FACULTY OF ELECTRONICS,
TELECOMMUNICATIONS
AND INFORMATICS

Bezpieczeństwo Systemów Komputerowych – Projekt
Security of Computer Systems – Project

Tool for Emulating the PAdES Qualified Electronic Signature

Version 1.00, Gdańsk, 02.02.2025

1. The goal of project classes

The main goal of the project is to realize a software tool for emulating the qualified electronic signature, i.e. signing *.pdf documents. The goal is to fully emulate the process, including the hardware toll needed for person identification.

The project is marked regarding the following rules (40 points in total):

- Correct realization of the project before the deadline, presentation during submission – 20 points.
- Technical report – 15 points.
- Presentation the initial stage of project realization (during control meeting) – 5 points.

The details of project evaluation are described in Section 3. and presented in Tab. 1

Each student must select the project group on eNauczanie platform. The group (half of “schedule group”) is automatically assigned to final project submission date. The control meeting is scheduled for full “schedule group”.

2. Project tasks

The main task of the project is to design and develop an application to make a qualified electronic signature according to PAdES (PDF Advanced Electronic Signature) standard concept. In general, the application must take a form of a *set of tools for realization the qualified electronic signature*. The general concept is pointed out in Fig. 1. Additionally to the main application, user has a 2nd auxiliary application for generating a pair of RSA keys and securing the private key. The encrypted private key will be stored on a pendrive and it will be used for signing the document.

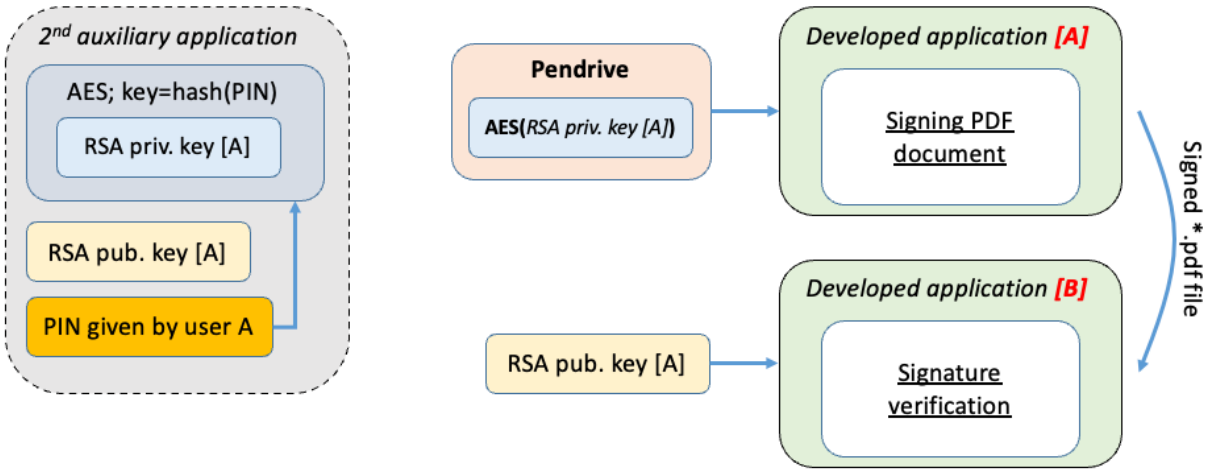

Fig. 1 – Block diagram of the project concept.

General usage scenario:

The user **A** has a hardware tool (a pendrive) with encrypted private RSA key. The key is encrypted by AES algorithm using as a key user's PIN number. The PIN is given by the user during generating a pair of RSA keys using an auxiliary application. The application for PAdES signature must automatically identify the hardware tool and read the private key to sign the *.pdf document. Before signing the document, user **A** must enter the PIN number to decrypt the private RSA key. Regarding the concept of PAdES standard, the electronic signature is attached inside *.pdf document.

The second user **B** must have a possibility to verify the signature (using the same application), by having the public key of user **A** and the signed document. During the verification user **B** generates the hash from the document and verifies it with the hash generated by user **A** (after proper RSA decryption).

Key requirements:

- The GUI interface must allow to select any document (**.pdf*) that will be signed.
- The signature must use the RSA algorithm with a 4096-length key.
- A pseudorandom generator must be used to generate the RSA keys.
- The private key stored on the pendrive must be encrypted by the AES algorithm, where the 256-bit key is the hash from PIN known only to user A.
- Pendrive usage for storing the private RSA key is obligatory. The pendrive must be detected and the encrypted RSA key automatically loaded to the main application.
- The public key can be stored on the hard disk of the computer or be transferred to another physical computer to verify the signature.
- It is obligatory to implement status/message icons to present the state of the application (recognition of hardware tool, reading the private key, signature status).
- It is assumed that only user A can sign the documents, there is no need to create keys for two or more users.
- It is allowed to use the available implementations / libraries of the AES, RSA, SHA algorithms.
- Any language can be used to develop the application.
- In the report, a brief description of performed tests must be included (e.g. signing scenarios, signature verification, encryption/decryption of other files).
- In the report the code of the application must be partially included in a form of listings, pointing out the main functions of the application. It is strongly advised to provide a short and substantive description.
- The full code documentation must be created using Doxygen documentation generator.
- It is obligatory to include the applications code in GitHub repository and provide the link in the final report.
- Application functionality must be presented during project submission.

Note:

- The parameters of the cipher (algorithm type, key size, block size, cipher mode, initial vector) can be set as constants.

3. Project submission and presentation rules

During control meeting each project group (the presence is obligatory for both students) shows current progress in project realization. Additionally, on eNauczanie platform a link to GitHub repository must be included (in dedicated section). Details are presented in Tab. 1.

During final submission each group (the presence is obligatory for both students) presents the project realization, showing application and implemented functionality. After project presentation each group submits a report and provides a link (in the report) to GiHub repository. Details are presented in Tab. 1.

Dates for control meeting and final project presentation are published on eNauczanie platform.

Only one submission date is planned. In a case of obtaining insufficient number of points to obtain a positive mark from the project classes, a 2nd date for submission the project is proposed, but in that case 60% of points in total can be obtained.

Tab. 1 – Detailed project evaluation

PROJECT SUBMISSION – Presentation during classes		
	Task	Points
1	Generation of RSA keys, storing private key in a secure form – 2nd auxiliary application	3
2	Usage of hardware tool (pendrive with encrypted private key) during signing procedure, automatic key detection must be implemented	3
3	Generation of correct signature file – the modified *.pdf with signature details, associated with signed document	4
4	Presenting of correct and incorrect validation of signature by user B (pointing out resistance to document modification).	5
5	Presentation the functioning main and auxiliary applications during project submission	5
REPORTS – The report is evaluated only after project presentation		
6	Partial report (presentation only) for the control meeting (+ code, + presentation during classes)	5
	Minimal requirements: - Presentation: e.g. possibility of generating RSA keys (auxiliary application with GUI), basic project of main application (2 points). - Code in *GitHub* repository (3 points).	
7	Project report (+ code, pointing bibliography in the report)	15
	- Description of realised task (4 points). - Description of key application functionality, pointing out code fragments (4 points). - Code documentation using Doxygen (5 points). - pointing out the bibliography (1 points). - code in *GitHub* repository (no *zip archive allowed) (1 points).	