Highly Dependable Systems

Project 2: Extending the file system to support smartcard authentication

Group 15

70613 - Nuno Nogueira, **70638** - João Sampaio, **73991** - Pedro Braz

New Features

- **Smartcard authentication -** users are registered and authenticated in the system via their Portuguese Citizen Card, enabling their identity to be confirmed by the server and other clients.
- **Client Listing** - the application can list the public keys from all the registered clients.
- **Replay Attack Prevention -** replicating old public key blocks to override newer blocks is no longer possible, by enabling the server and clients to detect and deny replicated content.

Integrity guarantees provided

On receival of a client's certificate at the server, its integrity is validated by cross checking against the CA's certificate chain until a root anchor is found.
At the client side, the same procedure is done during the FS_list procedure so that only verified certificates are returned to the client.

We also added replay attack prevention, identifying out of date public key blocks that were delivered at the client or server, so that these blocks can't be accepted:

- When an ill-intentioned user submits an out of date public key block in place of its original owner;
- When a client receives an old public key block from the server.

When a client submits an updated public key block, he inserts a sequential identifier in the block, to identify its age. At the server side, the identifier is checked against the previous public key block, if it exists. If the new block doesn't have an identifier higher than the previous one, it is refused.

When the client requests a public key block he inserts a random id in the request header. At the server, the random id and the block's content are hashed (id + PKBlock) and placed in the response header. When the client receives the public key block he checks if the hash in the header matches the hash of the returned content with the id.

If the hash is different, then it is not the same request made by the client, enabling him to detect the replay attack.

Threat Model

Our updated system is able to guarantee file integrity under new scenarios:

- **Replay attacks** where an outdated public key block is resubmitted to the server by a malicious user in place of the original client, overwriting the updated data. And where a client reads from a public key block, which is replaced by outdated data.

- **Invalid Certificate** Where a client submits an invalid certificate for its key, and the client's identity can't be verified.

The following tests are performed on the system to evaluate the correct detection of integrity failures:

- **Receive invalid certificate on readPubKeys** - The client calls the server with readPubKeys to received the certificates from all registered users. The server will respond with a certificate that can't be verified. The client needs to invalidate those certificates;

- **Receive outdated public key data on get -** The client requests a public key block from the server, sending a token. The server then returns a block with an invalid session token hash, as in a replay attack. The client won't accept the public key block.

- **Send invalid certificate to server** - The client initializes the file system with an invalid certificate. The server won't accept this client.