Assignment 11

IDATT2503 - Security in programming and cryptography
Fall 2023

Francesca Grimaldi

1 Exercise 1

1.1 What is Alice's private key?

To compute Alice's private key, one has to calculate the multiplicative inverse `d` to `e`, modulo $(p-1)(q-1)$. In this case, it would be

$$d \times 3 \equiv 1 \mod 46 \times 82.$$

Using Euclid's extended algorithm, we find that $d = 2515$. Alice's private key is the tuple (n, d), with $n = p \times q$, therefore $(3901, 2515)$.

1.2 Verify if 964 is a valid signature of Alice for the message 100.

To verify it, we decrypt 964 using Alice's public key, so we compute

$$964^3 \mod 3901.$$

The result is 100, meaning that the number 964 is indeed a valid signature for that message.

1.3 Alice wants to encrypt and sign the message 100 to Bob. What are the exact steps? Assume Bob has the RSA with public key $(n_B, e_B) = (3127, 33)$.

First, Alice encrypts the message using Bob's public key.

$$C \equiv M^{e_B} \mod n_B$$

$$C \equiv 100^{33} \mod 3127$$

Which gives us $C \equiv 487$. Then, she signs the message using her private key.

$$S \equiv C^{d_A} \mod n_A$$

$$S \equiv 487^{2515} \mod 3901$$

Which gives us $S \equiv 964$. This is a valid signature for the message as previously verified. She then sends both the encrypted message and the signature.

2 Exercise 2

Both MACs (Message Authentication Codes) and Digital Signatures are cryptographic techniques used to verify that a message came from the stated sender and that it has not been altered.

One of the key differences is that MACs are used to *authenticate* messages and *verify* their integrity, whereas Digital Signatures also ensure *non-repudiation.* This means that a sender cannot deny authenticity, as their private key is used to encrypt the hash tag.

MACs typically use symmetric key cryptography, whereas Digital Signatures work with asymmetric-key cryptography (an example is the RSA-based signature).

While MACs require shared keys between the parties exchanging the messages, Digital Signatures do not. For this reason, the latter option is more suitable when the messages must be verified but there is no way to securely share the common secret key. MACs are employed, for example, in the context of network communication protocols.

3 Exercise 3

3.1 Name the two parts/layers of TLS 1.3, and what their purporses are

The TLS 1.3 (Transport Layer Security) is a cryptographic protocol used in client-server communications. It consists of two layers: the TLS Handshake and the TLS Record protocols.

- The Handshake protocol is utilized when a connection starts, as part of the setup to send all the necessary data that both parties need in order to share actual application data. Cipher and hash functions are established.
- The Record protocol is responsible for application data security as well as integrity and origin verification. It applies MAC and encrypts outgoing messages, and verifies and decrypts the ingoing ones.

3.2 Explain how TLS uses both symmetric and asymmetric cryptography

To securely generate the session key, TLS uses asymmetric cryptography with algorithms like Diffie-Hellman or its variants (such as ECDHE).

A secure connection is then started using the newly established key, using a symmetric cipher, for example AES, to encrypt the majority of the data exchanged by client and server.

Moreover, TLS typically uses Digital Signatures (which work with asymmetric-key cryptography) for server authentication.

4 Exercise 4

4.1 What are the two main types of password attacks?

The attacks aimed at cracking passwords are mainly brute force attacks (with their variants such as dictionary attacks and use of common patterns). These can be divided in two types:

1. Online attacks, where an attacker uses the same interface (or a very similar one) that users use to log in to the system.
2. Offline attacks, where the attackers may conduct the attack on their own hardware because they have gained access to files containing information about user accounts and passwords.

4.2 Why should passwords not be stored encrypted?

First of all, if passwords are stored encrypted and an attacker gains access to both the list of encrypted passwords and the decryption key, they could potentially obtain all users' passwords at once.

Moreover, most encryption algorithms are block ciphers that encrypt in blocks of n characters. With this system, passwords of different lengths are encrypted into strings of different lengths. Storing them encrypted would inevitably reveal information about their length, making it easier for an attacker to crack them.

4.3 What is the advantage to use a specialized password algorithms over general cryptographic hash functions using a salt?

The main advantage is that specialized password algorithms are designed to be computationally intensive and adaptive, using a lot of memory or RAM with the purpose of making password attacks slower. These algorithms are also intended to remain secure even as hardware capabilities evolve, therefore are more suitable with respect to cryptographic hash functions using salt.