto": "attestazioni_di_frequenza_65468259",
          "nome_corso": "Fisica 1"
        },
        "percentuale_frequentata": {
          "id_oggetto": "attestazioni_di_frequenza_65468259",
          "percentuale_frequentata": 78
        }
      }
    ],

    "titoli_di_studio": [
      {
        "id_oggetto": "titoli_di_studio_121548448",
        "nome": {
```

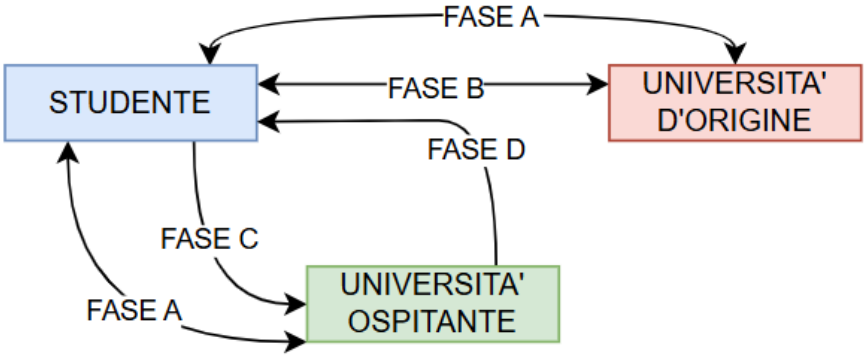
```
          "id_oggetto": "titoli_di_studio_121548448",
          "nome": "Laurea Triennale in Ingegneria Informatica"
        },
        "voto": {
          "id_oggetto": "titoli_di_studio_121548448",
          "voto": 110
        },
        "lode": {
          "id_oggetto": "titoli_di_studio_121548448",
          "lode": true
        }
      },
      {
        "id_oggetto": "titoli_di_studio_121548158",
        "nome": {
          "id_oggetto": "titoli_di_studio_121548158",
          "nome": "Diploma di Liceo Scientifico Indirizzo Scienze
Applicate"
        },
        "voto": {
          "id_oggetto": "titoli_di_studio_121548158",
          "voto": 100
        },
        "lode": {
          "id_oggetto": "titoli_di_studio_121548158",
          "lode": true
        }
      }
    ],

    "esami_conseguiti": [
      {
        "id_oggetto": "esami_conseguiti_121548484",
        "codice_esame": {
          "id_oggetto": "esami_conseguiti_121548484",
          "codice_esame": "123"
        },
        "nome_esame": {
          "id_oggetto": "esami_conseguiti_121548484",
          "nome_esame": "Algoritmi e Protocolli per la Sicurezza"
        },
        "numero_cfu": {
          "id_oggetto": "esami_conseguiti_121548484",
          "numero_cfu": 9
        },
        "conseguito": {
```

```
          "id_oggetto": "esami_conseguiti_121548484",
          "conseguito": false
        },
        "data_conseguimento": {
          "id_oggetto": "esami_conseguiti_121548484",
          "data_conseguimento": null
        },
        "voto": {
          "id_oggetto": "esami_conseguiti_121548484",
          "voto": null
        },
        "professore_responsabile": {
          "id_oggetto": "esami_conseguiti_121548484",
          "professore_responsabile": "Carlo Mazzocca"
        },
        "universita_di_conseguimento": {
          "id_oggetto": "esami_conseguiti_121548484",
          "universita_di_conseguimento": "Università degli Studi di
Salerno"
        }
      },
      {
        "id_oggetto": "esami_conseguiti_487845485",
        "codice_esame": {
          "id_oggetto": "esami_conseguiti_487845485",
          "codice_esame": "124"
        },
        "nome_esame": {
          "id_oggetto": "esami_conseguiti_487845485",
          "nome_esame": "Analyse des données"
        },
        "numero_cfu": {
          "id_oggetto": "esami_conseguiti_487845485",
          "numero_cfu": 9
        },
        "conseguito": {
          "id_oggetto": "esami_conseguiti_487845485",
          "conseguito": true
        },
        "data_conseguimento": {
          "id_oggetto": "esami_conseguiti_487845485",
          "data_conseguimento": "15_01_2025"
        },
        "voto": {
          "id_oggetto": "esami_conseguiti_487845485",
          "voto": 30
```

```
        },
        "professore_responsabile": {
          "id_oggetto": "esami_conseguiti_487845485",
          "professore_responsabile": "Jean-Pierre"
        },
        "universita_di_conseguimento": {
          "id_oggetto": "esami_conseguiti_487845485",
          "universita_di_conseguimento": "Université de Rennes"
        }
    }
    ]
}
```

WP2

Sistema di gestione delle credenziali + Azioni delle parti oneste



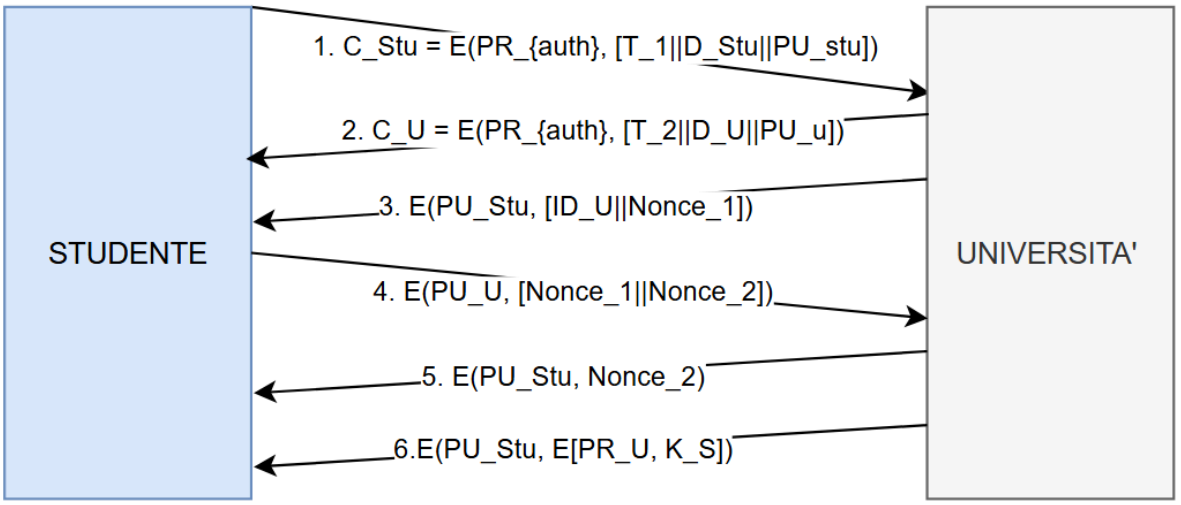
Precondizioni:

- gli utenti e le Università sono accreditati presso la CA, ognuno di loro dispone di un certificato che leghi la propria identità ad una chiave pubblica;
- ogni Università dispone di un database contenente le credenziali degli studenti presso di loro accreditati;
- la firma avviene utilizzando un Padding appropriato in modo da creare una struttura rigida al fine di impedire attacchi senza messaggio e assenza di relazioni moltiplicative per l'attacco di falsificazione con messaggi arbitrari;
- presenza di un sistema di revoche basato su blockchain, ogni evento di revoca viene registrato sulla blockchain e se si desidera controllare se un certificato è stato revocato basta controllare se il suo codice identificativo è presente sulla blockchain.

Funzionamento:

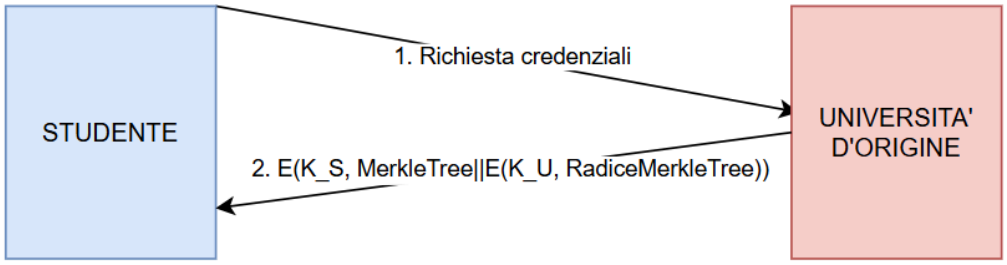
A. Inizio comunicazione tra studente e università
    - Se non si è in possesso dei rispettivi certificati di identità lo studente invia il proprio Certificato di Identità all'Università e viceversa, in modo da proteggere l'integrità della trasmissione per mezzo dell'uso delle chiavi pubbliche.
    - Al fine di garantire l'autenticazione si usa il protocollo di distribuzione sicura della chiave di sessione.
        - L'università invia un codice casuale allo studente crittografato con la chiave pubblica dello studente stesso, lo studente deve decriptarlo con la propria chiave privata e poi rimandarlo all'università criptato con la chiave pubblica della stessa, a questo aggiunge anche un altro Nonce crittografato che si aspetta di ricevere per avere certezza di stare interloquendo con l'università.

FASE A: INIZIO COMUNICAZIONE STUDENTE - UNIVERSITA'



B. Richiesta certificato all'università
   - Inizio comunicazione tra studente e università
   - Lo studente chiede le proprie credenziali
   - L'università recupera dal proprio database i dati dello studente e compone un Merkle Tree che ha per foglie le credenziali dello studente, cripta la radice dell'albero con la propria chiave privata, in questo modo è salvaguardata l'integrità dell'informazione.
   - L'università crea il certificato accoppiando il Merkle Tree e la radice criptata.
   - L'università invia il certificato, allo studente. Per fini di confidenzialità questa trasmissione avviene in forma criptata usando la chiave di sessione.
   - Lo studente riceve le informazioni e le decripta con la chiave di sessione.

FASE B: RICHIESTA CERTIFICATO ALL'UNIVERSITA'

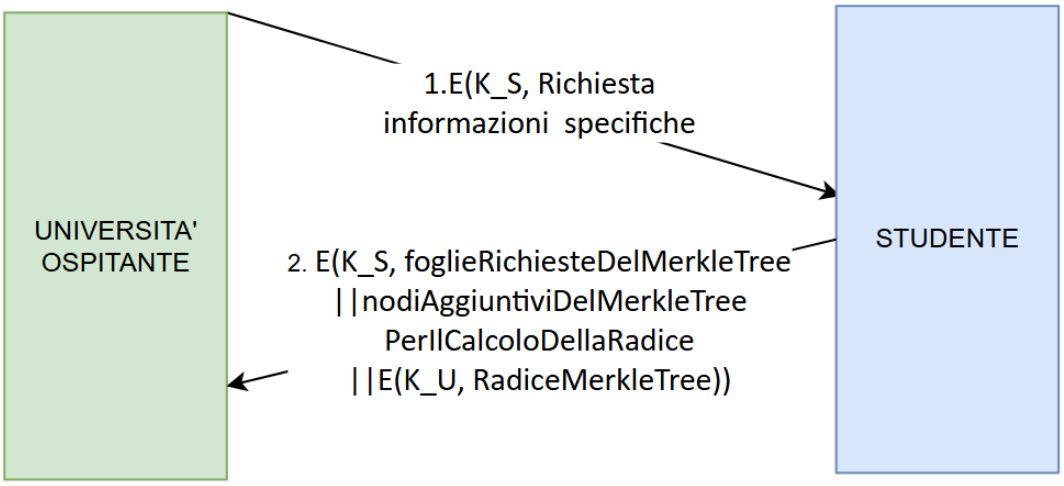


C. Invio certificato all'università
   - Inizio comunicazione tra studente e università
   - Lo studente estrapola dal Merkle Tree le foglie relative alle informazioni richieste dall'università, nel caso queste informazioni non bastino al calcolo della radice del Merkle Tree lo studente estrapola anche gli hash necessari al completamento del calcolo in questione, avendo cura di sceglierli in modo da minimizzare il numero di elaborazioni necessarie lato ricevente.
   - Lo studente invia il certificato, con le sole informazioni strettamente necessarie precedentemente estrapolate, all'università richiedente. Per

salvaguardare la confidenzialità tutte queste informazioni sono criptate con la chiave di sessione.

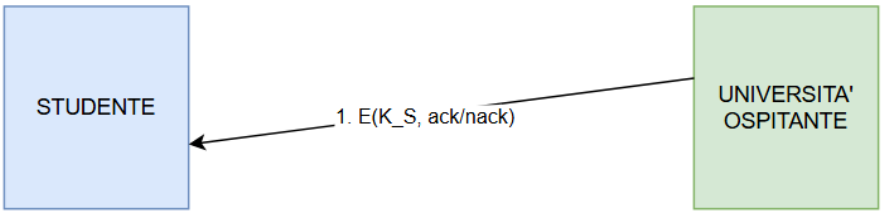
- L'università riceve il ciphertext e lo decripta con la chiave di sessione.
- Verifica certificato.

FASE C: INVIO CERTIFICATO ALL'UNIVERSITA'



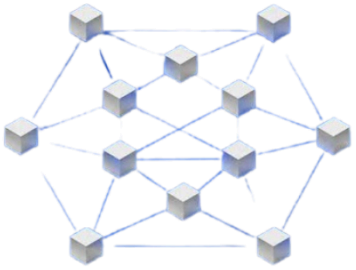
D. Verifica certificato
   - Si realizza il calcolo della radice del Merkle Tree sulla base delle informazioni fornite, dopodiché si decripta la firma del certificato usando la chiave pubblica dell'università emissaria. A questo punto si è in possesso di due digest e si procede a verificare la corrispondenza tra i due, se la corrispondenza è verificata le informazioni sono valide. (Se non si è in possesso del certificato di identità dell'altra università lo si richiede, fase a).
   - Se il certificato è valido l'università verifica se il suo codice è presente sulla blockchain, il che indicherebbe che il certificato è stato revocato.
   - Si comunica allo studente se il certificato è risultato valido o meno.

FASE D: VERIFICA CERTIFICATO



E. Revoca di un certificato
   - Se un'università vuole revocare un certificato da essa emesso può effettuare l'aggiunta alla Certificate Revocation List presente sulla blockchain del codice identificativo del certificato da revocare. Da questo momento in poi quando un'università riceve il certificato in questione quando effettuerà il controllo sulla blockchain troverà il suo codice e quindi saprà che è stato revocato.

FASE E: REVOCA CERTIFICATO



Aggiunta alla CRL sulla blockchain del codice del certificato da revocare

UNIVERSITA' D'ORIGINE

WP3

Analisi della sicurezza

Threat Model 1: Adulterazione delle credenziali di uno studente

Il sistema sviluppato risulta in grado di impedire l'adulterazione delle credenziali di uno studente, ad esempio al fine di aumentare la media dei suoi esami, da parte dello studente malintenzionato che ovviamente è in possesso della propria chiave privata e di tutti i propri certificati.

Ciò è vero in quanto l'impiego della chiave privata permette all'attaccante di instaurare la connessione con l'università in maniera corretta ma comunque la modifica delle credenziali porta ad un'alterazione della radice del Merkle Tree corrispondente. Quindi la radice del Merkle Tree non sarà uguale a quella che si ottiene dalla decriptazione della firma del certificato usando la chiave pubblica dell'università; ciò evidenzia la manomissione delle credenziali trasmesse, salvaguardando l'integrità.

L'unico modo in cui questo attacco potrebbe funzionare è se l'attaccante riuscisse a trovare credenziali diverse ma comunque in grado di produrre lo stesso hash di certificati esistenti. Anche ricordando che l'attaccante è già in possesso di tutti i propri certificati, riuscire a trovare nuove credenziali con lo stesso Hash è, in termini realistici, impossibile viste le caratteristiche di non collisione e non invertibilità dell'Hash.

Threat Model 2: Furto di credenziali di uno studente fingendosi tale

Il sistema sviluppato risulta in grado di impedire ad un malintenzionato che è in possesso del Certificato di Identità dello studente di impersonare lo studente in una comunicazione con l'università al fine di ottenere le credenziali riservate per poterle divulgare a terze parti.

Ciò è vero in quanto per iniziare la comunicazione non basta il semplice scambio dei certificati di identità, vi è infatti poi l'avvio di un protocollo sicuro di scambio della chiave di sessione che prevede che ambo le parti dimostrino di essere davvero chi dicono di essere decriptando un Nonce criptato con la chiave pubblica da loro fornita.

Lo schema sarebbe violato solo se l'attaccante fosse in possesso della chiave privata dello studente, ma questo è al di là delle risorse a sua disposizione in questo Threat Model.

Quanto detto permette di affermare che il sistema è in grado di salvaguardare la privacy dello studente.

Threat Model 3: Furto di credenziali di uno studente fingendosi un'università

Il sistema sviluppato risulta in grado di impedire che un hacker possa impersonare un'università per farsi inviare le credenziali di uno studente, essendo in possesso del Certificato di Identità dell'Università grazie a comunicazioni pregresse.

Ciò è vero in quanto anche essendo in possesso del Certificato di Identità dell'Università l'hacker non potrà terminare la configurazione della comunicazione, in quanto a questo scopo è necessario che dopo lo scambio dei certificati venga eseguito il protocollo di distribuzione sicura della chiave.

Questo protocollo impedisce questa tipologia di attacchi in quanto richiede che le parti coinvolte dimostrino di essere chi dicono per mezzo della decriptazione di un Nonce criptato con la chiave pubblica da loro fornita.

Per superare il protocollo di distribuzione sicura della chiave l'attaccante dovrebbe essere in possesso della chiave privata dell'università, il che è oltre le risorse specificate nel Threat Model.

Il sistema risulta quindi in grado di tutelare la privacy dello studente.

Threat Model 4: Man in the middle durante lo scambio di chiavi

Il sistema sviluppato risulta in grado di impedire ad un hacker che ha accesso attivo al canale di comunicazione di ottenere la chiave di sessione per mezzo di un attacco Man in the Middle.

Il sistema infatti prevede che le parti coinvolte dimostrino la loro identità per mezzo della decriptazione di Nonce crittografati con la loro chiave pubblica. Questo processo, che fa parte del protocollo di distribuzione sicura delle chiavi, impedisce l'attacco Man in the Middle.

Un attaccante intenzionato a mettere in atto questa tipologia di attacco dovrebbe disporre delle chiavi private delle parti comunicanti per riuscire a superare la fase di set-up della comunicazione, questo va al di là delle risorse di questo Threat Model.

Il sistema protegge quindi la chiave di sessione per la comunicazione simmetrica e tutto ciò che ne consegue.

Threat Model 5: Invio di informazioni incoerenti ma valide

Il sistema sviluppato risulta in grado di impedire l'invio di informazioni incoerenti ma valide da parte di un attaccante che vuole mostrare credenziali migliori di quelle di cui è in possesso in realtà.

Ciò è vero in quanto se si invia il nome di un esame e il voto di un altro, il ricevente confermerà che il certificato è valido ma sarà in grado di constatare che le informazioni sono incoerenti in quanto le informazioni composite sono contrassegnate da un ID che le lega e il nome dell'esame e il voto dell'esempio avranno ID diversi.

L'attacco potrebbe funzionare solo se fosse possibile modificare l'ID di uno dei due oggetti per renderlo uguale all'altro, questo però altererebbe la radice del Merkle Tree invalidando il certificato.

Il sistema quindi salvaguarda la validità e coerenza delle informazioni scambiate.

Threat Model 6: Cambio dello stato di revoca di un certificato non revocato

Il sistema sviluppato risulta in grado di impedire modifiche dello stato di revoca al fine di rendere revocato un certificato se l'attaccante è una terza parte malintenzionata che ha avuto modo di impossessarsi di Certificati revocati e non dello studente specifico.

Ciò è vero in quanto anche essendo proponente, se si propone un blocco non corretto alla blockchain gli attestatori voteranno contro la sua aggiunta alla blockchain.

Il sistema risulta quindi in grado di tutelare la consistenza della revoca.

Threat Model 7: Cambio dello stato di revoca di un certificato revocato

Il sistema sviluppato risulta in grado di impedire modifiche dello stato di revoca al fine di rendere non revocato un certificato revocato se l'attaccante è una terza parte malintenzionata che ha avuto modo di impossessarsi di Certificati revocati e non dello studente specifico.

Ciò è vero in quanto anche essendo un proponente per la blockchain resta il fatto che si tratta di una struttura dati append-only e non sono quindi possibili modifiche a dati già aggiunti.

Il sistema risulta quindi in grado di tutelare la consistenza della revoca.

WP4

Implementazione

Lo sviluppo del software è avvenuto in accordo ai principi dell'OOP in modo da favorire un prodotto che fosse leggibile (in modo da agevolare la correzione), manutenibile e con parti potenzialmente riutilizzabili (soprattutto la classe CryptoUtils).

Si precisa che per semplicità è stata implementata una versione delle credenziali accademiche che include solo le informazioni di base.

Di seguito il json delle credenziali accademiche effettivamente modellate in codice:

```
{
    "matricola_casa": "0123456789",
    "matricola_ospitante": "0123456789",
    "nome": "Mario",
    "cognome": "Rossi",
    "email_casa": "mario.rossi@studenti.casa.it",
    "email_ospitante": "mario.rossi@etudiant.maison.fr",
    "data_di_nascita": "01/01/2000",

    "codice_corso_di_laurea": "LM-32",
    "nome_corso_di_laurea": "Magistrale in Ingegneria Informatica",
    "cfu_totali_conseguiti": 100,
    "media_voti": 28,
}
```

Segue una descrizione precisa dei vari moduli.

Actors

Il modulo actors contiene le classi che rappresentano gli attori del programma.

Questi sono:

- CertifiedCommunicatingParty, che rappresenta una generica parte certificata comunicante in una comunicazione asimmetrica e simmetrica. Questa classe è estesa da
    - Student;
    - University;
- Blockchain, la blockchain che esegue lo smart contract che permette il salvataggio di una Certificate Revocation List accessibile in maniera indipendente da tutte le parti in gioco.

Il modulo contiene anche StudentInfo, una classe di utilità che serve a raggruppare tutte le informazioni specifiche di uno studente.

CertificateAuthority

Questo modulo contiene l'omonima classe che rappresenta una astrazione semplificata di una CertificateAuthority che emette certificati di identità che scadono dopo 30 giorni.

Il modulo contiene inoltre la classe CertificateOfIdentity, che è proprio la classe che rappresenta il certificato emesso dalla CA.

Utils

Questo modulo contiene la classe che regge la logica crittografica di tutto il progetto: CryptoUtils, questa classe contiene metodi per la crittografia simmetrica e asimmetrica.

Al fine di semplificare la gestione delle informazioni nelle classi comunicanti sono state anche create le classi AsymmetricEncryptionInformation e SymmetricEncryptionInformation, che incapsulano rispettivamente le informazioni per la crittografia asimmetrica e simmetrica.

Infine il modulo Utils contiene la classe MerkleTree, fondamentale per la generazione delle credenziali accademiche certificate e per la loro verifica.

Main

Sono allegati due main, entrambi svolgono la stessa simulazione, l'unica differenza è che mentre uno è un semplice python script l'altro è un notebook auto documentante che contiene la descrizione dei vari blocchi e favorisce una comprensione "at first glance" dei vari passaggi.

Analisi dettagliata della simulazione proposta

Il progetto consiste in un sistema completo per la condivisione di credenziali accademiche realizzato in accordo a quanto specificato nei primi 3 WP.

Al fine di dimostrare la genuinità del codice prodotto questo viene presentato con un main che esegue una simulazione di un flusso di lavoro completo, il main è fornito sia come script che come notebook auto documentante. La documentazione inclusa nel notebook è anche riproposta di seguito per completezza.

Set-up degli attori del sistema

Per prima cosa il main provvede ad istanziare:

- uno studente;

- un'università che farà da università casa dello studente;

- un'altra università che farà da università ospitante;

- una istanza della classe CertificateAuthority che "emula" il comportamento di una CA;

- una istanza della classe Blockchain che "emula" il comportamento di uno smart contract per la memorizzazione della Certificate Revocation List sulla blockchain.

Inoltre vengono aggiunte le informazioni dello studente alla lista delle informazioni sugli studenti dell'università casa, in modo che questa possa, quando sarà richiesto, generare il certificato con le informazioni da lei autenticate.

Descrizione generale della simulazione

La simulazione è volta a mostrare il funzionamento di tutte le fasi descritte nel Work package 3.

- Fase A1, uno studente inizia una comunicazione con la sua università d'origine;
- Fase B, lo studente richiede alla sua università di origine le proprie informazioni certificate;
- Fase A2, lo studente inizia una comunicazione con la sua università ospitante;
- Fasi C/D (1), l'università ospitante chiede delle informazioni specifiche allo studente, che le invia e riceve un ack;
- Fase E, l'università d'origine revoca il certificato dello studente;
- Fasi C/D (2), l'università ospitante chiede delle informazioni specifiche allo studente, che le invia e riceve un nack.

Questo flusso di lavoro permette di dimostrare il funzionamento delle principali funzionalità del progetto.

Nota: Dopo le fasi C/D (1) la connessione simmetrica tra studente e università ospitante è stata lasciata aperta per evitare di dover ripetere una terza volta la fase A.

FASE A1: INIZIO COMUNICAZIONE STUDENTE - UNIVERSITA'

Lo studente e l'università di origine richiedono alla certificate authority i propri certificati, dopodiché se li scambiano e procedono all'esecuzione del protocollo di distribuzione sicura della chiave di sessione.

FASE B: RICHIESTA CERTIFICATO ALL'UNIVERSITA'

Lo studente richiede all'università d'origine il proprio certificato accademico autenticato.

FASE A2: INIZIO COMUNICAZIONE STUDENTE - UNIVERSITA' OSPITANTE

Al fine di simulare lo scambio tra studente e università ospitante è necessario ripetere la fase A, questa volta tra questi due enti.

FASE C: INVIO CERTIFICATO ALL'UNIVERSITA' e FASE D VERIFICA CERTIFICATO

L'università ospitante richiede una certa informazione allo studente, quest'ultimo la fornisce insieme alla merkle proof necessaria a verificare l'autenticità dei dati. Infine l'università ospitante fornisce un ACK per confermare che i dati erano corretti. La comunicazione simmetrica tra le due parti non viene chiusa in quanto si intende riprovare questa fase dopo aver revocato il certificato e si vuole evitare una terza esecuzione della fase per brevità.

FASE E: REVOCA CERTIFICATO

L'università di origine esegue la revoca del certificato rilasciato allo studente aggiungendo il codice del certificato alla Certificate Revocation List ospitata dalla blockchain.

FASE C: INVIO CERTIFICATO ALL'UNIVERSITA' e FASE D: VERIFICA CERTIFICATO

L'università ospitante richiede una certa informazione allo studente, quest'ultimo la fornisce insieme alla merkle proof necessaria a verificare l'autenticità dei dati. Infine l'università ospitante fornisce un NACK in quanto il certificato risulta revocato.

Output della simulazione

Di seguito si presenta l'output completo della simulazione, una visualizzazione più agevole e partizionata per le varie fasi è reperibile nel notebook main_documentato.ipynb.

== FASE A == INIZIO COMUNICAZIONE STUDENTE - UNIVERSITA' DI ORIGINE ==

=== MESSAGGIO 1 ===

Mittente : Studente

Destinatario : Università di origine

Descrizione : Certificato firmato dalla CA

Contenuto : C_Stu = E(PR_{auth}, [T_1 || ID_Stu || PU_Stu])

NON è passato più di un mese, certificato accettato.

=== MESSAGGIO 2 ===

Mittente : Università di origine

Destinatario : Studente

Descrizione : Certificato firmato dalla CA

Contenuto : C_U = E(PR_{auth}, [T_2 || ID_U || PU_U])

NON è passato più di un mese, certificato accettato.

=== MESSAGGIO 3 ===

Mittente : Università di origine

Destinatario : Studente

Descrizione : Inizio della challenge - mutual authentication protocol

Contenuto : E(PU_Stu, [ID_U || Nonce_1])

=== MESSAGGIO 4 ===

Mittente : Studente

Destinatario : Università di origine

Descrizione : Risposta alla challenge e invio sfida di autenticazione all'università

Contenuto : E(PU_U, [Nonce_1 || Nonce_2])

=== MESSAGGIO 5 ===

Mittente : Università di origine

Destinatario : Studente

Descrizione : Conclusione autenticazione reciproca

Contenuto : E(PU_Stu, Nonce_2)

=== MESSAGGIO 6 ===

Mittente : Università di origine

Destinatario : Studente

Descrizione : Distribuzione chiave simmetrica

Contenuto : E(PU_Stu, E(PR_U, K_S))

== FASE B == RICHIESTA CERTIFICATO ALL'UNIVERSITA' ==

=== MESSAGGIO 1 ===

Mittente : Studente

Destinatario : Università di origine

Descrizione : Richiesta credenziali

=== MESSAGGIO 2 ===

Mittente : Università di origine

Destinatario : Studente

Descrizione : Invio credenziali con Merkle Tree per verificarne l'autenticità

Contenuto : E(K_S, MerkleTree||E(K_U, RadiceMerkleTree))

== FASE A == INIZIO COMUNICAZIONE STUDENTE - UNIVERSITA' OSPITANTE ==

=== MESSAGGIO 1 ===

Mittente : Studente

Destinatario : Università ospitante

Descrizione : Certificato firmato dalla CA

Contenuto : C_Stu = E(PR_{auth}, [T_1 || ID_Stu || PU_Stu])

NON è passato più di un mese, certificato accettato.

=== MESSAGGIO 2 ===

Mittente : Università ospitante

Destinatario : Studente

Descrizione : Certificato firmato dalla CA

Contenuto : C_U = E(PR_{auth}, [T_2 || ID_U || PU_U])

NON è passato più di un mese, certificato accettato.

=== MESSAGGIO 3 ===

Mittente : Università ospitante

Destinatario : Studente

Descrizione : Inizio della challenge - mutual authentication protocol

Contenuto : E(PU_Stu, [ID_U || Nonce_1])

=== MESSAGGIO 4 ===

Mittente : Studente

Destinatario : Università ospitante

Descrizione : Risposta alla challenge e invio sfida di autenticazione all'università

Contenuto : E(PU_U, [Nonce_1 || Nonce_2])

=== MESSAGGIO 5 ===

Mittente : Università ospitante

Destinatario : Studente

Descrizione : Conclusione autenticazione reciproca

Contenuto : E(PU_Stu, Nonce_2)

=== MESSAGGIO 6 ===

Mittente : Università ospitante

Destinatario : Studente

Descrizione : Distribuzione chiave simmetrica

Contenuto : E(PU_Stu, E(PR_U, K_S))

== FASE C == INVIO CERTIFICATO ALL'UNIVERSITA' ==

=== MESSAGGIO 1 ===

Mittente : Università ospitante

Destinatario : Studente

Descrizione : Richiesta informazioni specifiche

Contenuto : E(K_S, Richiesta mail_casa)

=== MESSAGGIO 2 ===

Mittente : Studente

Destinatario : Università ospitante

Descrizione : Informazioni specifiche richiesta con Markle Tree per verificarte l'auteticità

Contenuto : E(K_S, foglieRichiesteDelMerkleTree||nodiAggiuntiviDelMerkleTreePerIlCalcoloDellaRadice||E(K_U, RadiceMerkleTree))

== FASE D == VERIFICA CERTIFICATO ==

=== MESSAGGIO 1 ===

Mittente : Università ospitante

Destinatario : Studente

Descrizione : Notifica di ricezione di certificato corretto o no

Contenuto : E(K_S, ack/nack)

ACK

Il certificato delle informazioni sullo studente è stato

ACCETTATO

== FASE E == REVOCA CERTIFICATO ==

=== MESSAGGIO 1 ===

Mittente : Università ospitante

Destinatario : Studente

Descrizione : Aggiunta alla CRL sulla blockchain del codice del certificato da revocare

certificato di 01 revocato

== FASE C == INVIO CERTIFICATO ALL'UNIVERSITA' ==

=== MESSAGGIO 1 ===

Mittente : Università ospitante

Destinatario : Studente

Descrizione : Richiesta informazioni specifiche

Contenuto : E(K_S, Richiesta mail_casa)

=== MESSAGGIO 2 ===

Mittente : Studente

Destinatario : Università ospitante

Descrizione : Informazioni specifiche richiesta con Markle Tree per verificarte l'auteticità

Contenuto : E(K_S, foglieRichiesteDelMerkleTree||nodiAggiuntiviDelMerkleTreePerIlCalcoloDellaRadice||E(K_U, RadiceMerkleTree))

== FASE D == VERIFICA CERTIFICATO ==

=== MESSAGGIO 1 ===

Mittente : Università ospitante

Destinatario : Studente

Descrizione : Notifica di ricezione di certificato corretto o no

Contenuto : E(K_S, ack/nack)

NACK

Il certificato delle informazioni sullo studente è stato

RIFIUTATO

Prestazioni

Dimensione delle credenziali

In questa sezione si esplorano le prestazioni a livello di complessità spaziale dei meccanismi di certificazione e crittografia asimmetrica/simmetrica implementati.

Al fine di visualizzare i dati alla base dei successivi paragrafi è necessario impostare a True la costante VERBOSE_MESSAGE_SIZE nei documenti:

- University.py;
- CertificateAuthority.py;
- CryptoUtils.py.

Nota: tutti i dati riportati sono relativi ai messaggi e ai certificati creati durante la simulazione, i valori variano leggermente ad ogni esecuzione e potrebbero cambiare molto se venissero cambiati i messaggi coinvolti, ad esempio cambiando significativamente la lunghezza.

Dimensione del certificato di identità

La CA rilascia certificati complessivamente lunghi 1369 caratteri, di questi 256 caratteri sono occupati dalla firma e i restanti 1113 caratteri sono le informazioni certificate. Questo vuol dire che i certificati rilasciati dalla CA hanno il 18.7% di overhead a livello di complessità spaziale.

Dimensione del certificato studentesco

Le università rilasciano certificati che con dati di lunghezza ad esempio 329 caratteri arrivano a 1711 caratteri post certificazione.

Questo implica un overhead del 520% per garantire autenticazione decentralizzata dei dati.

Overhead della crittografia asimmetrica

Il processo di crittografia asimmetrica per garantire confidenzialità e integrità che è stato implementato produce, per un messaggio ad esempio di lunghezza 159 caratteri, un ciphertext di lunghezza 723 caratteri.

Questo implica un overhead del 454% per garantire confidenzialità e integrità con la crittografia asimmetrica.

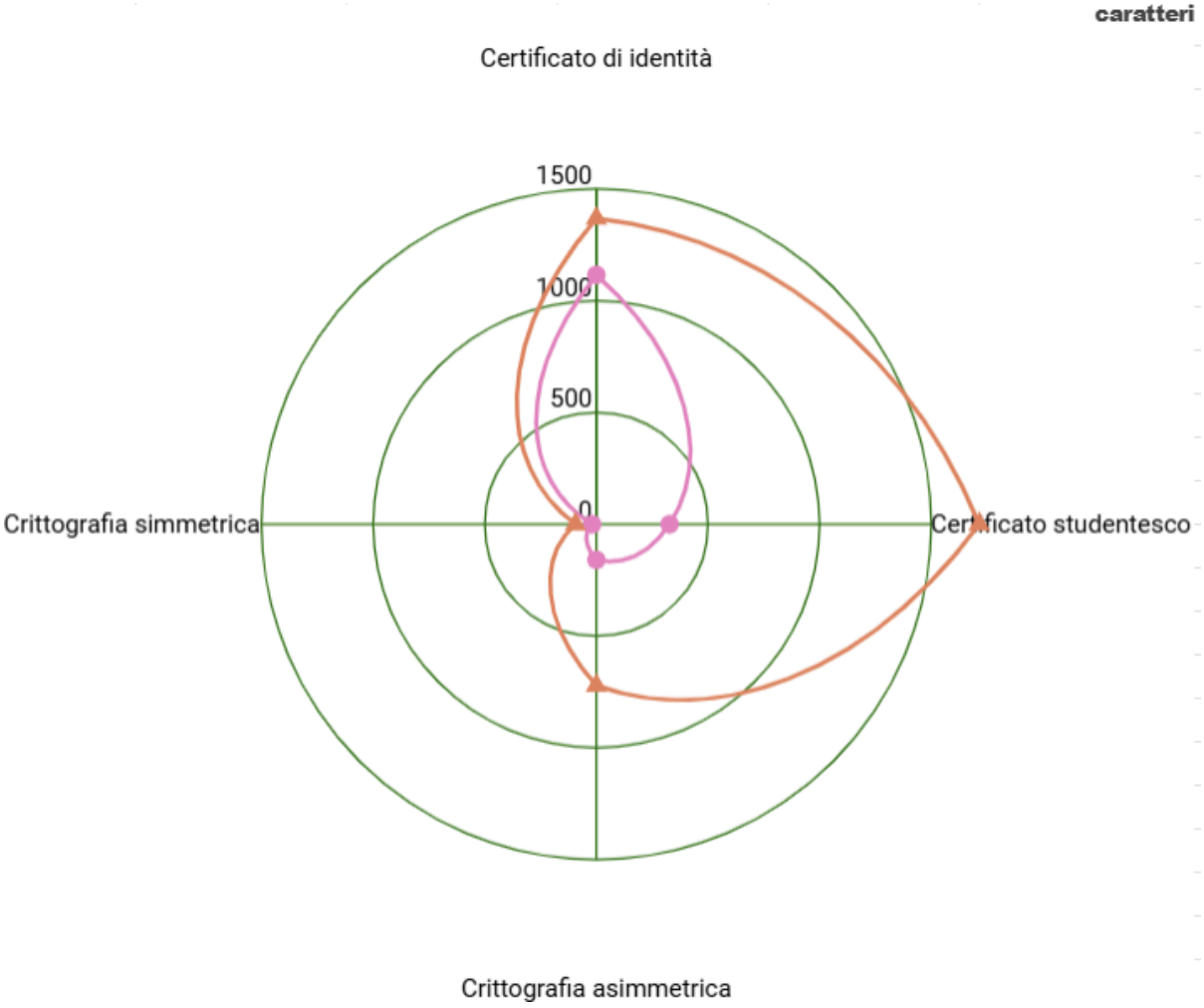
Overhead della crittografia simmetrica

Il processo di crittografia simmetrica per garantire confidenzialità e integrità che è stato implementato produce, per un messaggio ad esempio di lunghezza 20 caratteri, un ciphertext di lunghezza 96 caratteri.

Tuttavia un messaggio di lunghezza 1711 caratteri diventa un ciphertext di 1904 caratteri.

Nel primo caso abbiamo un overhead del 480% ma con un messaggio significativamente più lungo l'overhead scende a 111%.

Questo è in linea con il risultato atteso che l'overhead della crittografia simmetrica si ammortizza più rapidamente rispetto a quello della crittografia asimmetrica.



Presentazione tra le parti

Per la presentazione tra le parti comunicanti è stato usato il protocollo di distribuzione sicura della chiave di sessione, questo implica una latenza complessiva di circa 1,5 secondi, ma combinato alla fiducia nei certificati emanati dalla Certificate Authority garantisce una comunicazione confidenziale e autenticata.

Latenza di verifica

In questa sezione si esplorano le prestazioni a livello della latenza di verifica dei meccanismi di certificazione e crittografia asimmetrica/simmetrica implementati.

Al fine di visualizzare i dati alla base dei successivi paragrafi è necessario impostare a True la costante VERBOSE_MESSAGE_TIME nei documenti:

- University.py;
- CertificateAuthority.py;
- CryptoUtils.py.

Nota: tutti i dati riportati sono relativi ai messaggi e ai certificati creati durante la simulazione, i valori variano leggermente ad ogni esecuzione e potrebbero cambiare molto se venissero cambiati i messaggi coinvolti, ad esempio cambiando significativamente la lunghezza.

Latenza della creazione di un certificato di identità

La creazione di un certificato di identità richiede, secondo i dati registrati, al massimo 100 millisecondi.

Latenza della creazione di un certificato accademico da parte dell'università

La creazione di un certificato accademico richiede, secondo i dati registrati, al massimo 150 millisecondi.

Latenza della verifica di un certificato accademico da parte dell'università

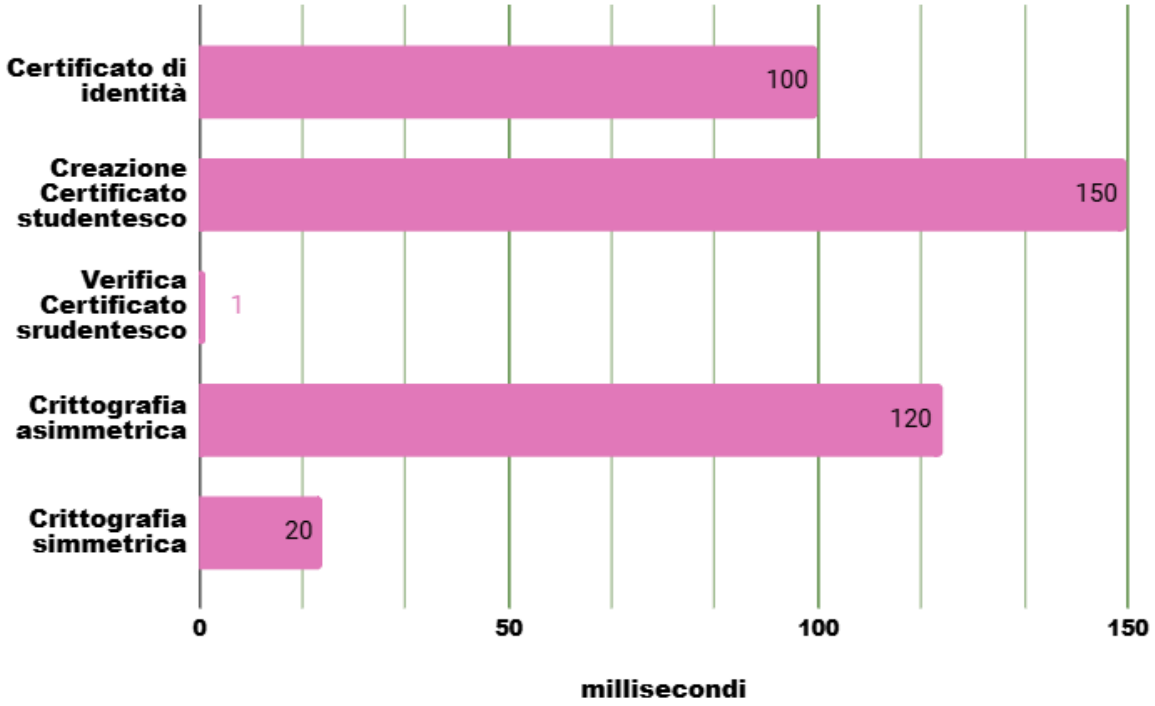
La verifica di un certificato accademico richiede, secondo i dati registrati, circa 1 millisecondo.

Latenza della crittografia simmetrica

Il processo di crittografia simmetrica per garantire confidenzialità e integrità che è stato implementato richiede una latenza di circa 20 millisecondi per l'encryption e altrettanti per la decryption.

Latenza della crittografia asimmetrica

Il processo di crittografia asimmetrica per garantire confidenzialità e integrità che è stato implementato richiede una latenza di circa 120 millisecondi per l'encryption e altrettanti per la decryption.


Certificato di identità
100
Creazione Certificato studentesco
150
Verifica Certificato srudentesco
1
Crittografia asimmetrica
120
Crittografia simmetrica
20
0
50
100
150
millisecondi