ETH Zürich, D-INFK
Spring 2019

Prof. Ueli Maurer
Fabio Banfi
Daniel Jost
Jiamin Zhu

Cryptography Foundations

Exercise 3

3.1 Combining Cryptographic Schemes

Goal*: Cryptographic schemes should be composed carefully. We examine a concrete scenario that causes vulnerabilities and discuss security requirements for schemes combining encryption and authentication.*

a) Let (E, d) be a CPA-secure encryption scheme and f a secure MAC-function (under a chosen-message attack), again with the usual understanding of security as hardness of the respective problems (CPA bit-guessing problem and MAC forgery game, respectively).

Assume someone combines these schemes and sends the pair (c, t) of the ciphertext $c = E(m, k)$ and the tag $t = f(m, k')$, computed with two different keys k, k'. This methodology is called *Encrypt-and-MAC*. Show that it is possible that an adversary learns the full message that is sent. Which additional property must the MAC have to prevent this?

As the plaintext (as opposed to the ciphertext) is authenticated, a modified ciphertext will still be accepted by the receiver if it decrypts to the same plaintext. Suppose the protocol aborts the connection in case a MAC verification fails, and such an abort is visible to an attacker, e.g., by observing a connection reset (in the TCP connection).

b) Assume that the MAC does not exhibit the weakness discussed in **a)**. Show that the error message can still be exploited to attack the security of the protocol, even if the encryption scheme is secure according to the standard definitions!

Hint: Devise a tailor-made encryption scheme, e.g., use the one-time pad with an encoding that expands the plaintext before encrypting.

3.2 Computing Discrete Logarithms in Specific Groups

Goal*: Discover that, while computing discrete logarithms is believed to be hard for certain groups, it is actually easy in other groups.*

Let $\mathbb{G}$ be a cyclic group of order $n = |\mathbb{G}|$. For a generator g of $\mathbb{G}$ and an element y in $\mathbb{G}$ we want to compute the discrete logarithm x of y to the base g. This means to find the $x \in \{0, \ldots, n-1\}$ such that $y = g^x$.

a) Show that when $\mathbb{G} = \mathbb{Z}_n$ is the additive group of integers modulo n, discrete logarithms can easily be computed.

b) Assume that you have an efficient algorithm to compute discrete logarithms to the base g in $\mathbb{G}$. Let $h \in \mathbb{G}$ be another generator of $\mathbb{G}$. Show how you can use your algorithm to efficiently compute discrete logarithms to the base h.

c) Assume that $\mathbb{G}$ has even order, so $n = 2m$. Show how, in this case, you can efficiently compute whether the discrete logarithm x of y to the base g is even or odd. This means to compute $x \bmod 2$.

d) Suppose now that the order of $\mathbb{G}$ is divisible by 2^k, so $n = 2^k m$. Construct an efficient algorithm to compute $x \bmod 2^k$.

e) Now generalize your idea of **c)** to the case of other small prime divisors. So, give an efficient algorithm to compute $x \bmod p^k$ if p^k divides n (and p is small compared to n).

f) Finally show how to efficiently compute x if all prime divisors of n are small.

3.3 Diffie-Hellman Key-Agreement

Goal*: Understand that the decisional Diffie-Hellman (DDH) assumption is sufficient for the Diffie-Hellman key-agreement to be secure.*

Consider the Diffie-Hellman construction presented in the lecture using a cyclic group $\mathbb{G} = \langle g \rangle$. In the following, let $\mathsf{AUT}_1^{\rightarrow}$ denote the authenticated single-message (i.e., $t = 1$) channel specified in Exercise 2.2, and let $\mathsf{AUT}_1^{\leftarrow}$ denote the same channel but from Bob to Alice.

a) Specify the initiator's converter $\mathsf{dh_I}$ and the responder's converter $\mathsf{dh_R}$ that are attached at interfaces A and B of the assumed resource. That is, specify the converter used in the expression $\mathsf{dh_I^A}\,\mathsf{dh_R^B}\,[\mathsf{AUT}_1^{\rightarrow}, \mathsf{AUT}_1^{\leftarrow}]$.

b) Now also specify the constructed key-resource $\mathsf{KEY}_{\mathbb{G}}$. Note that this resource has to differ slightly from the one introduced in Exercise 2.2. In particular, pay attention to when the key is output and what information is leaked at interface E.

c) Show that there exists a simulator sim such that for every distinguisher D there exists another one D' (internally using D) such that

$$\Delta^{D}\big(\mathsf{dh_I^A}\,\mathsf{dh_R^B}\,[\mathsf{AUT}_1^{\rightarrow}, \mathsf{AUT}_1^{\leftarrow}]\,, \mathsf{sim^E}\,\mathsf{KEY}_{\mathbb{G}}\big) = \Delta^{D'}\big(\mathsf{DDH}^0, \mathsf{DDH}^1\big),$$

where $\langle \mathsf{DDH}^0 \,|\, \mathsf{DDH}^1 \rangle$ denotes the decisional Diffie-Hellman problem for $\mathbb{G}$ described in Definition 3.10.