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 ... **3**
Honest Actors and their Goals ... **3**
Student ... 3
Host University ... 3
University of Origin ... 3
Functionality that is intended to be implemented ... **3**
Adversaries (threat model)+ Objectives/attacks of adversaries ... **3**
Threat Model 1: Adulteration of a student's credentials ... 3
Threat Model 2: Theft of a student's credentials by posing as such ... 4
Threat Model 3: Theft of a student's credentials by posing as a university ... 4
Threat Model 4: Man in the middle during key exchange ... 4
Threat Model 5: Sending inconsistent but valid information ... 5
Threat Model 6: Changing the revocation status of an unrevoked certificate ... 5
Threat Model 7: Changing the revocation status of a revoked certificate ... 5
Properties that you would like to be able to preserve ... **5**
Defining the structure (fields) of the credential ... **6**
WP2 ... **10**
Credential management system+ Actions of honest parties ... **10**
WP3 ... **14**
Security analysis ... **14**
Threat Model 1: Adulteration of a student's credentials ... 14
Threat Model 2: Theft of a student's credentials by posing as such ... 14
Threat Model 3: Theft of a student's credentials by posing as ... 14
Threat Model 4: Man in the middle during key exchange ... 15
Threat Model 5: Sending inconsistent but valid information ... 15
Threat Model 6: Changing the revocation status of an unrevoked certificate ... 16
Threat Model 7: Change of revocation status of a revoked certificate ... 16
WP4 ... **17**
Implementation ... **17**
Actors ... 17
CertificateAuthority ... 17
Utils ... 17
Main ... 18
Detailed analysis of the proposed simulation ... **18**
Set-up of system actors ... 18
General description of the simulation ... 18
PHASE A1: BEGINNING STUDENT-UNIVERSITY COMMUNICATION ... 19
STEP B: CERTIFICATE REQUEST TO THE UNIVERSITY ... 19
PHASE A2: BEGINNING STUDENT-HOST UNIVERSITY COMMUNICATION ... 19
STEP C: SEND CERTIFICATE TO UNIVERSITY and STEP D CERTIFICATE ... 19
PHASE E: REVOCATION OF CERTIFICATE ... 19
STEP C: SEND CERTIFICATE TO UNIVERSITY and STEP D: CERTIFICATE ... 20
Simulation output ... **20**
Performance ... **25**
Size of credentials ... 25

Size of identity certificate 26
Size of student certificate 26
Asymmetric Cryptography Overhead 26
Overhead of symmetric cryptography 26
Presentation between the parties 27
Verification latency 27
Latency of identity certificate creation 28
Latency of the creation of an academic certificate by 28
Latency of verification of an academic certificate by the university 28
Latency of symmetric encryption 28
Latency of asymmetric cryptography 28

WP1

Honest Actors and their Goals

The following are the 3 honest actors identified during WP1 and their objectives pursued while using the produced software.

Student

- Receive certified credentials from his or her University of Origin;
- Receive certified credentials from his or her Host University;
- Selectively send certified credentials to his or her University of Origin;
- Selectively send certified credentials to his or her Host University.

Host University

- Create certified credentials;
- Verify validity of certified credentials;
- Revoke issued certificates.

Originating University

- Create certified credentials;
- Verify validity of certified credentials;
- Revoke issued certificates.

Functionality intended to be implemented

- Sending credentials;
- Requesting credentials;
- Checking the identity of the interlocutor;
- Check of requested credentials;
- Check Revocation;
- Secure session key distribution protocol to verify authenticity.

Adversaries (threat model)+ Adversary targets/attacks.

Threat Model 1: Adulteration of a student's credentials.

Goal: To alter a student's credentials in order to provide him or her, for a fee, with an increase in GPA.

Impact: inability to recognize any adulteration in the message content.

Attacker: malicious student.

Assets: access to one's private key and all one's certificates, related to credentials and identity.

Asset to be protected: integrity of credentials.

Threat Model 2: Theft of a student's credentials by posing as a student

Goal: To pretend to be a student in order to obtain his or her credentials by requesting them from the university and then enact identity theft.

Impact: unlawful disclosure of student's private information to a third party.

Attacker: acquaintance of the student.

Asset: access to the student's Certificate of Identity obtained t h r o u g h prior communication with the student.

Asset to be protected: privacy of the student.

Threat Model 3: Theft of credentials of one student Pretending to be a university

Goal: To pretend to be a university in order to receive a student's credentials and enact the sale of the student's credentials.

Impact: illicit disclosure of students' private information to third parties.

Attacker: professional hacker.

Asset: access to the University"s Certificate of Identity obtained t h r o u g h prior communication with the same.

Asset to be protected: student's privacy.

Threat Model 4: Man in the middle during key exchange

Goal: obtain the session key

Impact: loss of security in information exchange by both honest parties.

Attacker: professional hacker.

Asset: active access to communication.

Asset to protect: session key for symmetric communication.

Threat Model 5: Sending inconsistent but valid information.

Goal: Send valid and certified Credentials but exploit selective disclosure in order to send inconsistent information to take advantage of it, such as sharing the name of an exam only attended and the grade of an exam taken.

Impact: risk of validating an exam not taken.

Attacker: student who wants to "steal" an exam.

Assets: access to one's private key and all one's certificates, related to credentials and identity.

Asset to protect: validity of the information exchanged.

Threat Model 6: Changing the revocation status of a certificate not revoked

Goal: Change the revocation status of a certificate to make a valid certificate appear revoked.

Impact: loss of trust in the university and the certificates it issues.

Attacker: malicious third party.

Assets: being a proponent of the blockchain that hosts the revocation list.

Assets to protect: university credibility and certificate credibility.

Threat Model 7: Changing the revocation status of a revoked certificate.

Goal: Change the revocation status of a certificate to make a revoked certificate appear unrevoked.

Impact: loss of trust in the university and the certificates it issues.

Attacker: malicious third party.

Resources: being a proponent of the blockchain that hosts the revocation list.

Assets to be protected: credibility of the university and credibility of certifications.

Properties that one would like to be able to preserve

- confidentiality;
- integrity;
- non-repudiation;
- resilience (robustness);
- authentication.

Definition of the structure (fields) of the credential


```
{
    "matricola_casa": "0123456789",
    "freshman_host": "0123456789", "first_name":
    "Mario",
    "last_name": "Rossi",
    "home_email": "mario.rossi@studenti.casa.it ",
    "guest_email": "mario.rossi@etudiant.maison.fr ",
    "date_of_birth": "01/01/2000"

    "codice_corso_di_laurea": "LM-32",
    "degree_course_name": "Master's degree in Computer Engineering",
    "cfu_total_achieved": 100,
    "grade_average": 28,

    "attendance_certificates": [
      {
        "subject_id": "attendance_certifications_65468481",
        "course_name": {
          "object_id": "attendance_certifications_65468481", "course_name":
          "Mathematical Analysis 1"
        },
        "percentage_of_attendance": {
          "object_id": "attendance_certifications_65468481",
          "percentage_attended": 87
        }
      },
      {
        "id_object": "attendance_certifications_65468259",
        "course_name": {
          "object_id": "attendance_certifications_65468259", "course_name":
          "Physics 1"
        },
        "percentage_of_attendance": {
          "subject_id": "attendance_certifications_65468259",
          "percentage_attended": 78
        }
      }
    ],

    "titles_of_study": [
      {
        "id_object": "titles_of_study_121548448", "name":
        {
```

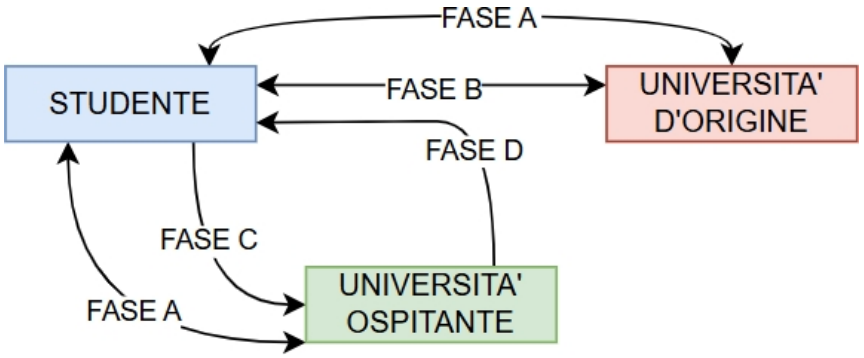
```
            "id_oggetto": "titoli_di_studio_121548448",
            "name": "Bachelor's Degree in Computer Engineering"
          },
          "grade": {
            "id_object": "titles_of_study_121548448", "grade":
            110
          },
          "praise": {
            "id_object": "titles_of_study_121548448", "praise":
            true
          }
        },
        {
          "object_id": "titles_of_study_121548158", "name":
          {
            "id_oggetto": "titoli_di_studio_121548158",
            "name": "High School Diploma of Scientific High School Addressing
Applied Sciences"
          },
          "grade": {
            "id_object": "titles_of_study_121548158", "grade":
            100
          },
          "praise": {
            "id_object": "titles_of_study_121548158", "praise":
            true
          }
        }
      ],

      "exams_accomplished": [
        {
          "id_object": "exams_accomplished_121548484",
          "exam_code": {
            "id_object": "exams_accomplished_121548484",
            "exam_code": "123"
          },
          "exam_name": {
            "subject_id": "exams_accomplished_121548484", "exam_name":
            "Algorithms and Protocols for Security"
          },
          "cfu_number": {
            "subject_id": "exams_accomplished_121548484",
            "number_cfu": 9
          },
          "achieved": {
```

```
            "id_object": "exams_accomplished_121548484",
            "achieved": false
          },
          "date_achieved": {
            "id_object": "exams_achieved_121548484", "date_achieved": null
          },
          "grade": {
            "id_object": "exams_achieved_121548484", "grade":
            null
          },
          "professor_responsible": {
            "id_object": "exams_accomplished_121548484",
            "professor_responsible": "Carlo Mazzocca"
          },
          "university_of_achievement": { "subject_id":
            "exams_achieved_121548484"
            "university_of_achievement": "University of
Salerno"
          }
        },
        {
          "id_object": "exams_accomplished_487845485",
          "exam_code": {
            "subject_id": "exams_accomplished_487845485",
            "exam_code": "124"
          },
          "exam_name": {
            "subject_id": "exams_accomplished_487845485",
            "exam_name": "Analyse des données"
          },
          "cfu_number": {
            "id_object": "exams_accomplished_487845485",
            "number_cfu": 9
          },
          "achieved": {
            "id_object": "exams_accomplished_487845485",
            "achieved": true
          },
          "date_achieved": {
            "id_object": "exams_accomplished_487845485", "date_achieved":
            "15_01_2025"
          },
          "grade": {
            "subject_id": "exams_achieved_487845485", "grade":
            30
```

```
        },
        "professor_responsible": {
          "id_object": "exams_accomplished_487845485",
          "professor_responsible": "Jean-Pierre"
        },
        "university_of_achievement": { "subject_id":
          "exams_achieved_487845485"
          "university_of_achievement": "Université de Rennes"
        }
    }
    ]
}
```

WP2

Credential management system+ Honest party actions



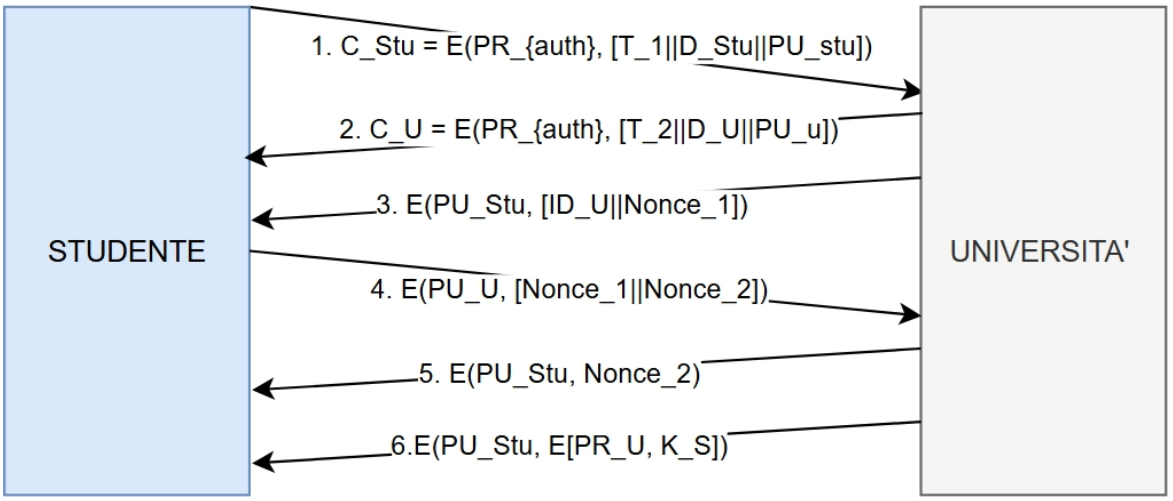
Preconditions:

- users and Universities are accredited with the CA; each of them has a certificate that binds their identity to a public key;
- each University has a database containing the credentials of students accredited with them;
- signing is done using appropriate Padding so as to create a rigid structure in order to prevent message-less attacks and absence of multiplicative relationships for the forgery attack with arbitrary messages;
- presence of a blockchain-based revocation system; every revocation event is recorded on the blockchain, and if you want to check whether a certificate has been revoked you only need to check whether its ID is on the blockchain.

Operation:

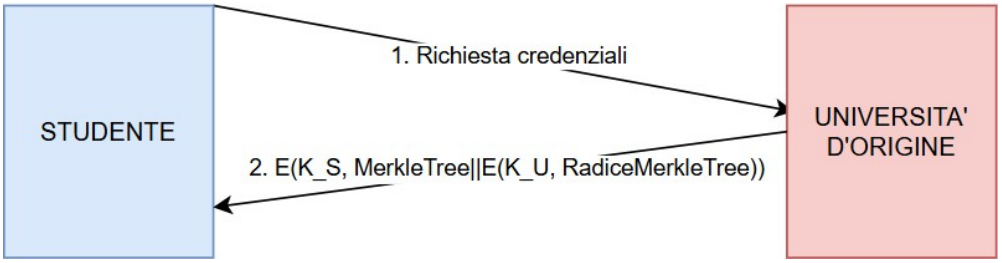
A. Initiation of communication between student and university
    - If you do not have the respective identity certificates, the student sends his or her Certificate of Identity to the university and vice versa, in order to protect the integrity of the transmission by the use of public keys.
    - In order to ensure authentication, the secure session key distribution protocol is used.
        - The university sends a random code to the student encrypted with the student's own public key, the student must decrypt it with his or her own private key and then send it back to the university encrypted with the university's public key, to which he or she also adds another encrypted Nonce that he or she expects to receive to be certain that he or she is interacting with the university.

FASE A: INIZIO COMUNICAZIONE STUDENTE - UNIVERSITA'



B. Requesting a certificate from the university
   - Communication between student and university begins
   - The student asks for his or her credentials
   - The university retrieves the student's data from its database and composes a Merkle Tree that has the student's credentials for leaves, encrypts the root of the tree with its own private key, so the integrity of the information is safeguarded.
   - The university creates the certificate by pairing the Merkle Tree and the encrypted root.
   - The university sends the certificate, to the student. For confidentiality purposes this transmission is done in encrypted form using the session key.
   - The student receives the information and decrypts it with the session key.

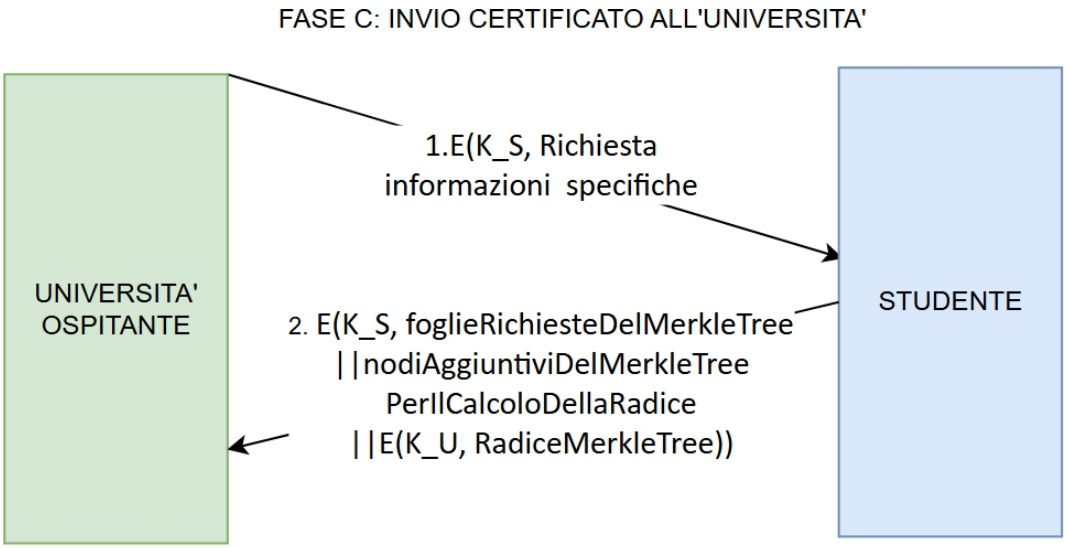
FASE B: RICHIESTA CERTIFICATO ALL'UNIVERSITA'



C. Sending certificate to university
   - Start of communication between student and university
   - The student extrapolates from the Merkle Tree the leaves related to the information requested by the university, in case this information is not sufficient for the calculation of the Merkle Tree root the student also extrapolates the hashes necessary for the completion of the calculation in question, taking care to choose them so as to minimize the number of processing required on the receiving side.
   - The student sends the certificate, with only the strictly necessary information previously extrapolated, to the requesting university. For

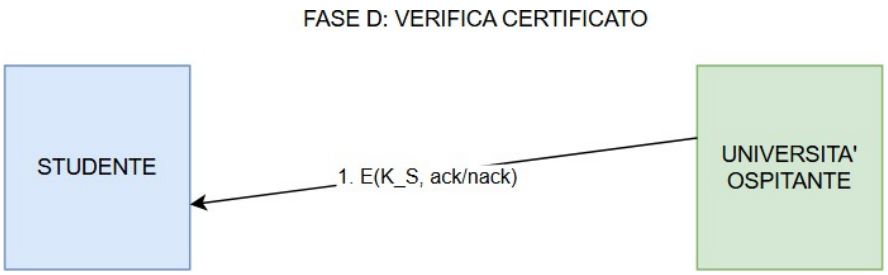
safeguard confidentiality all this information is encrypted with the session key.

- The university receives the ciphertext and decrypts it with the session key.
- Certificate Verification.



D. Certificate verification

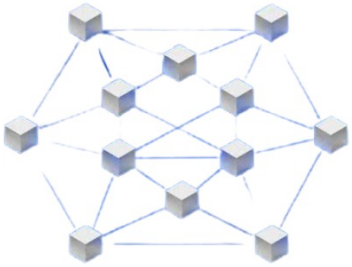
- You perform the Merkle Tree root calculation based on the information provided, after which you decrypt the certificate signature using the public key of the issuing university. At this point you are in possession of two digests and proceed to check the match between the two, if the match is verified the information is valid. (If you do not have the other university's identity certificate you request it, step a).
- If the certificate is valid the university checks to see if its code is on the blockchain, which would indicate that the certificate has been revoked.
- The student is notified whether the certificate was found to be valid or not.



E. Revocation of a certificate

- If a university wants to revoke a certificate it has issued, it can make the addition to the Certificate Revocation List on the blockchain of the identification code of the certificate to be revoked. From now on when a university receives the certificate in question when it checks in on the blockchain it will find its code and thus know that it has been revoked.

FASE E: REVOCA CERTIFICATO



Aggiunta alla CRL sulla blockchain del codice del certificato da revocare

UNIVERSITA' D'ORIGINE

WP3

Security Analysis.

Threat Model 1: Adulteration of a student's credentials.

The developed system turns out to be able to prevent the adulteration of a student's credentials, for example in order to raise his or her GPA, by the malicious student who obviously has possession of his or her own private key and all his or her certificates.

This is true in that the use of the private key allows the attacker to establish the connection with the university properly but still the modification of credentials leads to an alteration of the corresponding Merkle Tree root. Thus, the root of the Merkle Tree will not be the same as the one obtained from decrypting the certificate signature using the university's public key; this highlights the tampering of the transmitted credentials, safeguarding integrity.

The only way this attack could work is if the attacker could find different credentials but still be able to produce the same hash of existing certificates. Even remembering that the attacker already has all of his own certificates, being able to find new credentials with the same hash is, realistically speaking, impossible given the noncollision and noninvertibility characteristics of the hash.

Threat Model 2: Theft of a student's credentials by posing as a student

The developed system turns out to be able to prevent an attacker who is in possession of the student's Certificate of Identity from impersonating the student in a communication with the university in order to obtain the confidential credentials in order to disclose them to third parties.

This is true in that simply exchanging the identity certificates is not enough to initiate the communication, there is in fact then the initiation of a secure session key exchange protocol that requires both parties to prove that they really are who they say they are by decrypting an encrypted Nonce with the public key they provide.

The scheme would be breached only if the attacker were in possession of the student's private key, but this is beyond the resources available to him in this Threat Model.

The above makes it possible to say that the scheme is capable of safeguarding the student's privacy.

Threat Model 3: Theft of credentials of one student impersonating a university

The developed system turns out to be able to prevent a hacker from impersonating a university in order to have a student's credentials sent to him or her, being in possession of the University's Certificate of Identity due to past communications.

This is true in that even being in possession of the University's Certificate of Identity, the hacker will not be able to terminate the communication setup, as this requires the secure key distribution protocol to be executed after the certificate exchange.

This protocol prevents this type of attack because it requires the parties involved to prove that they are who they say they are by decrypting a Nonce encrypted with the public key they provide.

To overcome the secure key distribution protocol the attacker would have to be in possession of the university's private key, which is beyond the resources specified in the Threat Model.

Thus, the system turns out to protect the student's privacy.

Threat Model 4: Man in the middle during key exchange.

The developed system turns out to be able to prevent a hacker who has active access to the communication channel from obtaining the session key by means of a Man in the Middle attack.

In fact, the system requires the parties involved to prove their identity by decrypting Nonce encrypted with their public key. This process, which is part of the secure key distribution protocol, prevents a Man in the Middle attack.

An attacker intending to carry out this type of attack would have to have the private keys of the communicating parties in order to get through the set-up phase of the communication, this is beyond the resources of this Threat Model.

The system then protects the session key for symmetric communication and all that goes with it.

Threat Model 5: Sending inconsistent but valid information.

The developed system turns out to be able to prevent the sending of inconsistent but valid information by an attacker who wants to show better credentials than he actually possesses.

This is true in that if one sends the name of one exam and the grade of another, the receiver will confirm that the certificate is valid but will be able to see that the information is inconsistent because the composite information is marked with an ID that binds them, and the name of the exam and the grade of the example will have different IDs.

The attack could only work if it were possible to change the ID of one of the two objects to make it the same as the other, however, this would alter the root of the Merkle Tree invalidating the certificate.

The system then safeguards the validity and consistency of the exchanged information.

Threat Model 6: Change of revocation status of a certificate not revoked

The developed system turns out to be able to prevent changes in revocation status in order to make a certificate revoked if the attacker is a malicious third party who has had a way to get hold of revoked Certificates and not the specific student.

This is true in that even being a proposer, if an incorrect block is proposed to the blockchain the attesters will vote against adding it to the blockchain.

The system thus turns out to protect the consistency of the revocation.

Threat Model 7: Changing the revocation state of a revoked certificate.

The developed system turns out to be able to prevent changes in revocation status in order to make a revoked certificate non-revoked if the attacker is a malicious third party who has had a way to get hold of revoked Certificates and not the specific student.

This is true in that even being a blockchain proponent, the fact remains that it is an append-only data structure and thus changes to already added data are not possible.

Therefore, the system turns out to be able to protect the consistency of the revocation.

WP4

Implementation

Software development was done in accordance with OOP principles so as to favor a product that was readable (so as to facilitate correction), maintainable, and with potentially reusable parts (especially the CryptoUtils class).

It should be noted that for simplicity, a version of the academic credentials has been implemented that includes only the basic information.

Below is the json of the academic credentials actually modeled in code:

```
{
    "matricola_casa": "0123456789",
    "freshman_host": "0123456789", "first_name":
    "Mario",
    "last_name": "Rossi",
    "home_email": "mario.rossi@studenti.casa.it ",
    "guest_email": "mario.rossi@etudiant.maison.fr ",
    "date_of_birth": "01/01/2000"

    "codice_corso_di_laurea": "LM-32",
    "degree_course_name": "Master's degree in Computer Engineering",
    "cfu_total_achieved": 100,
    "grade_average": 28,
}
```

A precise description of the various modules follows.

Actors

The actors module contains the classes that represent the actors of the program.

These are:

- CertifiedCommunicatingParty, which represents a generic certified communicating party in an asymmetric and symmetric communication. This class is extended by
    - Student;
    - University;
- Blockchain, the blockchain that executes the smart contract that enables the storage of a Certificate Revocation List that can be accessed independently by all parties involved.

The module also contains StudentInfo, a utility class used to group all student-specific information together.

CertificateAuthority

This module contains the class of the same name that represents a simplified abstraction of a CertificateAuthority that issues identity certificates that expire after 30 days.

The module also contains the CertificateOfIdentity class, which is precisely the class that represents the certificate issued by the CA.

Utils

This module contains the class that holds the cryptographic logic for the entire project: CryptoUtils, this class contains methods for symmetric and asymmetric encryption.

In order to simplify information management in the communicating classes, the AsymmetricEncryptionInformation and SymmetricEncryptionInformation classes were also created, which encapsulate information for asymmetric and symmetric encryption, respectively.

Finally, the Utils module contains the MerkleTree class, which is essential for generating certified academic credentials and verifying them.

Main

Two mains are attached, both of which perform the same simulation, the only difference being that while one is a simple python script the other is a self-documenting notebook that contains descriptions of the various blocks and facilitates an "at first glance" understanding of the various steps.

Detailed analysis of the proposed simulation

The project consists of a complete academic credential sharing system implemented according to what was specified in the first 3 WPs.

In order to demonstrate the genuineness of the produced code this is presented with a main running a simulation of a complete workflow, the main is provided both as a script and as a self-documenting notebook. The documentation included in the notebook is also reproduced below for completeness.

Set-up of system actors

First, the main instantiates:

- a student;

- a university that will act as the student's home university;

- another university that will act as the host university;

- An instance of the CertificateAuthority class that "emulates" the behavior of a CA;

- an instance of the Blockchain class that "emulates" the behavior of a smart contract for storing the Certificate Revocation List on the blockchain.

In addition, the student's information is added to the home university's student information list so that it can, when requested, generate the certificate with the information authenticated by it.

General description of the simulation

The simulation is intended to show the operation of all the steps described in Work package 3.

- Step A1, a student initiates a communication with his home university;
- Step B, the student requests their certified information from their home university;
- Step A2, the student initiates communication with his or her host university;
- Stages C/D (1), the host university requests specific information from the student, who sends it and receives an ack;
- Step E, the home university revokes the student's certificate;
- Step C/D (2), the host university requests specific information from the student, who sends it and receives a nack.

This workflow makes it possible to demonstrate the operation of the project's main features.

Note: After steps C/D (1) the symmetrical connection between student and host university was left open to avoid having to repeat step A a third time.

STEP A1: BEGINNING STUDENT-UNIVERSITY COMMUNICATION.

The student and the home university request their certificates from the certificate authority, after which they exchange them and proceed to execute the secure session key distribution protocol.

STEP B: CERTIFICATE REQUEST TO THE UNIVERSITY

The student requests his or her authenticated academic certificate from the originating university.

STEP A2: START STUDENT - UNIVERSITY COMMUNICATION HOST UNIVERSITY

In order to simulate the exchange between student and host university, it is necessary to repeat Step A, this time between these two institutions.

STEP C: SEND CERTIFICATE TO UNIVERSITY and STEP D VERIFICATION CERTIFICATE

The host university requests certain information from the student, the student provides it along with the merkle proof needed to verify the authenticity of the data. Finally, the host university provides an ACK to confirm that the data were correct. The symmetric communication between the two parties is not closed because it is intended to retry this step after revoking the certificate and to avoid a third execution of the step for brevity.

PHASE E: CERTIFICATE REVOCATION

The originating university performs the revocation of the certificate issued to the student by adding the certificate code to the Certificate Revocation List hosted by the blockchain.

STEP C: SEND CERTIFICATE TO UNIVERSITY and STEP D: VERIFICATION CERTIFICATE

The host university requests certain information from the student, the student provides it along with the merkle proof needed to verify the authenticity of the data. Finally, the host university provides a NACK as the certificate is revoked.

Simulation output

The complete output of the simulation is presented below; an easier, partitioned visualization for the various phases can be found in the main_documented.ipynb notebook.

== PHASE A== START STUDENT-UNIVERSITY-OF-ORIGIN COMMUNICATION. ==

=== MESSAGE 1 ===

Sender : Student Recipient :

University of origin

Description : Certificate signed by the CA

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

NO more than one month has passed, certificate accepted.

=== MESSAGE 2 ===

Sender : University of origin

Recipient : Student

Description : Certificate signed by the CA

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

NO more than one month has passed, certificate accepted.

=== MESSAGE 3 ===

Sender : University of origin

Recipient : Student

Description : Start of challenge - mutual authentication protocol Content.

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

=== MESSAGE 4 ===

Sender : Student Recipient :

Home university

Description : Response to challenge and sending authentication challenge to

university Content : E(PU_U, [Nonce_1 || Nonce_2])

=== MESSAGE 5 ===

Sender : University of origin

Recipient : Student

Description : Mutual authentication conclusion Content

: E(PU_Stu, Nonce_2)

=== MESSAGE 6 ===

Sender : University of origin

Recipient : Student

Description : Symmetric key distribution Content.

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

== STEP B== CERTIFICATE REQUEST TO UNIVERSITY ==

=== MESSAGE 1 ===

Sender : Student Recipient :

Home University

Description : Request credentials

=== MESSAGE 2 ===

Sender : University of origin

Recipient : Student

Description : Sending credentials with Merkle Tree to verify their authenticity

Contents. : E(K_S, MerkleTree||E(K_U, RootMerkleTree))

== PHASE A== START COMMUNICATION STUDENT - HOST UNIVERSITY ==

=== MESSAGE 1 ===

Sender : Student Recipient :

Host University

Description : Certificate signed by the CA

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

NO more than a month has passed, certificate accepted.

=== MESSAGE 2 ===

Sender : Host university

Recipient : Student

Description : Certificate signed by the CA

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

NO more than one month has passed, certificate accepted.

=== MESSAGE 3 ===

Sender : Host University

Recipient : Student

Description : Start of challenge - mutual authentication protocol Content.

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

=== MESSAGE 4 ===

Sender : Student Recipient :

Host University

Description : Response to challenge and sending authentication challenge to

university Content. : E(PU_U, [Nonce_1 || Nonce_2])

=== MESSAGE 5 ===

Sender : Host University

Recipient : Student

Description : Mutual authentication conclusion Content

: E(PU_Stu, Nonce_2)

=== MESSAGE 6 ===

Sender : Host University

Recipient : Student

Description : Symmetric key distribution Content.

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

== STEP C== SENDING CERTIFICATE TO UNIVERSITY ==

=== MESSAGE 1 ===

Sender : Host university

Recipient : Student

Description : Specific information request Content

: E(K_S, Request mail_home)

=== MESSAGE 2 ===

Sender : Student Recipient :

Host University

Description : Specific information requested with Markle Tree to verify autheticity

Content : E(K_S, LeavesRequiredOfMerkleTree|||AdditionalNodesOfMerkleTreeToCalculateTheRadix||E(K_U, RootMerkleTree))

== STAGE D== CERTIFICATE VERIFICATION ==

=== MESSAGE 1 ===

Sender : Host University

Recipient : Student

Description : Notification of receipt of correct certificate or not

Content : E(K_S, ack/nack)

ACK

The certificate of student information has been ACCEPTED

== STEP E== CERTIFICATE REVOCATION ==

=== MESSAGE 1 ===

Sender : Host University

Recipient : Student

Description : Adding to the CRL on the blockchain the code of the certificate to be revoked

certificate of 01 revoked

== STEP C== SEND CERTIFICATE TO UNIVERSITY ==

=== MESSAGE 1 ===

Sender : Host university

Recipient : Student

Description : Specific information request Content

: E(K_S, Request mail_home)

=== MESSAGE 2 ===

Sender : Student Recipient :

Host University

Description : Specific information request with Markle Tree to verify autheticity

Content : E(K_S, LeavesRequiredOfMerkleTree|||AdditionalNodesOfMerkleTreeToCalculateTheRadix||E(K_U, RootMerkleTree))

== STAGE D== CERTIFICATE VERIFICATION ==

=== MESSAGE 1 ===

Sender : Host University

Recipient : Student

Description : Notification of receipt of certificate correct or not

Content : E(K_S, ack/nack)

NACK

The certificate of student information has been REJECTED

Performance

Size of credentials

This section explores the performance at the level of spatial complexity of the implemented asymmetric/symmetric certification and encryption mechanisms.

In order to display the data underlying the following sections, it is necessary to set the constant VERBOSE_MESSAGE_SIZE in the documents to True:

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

Note: All data shown are for the messages and certificates created during the simulation, the values vary slightly with each run and could change greatly if the messages involved were changed, for example by significantly changing the length.

Identity certificate size

The CA issues certificates that are a total of 1369 characters long, of these 256 characters are occupied by the signature and the remaining 1113 characters are the certified information. This means that the certificates issued by the CA have 18.7% overhead at the level of spatial complexity.

Student certificate size

Universities issue certificates that with data lengths of, for example, 329 characters go up to 1711 characters post-certification.

This implies 520% overhead to ensure decentralized authentication of data.

Asymmetric cryptography overhead.

The asymmetric encryption process for ensuring confidentiality and integrity that has been implemented produces, for a message of, say, 159 characters in length, a ciphertext of length 723 characters.

This implies a 454% overhead to ensure confidentiality and integrity with asymmetric encryption.

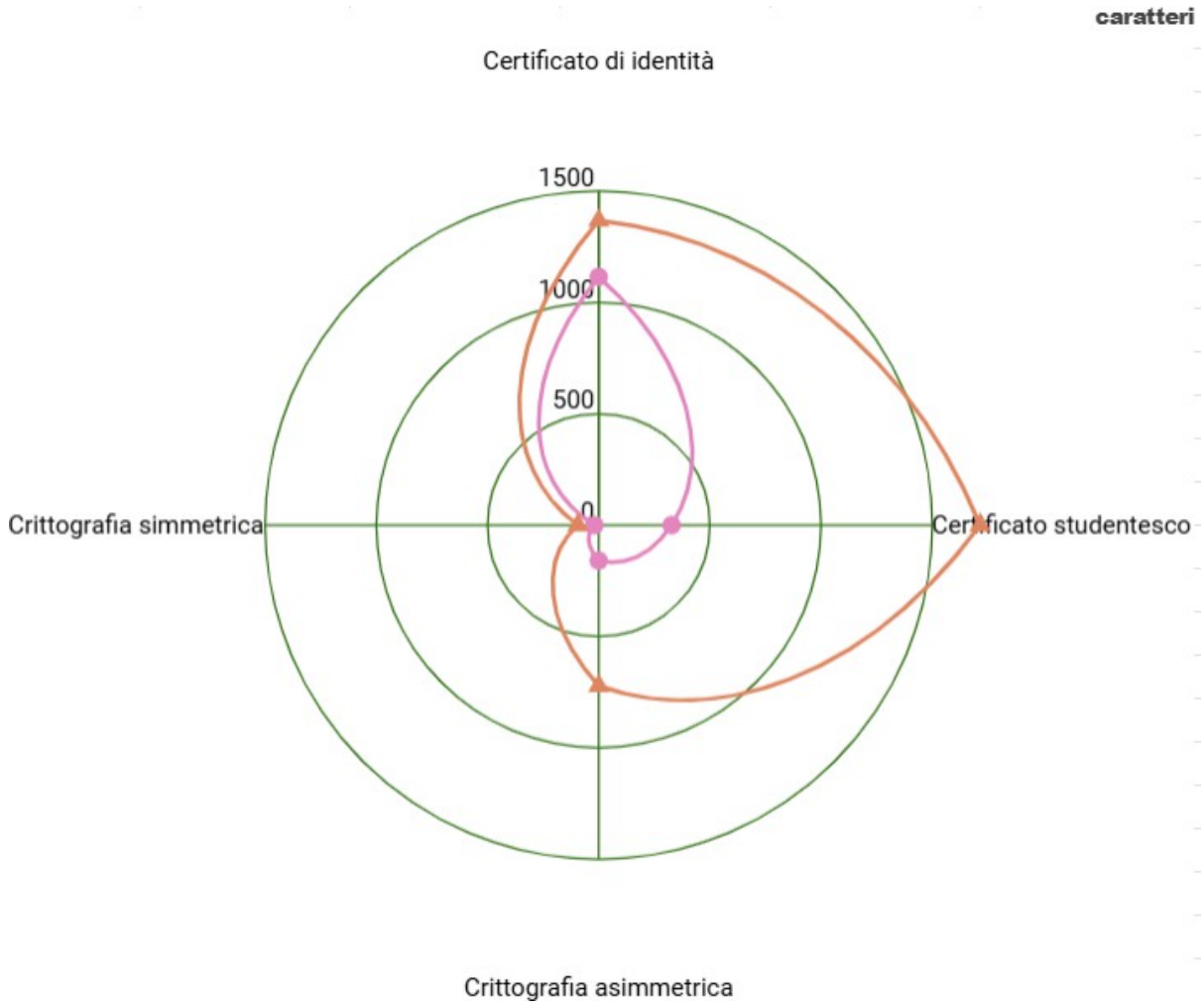
Overhead of symmetric encryption

The symmetric encryption process for ensuring confidentiality and integrity that has been implemented produces, for a message, for example, of length 20 characters, a ciphertext of length 96 characters.

However, a message of length 1711 characters becomes a ciphertext of length 1904 characters.

In the first case we have an overhead of 480% but with a significantly longer message the overhead drops to 111%.

This is in line with the expected result that the overhead of symmetric encryption depreciates faster than that of asymmetric encryption.



Presentation between parties

The secure session key distribution protocol was used for presentation between communicating parties, this implies an overall latency of about 1.5 seconds, but combined with the trust in certificates issued by the Certificate Authority ensures confidential and authenticated communication.

Verification Latency

This section explores the performance at the verification latency level of the implemented asymmetric/symmetric certification and encryption mechanisms.

In order to view the data underlying the following sections, it is necessary to set the constant VERBOSE_MESSAGE_TIME in the documents to True:

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

Note: All data shown are for the messages and certificates created during the simulation, the values vary slightly with each run and could change greatly if the messages involved were changed, for example by significantly changing the length.

Latency of creating an identity certificate

The creation of an identity certificate takes a maximum of 100 milliseconds, according to the recorded data.

Latency of the creation of an academic certificate by the university

Creation of an academic certificate takes, according to recorded data, at most 150 milliseconds.

Latency of verification of an academic certificate by the university

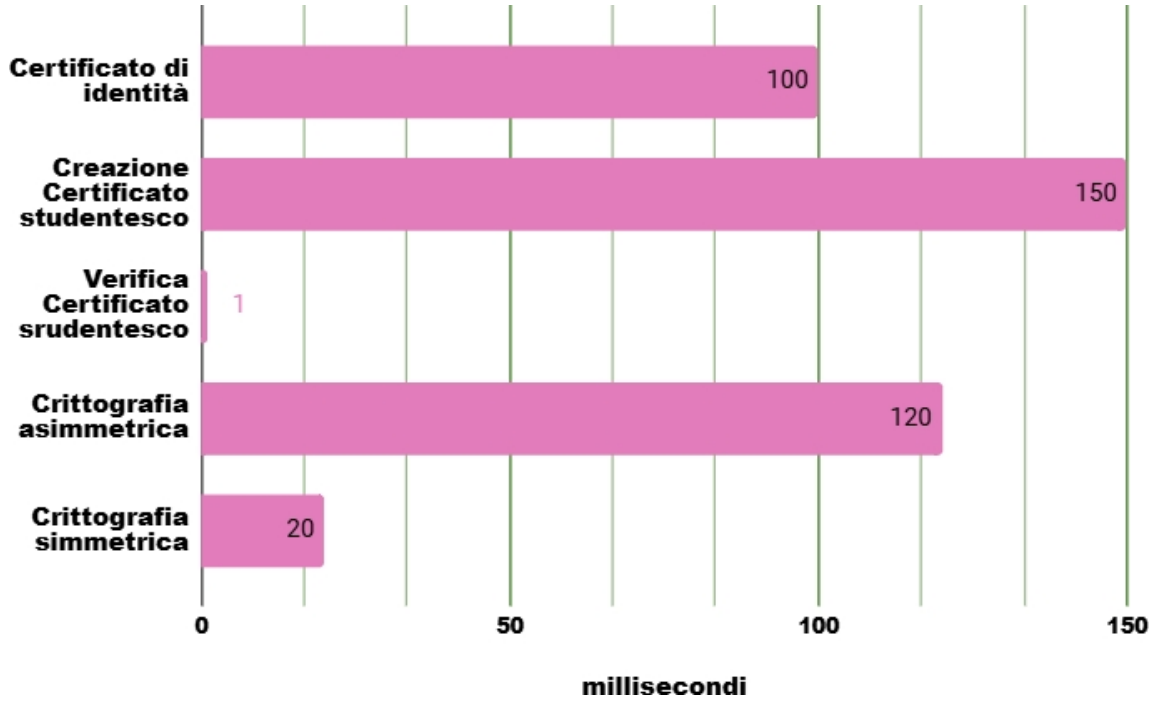
Verification of an academic certificate takes, according to recorded data, about 1 millisecond.

Latency of symmetric encryption

The symmetric encryption process to ensure confidentiality and integrity that has been implemented requires a latency of about 20 milliseconds for encryption and as many for decryption.

Latency of asymmetric encryption

The asymmetric encryption process to ensure confidentiality and integrity that has been implemented requires a latency of about 120 milliseconds for encryption and as many for decryption.


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