ETH Zürich, D-INFK
Spring 2019

Prof. Ueli Maurer
Fabio Banfi
Daniel Jost
Jiamin Zhu

Cryptography Foundations
Solution Exercise 6

6.1 The Lamport One-Time Signature Scheme

Let the key-pair distribution be the distribution induced by choosing $x_0^1, x_1^1, x_0^2, x_1^2, \dots x_0^n, x_1^n \in \mathcal{X}$ uniformly at random, setting $y_b^i = f(x_b^i)$ for $b \in \{0,1\}$, $i \in \{1,\dots,n\}$, and letting the verification key and signing key be $\mathrm{v} = (y_0^1, y_1^1, \dots y_0^n, y_1^n) \in \mathcal{V}$ and $\mathrm{z} = (x_0^1, x_1^1, \dots x_0^n, x_1^n) \in \mathcal{Z}$, respectively. Further let for $m = (m_1, \dots, m_n) \in \mathcal{M}$,

$$\sigma(m, \mathrm{z}) = (x_{m_1}^1, x_{m_2}^2, \dots, x_{m_n}^n) \in \mathcal{S}$$

and for $m \in \mathcal{M}$ and $(s_1, \dots, s_n) \in \mathcal{S}$,

$$\tau\big(m, (s_1, \dots, s_n), \mathrm{v}\big) = \begin{cases} 1, & \forall i \in \{1,\dots,n\} \quad f(s_i) = y_{m_i}^i \\ 0, & \text{otherwise.} \end{cases}$$

Note that the scheme is correct, i.e., $\tau\big(m, \sigma(m, \mathrm{z}), \mathrm{v}\big) = 1$ for all $m \in \mathcal{M}$ and all v, z generated according to the key-pair distribution, because $f(x_{m_i}^i) = y_{m_i}^i$ for all $i \in \{1,\dots,n\}$.

Now let $\mathcal{A}$ be an adversary for the signature forgery game for 1 message with success probability α. We can assume without loss of generality that $\mathcal{A}$ asks for exactly one message to be signed: If $\mathcal{A}$ asks for no signature, consider the adversary $\mathcal{A}'$ that does the same as $\mathcal{A}$, but when $\mathcal{A}$ wants to submit a forgery attempt (m', s'), $\mathcal{A}'$ before asks for a signature for an arbitrary $m \neq m'$, ignores the result, and then submits the forgery attempt (m', s'). We then clearly have that $\mathcal{A}'$ also has success probability α.

We define an algorithm for the inversion game for f as follows:

- Take as input a value $y \in \mathcal{Y}$.
- Choose $b' \in \{0,1\}$ and $i' \in \{1,\dots,n\}$ uniformly at random.
- For all $(i,b) \in \big(\{1,\dots,n\} \times \{0,1\}\big) \setminus \{(i',b')\}$, choose $x_b^i \in \mathcal{X}$ uniformly at random, and set $y_b^i \coloneqq f(x_b^i)$.
- Let $y_{b'}^{i'} \coloneqq y$.
- Give $\mathrm{v} \coloneqq (y_0^1, y_1^1, \dots y_0^n, y_1^n)$ as verification key to the adversary $\mathcal{A}$.
- When $\mathcal{A}$ asks for a signature for $m \in \mathcal{M}$:
  - If $m_{i'} \neq b'$, give $(x_{m_1}^1, x_{m_2}^2, \dots, x_{m_n}^n)$ as a signature to $\mathcal{A}$.
  - Otherwise, give up.
- When $\mathcal{A}$ submits a forgery attempt $(m', s') \in \mathcal{M} \times \mathcal{S}$, return $s'_{i'}$.

Note that the distribution of v matches the distribution of verification keys in the signature forgery game. Furthermore, the distribution of v does not depend on b' and thus, b' and the query m from $\mathcal{A}$ are independent. This implies that $m_{i'} \neq b'$ with probability $\frac{1}{2}$. In this case, $(x_{m_1}^1, x_{m_2}^2, \dots, x_{m_n}^n)$ is a valid signature for m.

If $\mathcal{A}$ wins the game, s' is a valid signature for m' and $m' \neq m$. Hence, the messages m and m' differ in at least one position. Since i' is uniform and independent from m and m', we have $m_{i'} \neq m'_{i'}$ with probability at least $\frac{1}{n}$. In this case and if $m_{i'} \neq b'$, we have $m'_{i'} = b'$. Since s' is a valid signature for m', we further have $f(s'_{i'}) = y_{b'}^{i'} = y$. Therefore, $s'_{i'}$ is a preimage of y under f. We conclude that the winning probability of our algorithm is at least $\frac{\alpha}{2n}$.

6.2 Signature Schemes from Trapdoor One-Way Permutations

a) Only knowing the RSA public key (n, e) one could select an arbitrary $s \in \mathbb{Z}_n^*$ and compute $m \coloneqq s^e \mod n$. (m, s) is a valid message-signature pair.

b) Given a concrete message m, the above attack does not work any more. But we can still find a forgery as follows: Choose $r \in \mathbb{Z}_n$ uniformly at random. If $\gcd(r, n) \neq 1$, then the scheme is broken (since we factor n). Otherwise, obtain the signatures s_1,s_2 on messages $m_1 \coloneqq r$ and $m_2 \coloneqq m \cdot r^{-1}$, respectively. Due to the homomorphic property of the RSA function, $s \coloneqq s_2 \cdot s_1 = (m \cdot r^{-1})^d \cdot r^d = m^d$ is a signature on m.

c) The inversion winner W' expects the problem instance parameter p and the problem instance $y = f(x, p)$ for a uniformly chosen value x. Note that y is also uniformly random over the set $\mathcal{Y}$ since $f(\cdot, p)$ is a bijection from $\mathcal{X}$ to $\mathcal{Y}$. The task of the winner W' is to find x with the help of an assumed winner W of the fixed-message signature forgery game $G_{t,\tilde{m}}^{\mathsf{sig\text{-}fix}}$ (in the random oracle model).

To this end, it emulates towards W an execution of $G_{t,\tilde{m}}^{\mathsf{sig\text{-}fix}}$. W' defines p as the public (verification) key and provides it to W. W' also manages an array of hashes $(y_1, \ldots, y_{t+1})$, and an array of signatures $(s_1, \ldots, s_{t+1})$, both initially set to $(\bot, \ldots, \bot)$.

The winner W will make two kinds of queries: signing queries and hash queries.

- Upon the ith query m_i by winner W:
  1. If it is a hash-oracle query:
     - If $m_i = \tilde{m}$, i.e., the fixed message, then set $y_i \coloneqq y$ and return y_i. That is, the problem instance is embedded into the output of the random oracle.
     - If $m_i \neq \tilde{m}$, then check whether there is an index $k < i$ s.t. $m_k = m_i$, then set $y_i \coloneqq y_k$, $s_i \coloneqq s_k$, and return y_i. Otherwise, (i.e., $m_i \neq m$ and m has not been queried before), choose a uniformly random element $s \in \mathcal{X}$, set $s_i \coloneqq s$ and define (and return) $y_i \coloneqq f(s_i, p)$. (Note that with this "trick" the reduction knows the pre-image s_i of y_i—i.e., the signature of m_i—without the trapdoor).
  2. If it is a signing query (and thus $m_i \neq \tilde{m}$):
     - W' proceeds as above to obtain the (emulated) random oracle output y_i and the corresponding pre-image s_i of y_i, that is, checks whether m_i has been queried before and otherwise samples them freshly.
     - W' outputs s_i as the signature.

Finally, upon the forgery output $\hat{s}$ by winner W, W' outputs $\hat{s}$ as the pre-image of y.

It remains to convince ourselves that W' succeeds with the same probability as W. To this end, first observe that W' correctly emulates the random oracle towards W: for each m_i, $h(m_i)$ is distributed uniformly over $\mathcal{Y}$, since $f(\cdot, p)$ is a bijection from $\mathcal{X}$ to $\mathcal{Y}$ and s_i is chosen uniformly at random. The also holds for $h(\tilde{m}) \coloneqq y$. Moreover, all the entries are independent. Thus, W' correctly emulates the random oracle towards W. Now consider the signing queries. For m_i, W' provides W with the signature s_i, where we have that $y_i = h(m_i) = f(s_i, p)$, which is the correct signature. Finally, W provides a valid signature

$\tilde{s}$ for $\tilde{m}$ if and only if $y = h(\tilde{m}) = f(\tilde{s}, p)$, hence W' wins the TOWP inversion game if and only if W wins $G^{\mathsf{sig\text{-}fix}}_{t,\tilde{m}}$.

d) We only provide an informal argument. A formal proof would be quite tedious.

When considering a winner W for the normal t-message forgery game G^{sig}_t, we do not know beforehand for which message W will provide a forgery. Thus, we do not know where in the random oracle to embed the challenge y. The best strategy is hence to simply guess the position. Obviously we do not want to guess the message, but we simply guess the index of the corresponding hash query (we can assume without loss of generality that W queries the hash function on $\tilde{m}$ before submitting a forgery attempt for $\tilde{m}$). Since in the execution of the game there are at most $t + 1$ hash queries involved (the last one for checking the forgery), intuitively we should guess correctly with probability $\frac{1}{t+1}$, and thus W' should have success probability $\frac{\alpha}{t+1}$.

This argument however assumes that the choice of $\tilde{m}$ by W is independent of our guessed index. Thus, the guess must not affect the observable behavior of the emulated forgery game towards W. In principle, y has the same distribution as the emulated y_i and hence W cannot tell them apart. If we however embed the challenge for a hash query m', i.e., set $h(m') = y$ and then W later chooses to sign the message m', then we cannot provide him with a valid signature. Since at this point (where W can now observe our guess) it is already clear that we guessed wrong, this however does not affect the success probability.

6.3 The Boneh–Lynn–Shacham Signature Scheme

a) The corresponding signature verification function $\tau\colon \mathcal{M} \times \mathbb{G} \times \mathbb{G} \to \{0, 1\}$ is defined as follows:

$$\tau(m, s, \mathrm{v}) \coloneqq \mathrm{E}(g, s) \stackrel{?}{=} \mathrm{E}(\mathrm{v}, h(m)),$$

where $\mathrm{v} \in \mathbb{G}$ denotes the verification key and $s \in \mathbb{G}$ the signature.

We now argue that this scheme is correct. To this end, consider a key-pair $(\mathrm{v}, \mathrm{z}) = (g^x, x)$. Since $h(m) \in \mathbb{G}$, there exists a $y \in \mathbb{Z}_n$ such that $g^y = h(m)$. The signature s is then computed as $s = \sigma(m, x) = (h(m))^x = g^{xy}$. In the verification, we then have $\mathrm{E}(g, s) = \mathrm{E}(g, g^{xy}) = \mathrm{E}(g, g)^{xy} = g_T^{xy}$ and $\mathrm{E}(g^x, h(m)) = \mathrm{E}(g^x, g^y) = \mathrm{E}(g, g)^{xy} = g_T^{xy}$, and thus the check succeeds.

b) In the signature forgery game, the winner W gets to see the verification key $\mathrm{v} = g^x$ for an x chosen uniformly at random from $\mathbb{Z}_n$. He can then query signatures for t messages $m_1, \ldots, m_t$ and finally submits a forgery attempt (m, s) with $m \neq m_i$ for all $i \leq t$. Observe that in the random oracle model, $h(m) = g^y$ for a $y \in \mathbb{Z}_n$ that is chosen uniformly at random and, importantly, independent of $h(m_i)$ for all $i \leq t$. He wins, if and only if $\mathrm{E}(g, s) = \mathrm{E}(\mathrm{v}, h(m))$, where

$$\mathrm{E}(g, s) = \mathrm{E}(g, g^z) = \mathrm{E}(g, g)^z = g_T^z$$
$$\mathrm{E}(\mathrm{v}, h(m)) = \mathrm{E}(g^x, g^y) = \mathrm{E}(g, g)^{xy} = g_T^{xy}.$$

That is, he wins if and only if $z = xy \pmod{|\mathbb{G}_T|}$. Since $|\mathbb{G}_T| = |\mathbb{G}| = n$, this corresponds exactly to solving the CDH problem in $\mathbb{G}$.

To prove this claim more formally, one would need to show a reduction analogous to **6.2 c)** and **d)**, where the reduction cleverly chooses the values in the random hash system $\mathbf{H}$, such that it can sign without being able break the CDH problem.