Security Overview

Security
Department of Electronics, Telecommunications and Informatics
University of Aveiro

José Pedro Almeida Duarte

November 11, 2017

Security Messaging Repository System

1 Abstract

"The objective of this project is to develop a system enabling users to exchange messages asynchronously. Messages are sent and received through a non-trustworthy central repository, which keeps messages for users until they fetch them. The resulting system is composed by a Rendezvous Point, or Server, and several clients exchanging messages."

2 Overview

2.1 Architecture

2.1.1 Server

Rendezvous point for all clients to connect.

2.1.2 Multiple Clients

Used for users to interact.

2.1.3 PKI

Public Key Infrastructure to define certification hierarchies, emit and distribute public key certificates and distribute revoking certificates.

2.1.4 Nonce Generator

Provides an arbitrary large number and guarantees that it had never been used before.

2.2 Specifications

2.2.1 Setup a session key between a client and the server prior to exchange any command/response

To setup a session key between a client and the server, a symmetric key will be generated in order to be possible for them to exchange commands and/or responses securely. This key will be valid only for that session and will be discarded when it ends. The symmetric keys created will be AES. It has strong security because none of the known attacks are practically possible. The cipher mode used will be CFB because it introduces confusion, it allows the reuse of the same key for different messages without increasing the risk, does not show any pattern from the original text and the successive change in the original message does not produce an conclusive result to crypto analysts or it is too hard to obtain. The value provided as the initialization vector will be generated by a Nonce generator so that the same initialization vector won't be used twice. To securely distribute

the symmetric key, the Diffie-Hellman algorithm will be used. Diffie-Hellman public values will be signed with the user's citizen card.

2.2.2 Authentication and integrity control of all messages exchanged between client and server

To ensure authentication and integrity control of all messages, the "Encryption + Authentication" will be used, more specifically the Encrypt-then-MAC because it's considered to be the most secure and provides integrity for both ciphertext and plaintext.

2.2.3 Add to each server reply a genuineness warrant

To prove that the reply is the correct one for the client's request, each command and/or reply will be authenticated using a MAC (Message Authentication Code). This MAC is produced given the symmetric key (session key), the message and the message ID. The last one prevents the same MAC to be obtained in case of the same user sends a message with the exact same content to the same user.

2.2.4 Register relevant security-related data in a user creation process

During the user creation process, the "secdata" field will be filled with both public key value and Diffie-Hellman value. Private key and private Diffie-Hellman value will be stored in the user's directory hashed by an user defined password (this password won't be stored anywhere, it's only remembered by memory and it's user's responsibility). All keys will be authenticated using the private key of the user's Citizen Card.

2.2.5 Involve the Citizen Card in the user creation process

When an user is the user is created, his keys will be also created and authenticated using his Citizen Card.

2.2.6 Encrypt messages delivered to other users

To encrypt messages delivered to other users, a hybrid cryptosystem will be used in order to use the efficiency of symmetric keys allied to the convenience of asymmetric keys. The message will be then encrypted with a combined symmetric key and, therefore, this symmetric key will be signed by the (asymmetric) public key of the recipient which can access it with his private key. It is needed that the sender is able to prove the authenticity of the public key of the recipient. The symmetric keys will be AES with CFP as cipher mode as like the session keys. To cipher the symmetric keys, RSA asymmetric keys will be used.

2.2.7 Signature of the messages delivered to other users and validation of those signatures

To sign the messages delivered to other users, will be used digital signatures. These signatures will be accomplished using the private key of the sender to cipher the message. This private key is obtained from the Citizen card. To validate the signature, the recipient can use the public key of the sender, the one corresponding to the same pair of asymmetric keys which the secret key used to cipher the message is part of.

2.2.8 Encrypt messages saved in the receipt box

An user will encrypt all his messages in the receipt box by using his own public key what makes him the only one who can decrypt it (using the corresponding private key).

2.2.9 Send a secure receipt after reading a message

When an user receives and validates a message, he sends a receipt to the sender proving that he read it. This receipt is encrypted with the public key of the sender so that he is the only one who can decrypt it and contains the plaintext of the received message signed with his own private key so that that sender can authenticate it.

2.2.10 Check receipts of sent messages

It is issued a STATUS message to the server. The user will get a list of messages he sent and corresponding receipts. He is able to decrypt and check these receipts using the public key of the corresponding user to which the message had been sent.

2.2.11 Proper checking of public key certificates from Citizen Cards

The public key certificates are validated through a certification chain. A Certificate Authority will be created to sign digital certificates. Users will trust that Certificate Authority and will need to validate all chain until get there. Will use OpenSSL Certificate Authority to create and manage the Certificate Authority.

2.2.12 Prevent a user from reading messages from other than their own message box

Users won't be able to read messages from other message box than their own because all messages are ciphered with the proper receiver's public key.

2.2.13 Prevent a user from sending a receipt for a message that they had not read

The receipts have to contain the message that had been read in order to prove that that message had in fact been read.