Advanced Cryptography — Midterm Exam

Serge Vaudenay

14.4.2022

- duration: 1h45
- any document allowed
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will **<u>not</u>** answer any technical question during the exam
- readability and style of writing will be part of the grade

1 2-Move Authenticated Key Agreement

The traditional Diffie-Hellman key agreement scheme is a 2-move protocol. The interface can be modeled in the following way:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$ sets up public parameters pp using a security parameter λ.
- $\mathsf{Send}(\mathsf{pp}) \to (\mathsf{esk}, \mathsf{epk})$ generates an ephemeral key pair where epk is to be sent to the counterpart.
- $\mathsf{Receive}(\mathsf{pp}, \mathsf{esk}, \mathsf{epk}) \to K$ receives the counterpart's epk and generates the shared key K.

Both participants are supposed to send and receive to derive their local K. The protocol is meant to resist to passive attacks with the notion of key indistinguishability. We extend this primitive in order to authenticate the final key by using a long-term key. For this, we add the following algorithm in the interface:

- $\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{lsk}, \mathsf{lpk})$ generates a long term key pair for a user, where lpk is publicly associated to the user and lsk is kept secret.

In addition, Send takes as input the long-term secret of the user and $\mathsf{Receive}$ takes as input the long-term public key of the counterpart.

Q.1 Rewrite the entire interface and define the correctness notion using a fully specified game.
To model security against active attacks, we can no longer assume that the protocol is honestly executed and give the transcript to the adversary. Instead, we use oracles to model honest Alice honest Bob running Send and $\mathsf{Receive}$. These oracles shall allow multiple concurrent sessions. Hence, we consider the game in Fig. 1.
The instruction **ensure** tests if the following predicate is true and causes the oracle to return $\perp$ if it is not the case.

Q.2 Fully define the key indistinguishability notion based on this game.
Motivate why $\mathsf{OReceive}$ returns whether $K_P \neq \perp$.
Explain why OTest ensures $K_1 \neq \perp$.

```
Game $\Gamma_b$:
 1: initialize state to empty
 2: $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$
 3: $\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{lsk}_A, \mathsf{lpk}_A)$
 4: $\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{lsk}_B, \mathsf{lpk}_B)$
 5: $\mathcal{A}^{\mathsf{oracles}}(\mathsf{pp}, \mathsf{lpk}_A, \mathsf{lpk}_B) \to z$
 6: return $z$

Oracle $\mathsf{OReceive}(P, \mathsf{sid}, \mathsf{lpk}, \mathsf{epk})$:
 7: ensure $P \in \{A, B\}$
 8: ensure $\mathsf{state}[P, \mathsf{sid}]$ exists with only two elements
 9: $\mathsf{state}[P, \mathsf{sid}] \to (\mathsf{esk}_P, \mathsf{epk}_P)$
10: $K_P \leftarrow \mathsf{Receive}(\mathsf{pp}, \mathsf{esk}_P, \mathsf{lpk}, \mathsf{epk})$
11: select $K_0$ at random
12: $\mathsf{state}[P, \mathsf{sid}] \leftarrow$
        $(\mathsf{esk}_P, \mathsf{epk}_P, \mathsf{lpk}, \mathsf{epk}, K_0, K_P)$
13: return $1_{K_P \neq \perp}$

Oracle $\mathsf{OSend}(P, \mathsf{sid})$:
14: ensure $P \in \{A, B\}$
15: ensure $\mathsf{state}[P, \mathsf{sid}]$ does not exist
16: $\mathsf{Send}(\mathsf{pp}, \mathsf{lsk}_P) \to (\mathsf{esk}_P, \mathsf{epk}_P)$
17: $\mathsf{state}[P, \mathsf{sid}] \leftarrow (\mathsf{esk}_P, \mathsf{epk}_P)$
18: return $\mathsf{epk}_P$

Oracle $\mathsf{OTest}(P, \mathsf{sid})$:
19: ensure $P \in \{A, B\}$
20: ensure $\mathsf{state}[P, \mathsf{sid}]$ exists with six elements
21: $\mathsf{state}[P, \mathsf{sid}] \to$
        $(\mathsf{esk}_P, \mathsf{epk}_P, \mathsf{lpk}, \mathsf{epk}, K_0, K_1)$
22: ensure $K_1 \neq \perp$
23: return $K_b$
```

Fig. 1. Key indistinguishability game

Q.3 By using an adversary who makes Alice and Bob honestly execute the protocol, prove that security in the sense of the above game can easily be broken.
Propose a way to fix the game to get a sound security notion.

Q.4 Propose a protocol. Note: we do not require a security proof. The grade for this question will depend on the security of the proposed protocol.

2 Redundant-RSA Decryption

Let n be an RSA modulus of unknown factorization. We know that given $(x+1)^3 \bmod n$ and $x^3 \bmod n$ we can easily compute $x \bmod n$.

Q.1 Given n, $a = (x+1)^5 \bmod n$, and $b = x^3 \bmod n$, show how to compute $x \bmod n$ efficiently. Hint: x is a root of any polynomial which is a combination of $(z+1)^5 - a$ and $z^3 - b$ in $\mathbf{Z}_n$.