A weaker notion of security
Lamport’s one-time signature scheme

Foundations of Cryptography
Computer Science Department
Wellesley College

Fall 2016

Table of contents

Introduction

One-time Signatures

Lamport
Security of signature schemes

Let $\Pi = (\text{Gen}, \text{Sign}, \text{Vrfy})$ be a signature scheme.

The signature experiment $\text{Sig-forge}_{A,\Pi}(n)$:

1. $\text{Gen}(1^n)$ is run to obtain keys $(pk, sk)$.
2. Adversary $A$ is given $pk$ and oracle access to $\text{Sign}_{sk}(\cdot)$. The adversary then outputs $(m, \alpha)$. Let $Q$ denote the set of messages whose signatures were requested by $A$ during its execution.
3. The output of the experiment is defined to be 1 if and only if (1) $\text{Vrfy}_{pk}(m, \alpha) = 1$, and (2) $m \notin Q$

Definition 12.2. A signature scheme $\Pi = (\text{Gen}, \text{Sign}, \text{Vrfy})$ is existentially unforgeable under an adaptive chosen-message attack if for all PPT adversaries $A$, there exists a negligible function $\negl$ such that

$$\Pr[\text{Sig-forge}_{A,\Pi}(n) = 1] \leq \negl(n).$$

Good enough by golly

- Definition 12.2 is the gold standard of security for digital signature schemes.
- However, weaker notions of security may be appropriate for certain restricted application and as building blocks for scheme satisfying stronger notions of security.
- Today we study a that is secure as long as it’s only used to sign one message.
One last experiment

The one-time signature experiment \( \text{Sig-forge}_{A, \Pi}^{1-time}(n) \).

1. \( \text{Gen}(1^n) \) is run to obtain keys \((pk, sk)\).
2. Adversary \( A \) is given \( pk \) and asks a single query \( m' \) to oracle \( \text{Sign}_{pk}(\cdot) \). \( A \) then outputs \((m, \alpha)\) where \( m \neq m' \).
3. The output of the experiment is defined to be 1 if and only if \( \text{Vrfy}_{pk}(m, \alpha) = 1 \).

Definition 12.14 A signature scheme \( \Pi = (\text{Gen}, \text{Sign}, \text{Vrfy}) \) is existentially unforgeable under a single-message attack, or is a one-time signature scheme, if for all PPT adversaries \( A \), there exists a negligible function \( \text{negl} \) such that:

\[
\Pr[\text{Sig-forge}_{A, \Pi}^{1-time}(n) = 1] \leq \text{negl}(n).
\]

One-way functions: a formal introduction

The inverting experiment \( \text{Invert}_{A, f}(n) \):

1. Choose input \( x \leftarrow \{0, 1\}^n \). Compute \( y := f(x) \).
2. \( A \) is given \( 1^n \) and \( y \) as input, and outputs \( x' \).
3. The output of the experiment is defined to be 1 if and only if \( f(x') = y \).

Definition 8.72. A function \( f : \{0, 1\}^* \rightarrow \{0, 1\}^* \) is one-way if the following two conditions hold:

1. \((\text{Easy to compute:}) \) There exists a polynomial-time algorithm that on input \( x \) outputs \( f(x) \).
2. \((\text{Hard to invert:}) \) For all probabilistic polynomial-time algorithms \( A \) there exists a negligible function \( \text{negl} \) such that

\[
\Pr[\text{Invert}_{A, f}(n) = 1] \leq \text{negl}(n).
\]
Lamport scheme used to sign message the 3-bit message 011

Let \( f \) be a one-way function, that is, \( f \) is easy to compute, but difficult to invert.

**Signing \( m = 011 \):**

\[
\begin{pmatrix}
x_{1,0} & x_{2,0} & x_{3,0} \\
x_{1,1} & x_{2,1} & x_{3,1}
\end{pmatrix} \Rightarrow \alpha = (x_{1,0}, x_{2,1}, x_{3,1})
\]

Verifying for \( m = 011 \) and \( \alpha = (x_1, x_2, x_3) \):

\[
\begin{pmatrix}
y_{1,0} & y_{2,0} & y_{3,0} \\
y_{1,1} & y_{2,1} & y_{3,1}
\end{pmatrix} \Rightarrow \begin{cases}
f(x_1) = y_{1,0} \\f(x_2) = y_{2,1} \\f(x_3) = y_{3,1}
\end{cases}
\]

---

**Construction 12.15 of Lamport scheme**

Let \( f \) be a one-way function. Construct a signature scheme for messages of length \( \ell = \ell(n) \) as follows:

- **Gen:** On input \( 1^n \), proceed as follows for \( i \in \{0, \ldots, \ell\} \):
  1. Choose random \( x_{i,0}, x_{i,1} \leftarrow \{0,1\}^n \).
  2. Compute \( y_{i,0} := f(x_{i,0}) \) and \( y_{i,1} := f(x_{i,1}) \).

The public key \( pk \) and the private key \( sk \) are

\[
\begin{pmatrix}
y_{1,0} & y_{2,0} & \cdots & y_{\ell,0} \\
y_{1,1} & y_{2,1} & \cdots & y_{\ell,1}
\end{pmatrix} \quad \begin{pmatrix}
x_{1,0} & x_{2,0} & \cdots & x_{\ell,0} \\
x_{1,1} & x_{2,1} & \cdots & x_{\ell,1}
\end{pmatrix}
\]

- **Sign:** On input a private key \( sk \) and a message \( m \in \{0,1\}^\ell \) with \( m = m_1 \ldots m_\ell \), output the signature \( (x_{1,m_1}, \ldots, x_{\ell,m_\ell}) \).

- **Vrfy:** On input a public key \( pk \), a message \( m \in \{0,1\}^\ell \) with \( m = m_1 \ldots m_\ell \), and a signature \( \alpha = (x_1, \ldots, x_\ell) \), output 1 if and only if \( f(x_i) = y_{i,m_i} \) for all \( 1 \leq i \leq \ell \).
**Security of Lamport scheme**

**Theorem 12.16.** Let \( \ell \) be any polynomial. If \( f \) is a one-way function, then Construction 12.7 is a one-time signature scheme for message of length \( \ell \).

**Proof.** Let \( \Pi \) denote the Lamport scheme. Let \( \mathcal{A} \) be a PPT adversary, and define

\[
\epsilon(n) \overset{\text{def}}{=} \Pr[\text{Sig-forge}^{1\text{-time}}_{\mathcal{A},\Pi}(n) = 1].
\]

In a particular execution, let \( m' \) denote the message whose signature is requested by \( \mathcal{A} \), and let \((m, \alpha)\) denote the final output of \( \mathcal{A} \).

We say that \( \mathcal{A} \) outputs a forgery at \((i, b)\) if \( \text{Vrfy}_{\text{pk}}(m, \alpha) = 1 \) and \( m_i = b \neq m'_i \). Note that if \( \mathcal{A} \) outputs a forgery, then it outputs a forgery at some \((i, b)\).

**Algorithm I:**

This algorithm is given \( y \) and \( 1^n \) as input.

1. Choose random \( i^* \leftarrow \{1, \ldots, \ell\} \) and \( b^* \leftarrow \{0, 1\} \). Set \( y_{i,b} := y \).
2. For all \( i \in \{1, \ldots, \ell\} \) and \( b \leftarrow \{0, 1\} \) with \((i, b) \neq (i^*, b^*)\):
   - Choose \( x_{i,b} \leftarrow \{0, 1\}^n \) and set \( y_{i,b} := f(x_{i,b}) \).
3. Run \( \mathcal{A} \) on input \( \text{pk} := \left( \begin{array}{c} y_{1,0} \ y_{2,0} \ \cdots \ y_{\ell,0} \\ y_{1,1} \ y_{2,1} \ \cdots \ y_{\ell,1} \end{array} \right) \).
4. When \( \mathcal{A} \) requests a signature on the message \( m' \):
   - If \( m'_i = b^* \), stop.
   - Otherwise, return the correct signature \( \alpha = (x_{1,m'_1}, \ldots, x_{\ell,m'_\ell}) \).
5. When \( \mathcal{A} \) outputs \((m, \alpha)\) with \( \alpha = (x_1, \ldots, x_p)\):
   - If \( \mathcal{A} \) outputs a forgery at \((i^*, b^*)\), then output \( x_{i^*} \).
Analysis of \( \mathcal{A} \)'s chances

\( \mathcal{A} \)'s changes of success. When \( \mathcal{A} \) outputs a forgery at \((i^*, b^*)\), algorithm \( \mathcal{I} \) succeeds in inverting its given input \( y \). What are the chances of that happening with \( x \) is chosen at random and \( y := f(x) \)?

Thought experiment. Imagine an experiment in which \( \mathcal{I} \) is given \( x \) at the outset, sets \( x_{i^*}, b^* := x \), and then always returns a signature to \( \mathcal{A} \) in step 4.* Then the view of \( \mathcal{A} \) being run as a subroutine by \( \mathcal{I} \) is distributed identically to the view of \( \mathcal{A} \) in experiment \( \text{Sig-forge}^{1\text{-time}}(n) \) and the probability \( \mathcal{A} \) outputs a forgery is \( \epsilon \).

Now the probability the \( \mathcal{A} \) outputs a forgery at \((i^*, b^*)\), conditioned on the fact that \( \mathcal{A} \) outputs a forgery is at least \( 1/2 \ell(n) \). We conclude that the probability \( \mathcal{A} \) outputs a forgery at \((i^*, b^*)\) in our thought experiment is at least \( \epsilon/2 \ell(n) \)

*Even if \( m'_{i^*} = b^* \).

Analysis of \( \mathcal{A} \)'s chances continued

Returning to the real experiment we note that the probability that \( \mathcal{A} \) outputs a forgery at \( i^*, b^* \) is unchanged since the experiments only differ if \( \mathcal{A} \) requests a signature on a message \( m' \) with \( m'_{i^*} = b^* \). But in that case, \( \mathcal{A} \) throws up its hands.

We conclude: In the real experiment

\[
Pr[\text{Invert}_{I,f}(n) = 1] \geq \epsilon(n)/2 \ell(n).
\]

But \( f \) is one-way, so the left hand term is bounded by a negligible function, and hence so is \( \epsilon \).
At last a provably secure signature scheme*

- **Corollary 12.17.** If one-way functions exist, then for any polynomial $\ell$ there exists a one-time signature scheme for message of length $\ell$.

- Good time to stop?

*Well, provable mostly.*