Definition 7.1: A *signature scheme* is a five-tuple $(\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$, where the following conditions are satisfied:

1. $\mathcal{P}$ is a finite set of possible *messages*
2. $\mathcal{A}$ is a finite set of possible *signatures*
3. $\mathcal{K}$, the *keyspace*, is a finite set of possible *keys*
4. For each $K \in \mathcal{K}$, there is a *signing algorithm* $\mathsf{sig}_K \in \mathcal{S}$ and a corresponding *verification algorithm* $\mathsf{ver}_K \in \mathcal{V}$. Each $\mathsf{sig}_K : \mathcal{P} \to \mathcal{A}$ and $\mathsf{ver}_K : \mathcal{P} \times \mathcal{A} \to \{true, false\}$ are functions such that the following equation is satisfied for every message $x \in \mathcal{P}$ and for every signature $y \in \mathcal{A}$:
$$\mathsf{ver}(x, y) = \begin{cases} true & \text{if } y = \mathsf{sig}(x) \\ false & \text{if } y \neq \mathsf{sig}(x). \end{cases}$$

A pair (x, y) with $x \in \mathcal{P}$ and $y \in \mathcal{A}$ is called a *signed message*.

Attacks

key-only attack

Oscar possesses Alice's public key, i.e., the verification function, ver_K.

known message attack

Oscar possesses a list of messages previously signed by Alice, say

$$(x_1, y_1), (x_2, y_2), \ldots,$$

where the x_i's are messages and the y_i's are Alice's signatures on these messages (so $y_i = \text{sig}_K(x_i)$, $i = 1, 2, \ldots$).

chosen message attack

Oscar requests Alice's signatures on a list of messages. Therefore he chooses messages $x_1, x_2, \ldots,$ and Alice supplies her signatures on these messages, namely, $y_i = \text{sig}_K(x_i)$, $i = 1, 2, \ldots$.

Attack Goals

total break

The adversary is able to determine Alice's private key, i.e., the signing function sig_K. Therefore he can create valid signatures on any message.

selective forgery

With some non-negligible probability, the adversary is able to create a valid signature on a message chosen by someone else. In other words, if the adversary is given a message x, then he can determine (with some probability) the signature y such that $\mathrm{ver}_K(x, y) = \text{true}$. The message x should not be one that has previously been signed by Alice.

existential forgery

The adversary is able to create a valid signature for at least one message. In other words, the adversary can create a pair (x, y) where x is a message and $\mathrm{ver}_K(x, y) = \text{true}$. The message x should not be one that has previously been signed by Alice.

common modulus RSA exploit!

$$\begin{array}{l} y_1 = \mathrm{sig}_K(x_1) \\ y_2 = \mathrm{sig}_K(x_2) \end{array} \rightarrow \mathrm{ver}_K(x_1 x_2 \bmod n, y_1 y_2 \bmod n) = T$$

Discrete Logarithm Algorithms in Practice

1. $G = (\mathbb{Z}_p{}^*, \cdot)$, p prime, α a primitive element modulo p
2. $G = (\mathbb{Z}_p{}^*, \cdot)$, p, q prime, $p \equiv 1 \bmod q$, α an element in $\mathbb{Z}_p$ having order q
3. $G = (\mathbb{F}_{2^n}{}^*, \cdot)$, α a primitive element in $\mathbb{F}_{2^n}{}^*$
4. $G = (E, +)$, where E is an elliptic curve modulo a prime p, $\alpha \in E$ is a point having prime order $q = \#E/h$, where (typically) $h = 1, 2$ or 4
5. $G = (E, +)$, where E is an elliptic curve over a finite field $\mathbb{F}_{2^n}$, $\alpha \in E$ is a point having prime order $q = \#E/h$, where (typically) $h = 2$ or 4

FIGURE 7.1
Signing a message digest

message	x	$x \in \{0,1\}^*$
	$\downarrow$	
message digest	$z = h(x)$	$z \in \mathcal{Z}$
	$\downarrow$	
signature	$y = sig_K(z)$	$y \in \mathcal{Y}$

hash
sign
encrypt

One-time Signatures

Winternitz OTS used in IOTA

Cryptosystem 7.6: *Lamport Signature Scheme*

Let k be a positive integer and let $\mathcal{P} = \{0,1\}^k$. Suppose $f : Y \to Z$ is a one-way function, and let $\mathcal{A} = Y^k$. Let $y_{i,j} \in Y$ be chosen at random, $1 \leq i \leq k$, $j = 0, 1$, and let $z_{i,j} = f(y_{i,j})$, $1 \leq i \leq k$, $j = 0, 1$. The key K consists of the $2k$ y's and the $2k$ z's. The y's are the private key while the z's are the public key.

For $K = (y_{i,j}, z_{i,j} : 1 \leq i \leq k, j = 0, 1)$, define

$$\mathrm{sig}_K(x_1, \ldots, x_k) = (y_{1,x_1}, \ldots, y_{k,x_k}).$$

A signature $(a_1, \ldots, a_k)$ on the message $(x_1, \ldots, x_k)$ is verified as follows:

$$\mathrm{ver}_K((x_1, \ldots, x_k), (a_1, \ldots, a_k)) = \mathrm{true} \Leftrightarrow f(a_i) = z_{i,x_i}, 1 \leq i \leq k.$$

Example 7.6 7879 is prime and 3 is a primitive element in $\mathbb{Z}_{7879}{}^*$. Define

$$f(x) = 3^x \bmod 7879.$$

Suppose $k = 3$, and Alice chooses the six (secret) random numbers

$$\begin{array}{lcl}
y_{1,0} = 5831 & & z_{1,0} = 2009 \\
y_{1,1} = 735 & & z_{1,1} = 3810 \\
y_{2,0} = 803 & \overset{f}{\Longrightarrow} & z_{2,0} = 4672 \\
y_{2,1} = 2467 & & z_{2,1} = 4721 \\
y_{3,0} = 4285 & & z_{3,0} = 268 \\
y_{3,1} = 6449. & & z_{3,1} = 5731.
\end{array}$$

These z's are published. Now, suppose Alice wants to sign the message

$$x = (1, 1, 0).$$

The signature for x is

$$(y_{1,1}, y_{2,1}, y_{3,0}) = (735, 2467, 4285).$$

To verify this signature, it suffices to compute the following:

$$3^{735} \bmod 7879 = 3810$$
$$3^{2467} \bmod 7879 = 4721$$
$$3^{4285} \bmod 7879 = 268.$$

Hence, the signature is verified.

Algorithm 7.1: LAMPORT-PREIMAGE(z)

external f, LAMPORT-FORGE
comment: we assume $f : Y \rightarrow Z$ is a bijection

choose a random $i_0 \in \{1, \ldots, k\}$ and a random $j_0 \in \{0, 1\}$
construct a random public key $\mathcal{Z} = (z_{i,j} : 1 \leq i \leq k, j = 0, 1)$
  such that $z_{i_0, j_0} = z$
$((x_1, \ldots, x_k), (a_1, \ldots, a_k)) \leftarrow$ LAMPORT-FORGE($\mathcal{Z}$)
if $x_{i_0} = j_0$
  then return (a_{i_0})
  else return (fail)

If $x_{i_0} = j_0$ in the forgery,

$$f(a_{i_0}) = z_{i_0, x_{i_0}} = z_{i_0, j_0} = z,$$

THEOREM 7.1 *Suppose that $f : Y \rightarrow Z$ is a one-way bijection, and suppose there exists a deterministic algorithm,* LAMPORT-FORGE, *that will create an existential forgery for the Lamport Signature Scheme using a key-only attack, for any public key $\mathcal{Z}$ consisting of $2k$ distinct elements of Z. Then there exists an algorithm,* LAMPORT-PREIMAGE, *that will find preimages of random elements $z \in Z$ with average probability at least* $1/2$.

Cryptosystem 7.7: *Full Domain Hash*

Let k be a positive integer; let $\mathcal{F}$ be a family of trapdoor one-way permutations such that $f : \{0,1\}^k \to \{0,1\}^k$ for all $f \in \mathcal{F}$; and let $G : \{0,1\}^* \to \{0,1\}^k$ be a "random" function. Let $\mathcal{P} = \{0,1\}^*$ and $\mathcal{A} = \{0,1\}^k$, and define

$$\mathcal{K} = \{(f, f^{-1}, G) : f \in \mathcal{F}\}.$$

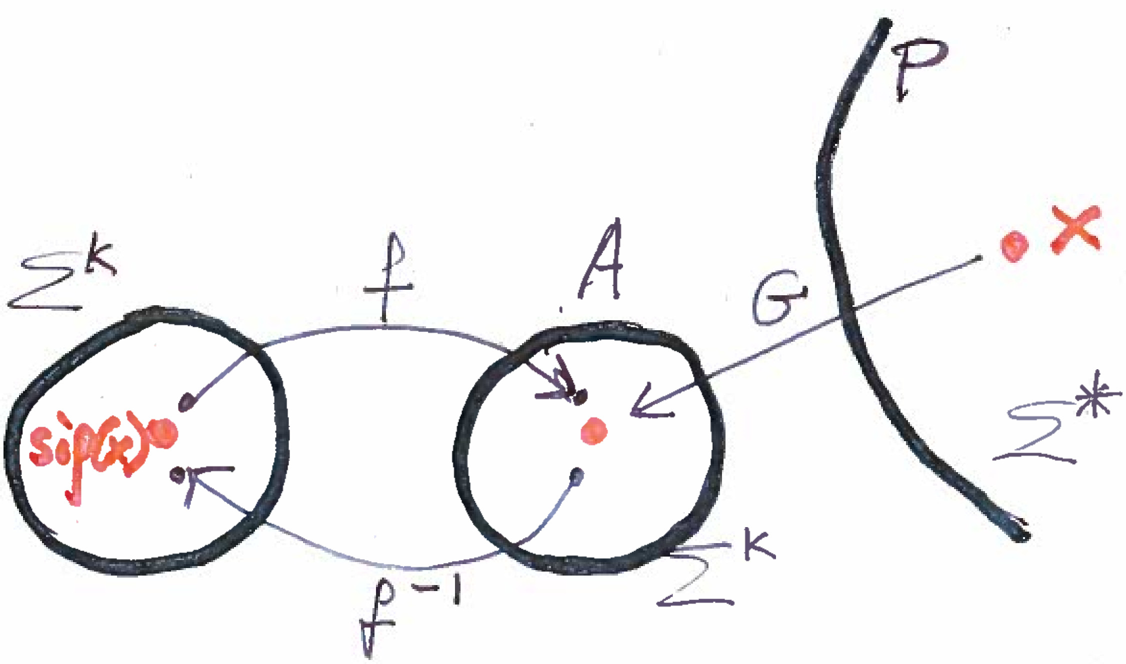
Given a key $K = (f, f^{-1}, G)$, f^{-1} is the private key and (f, G) is the public key.

For $K = (f, f^{-1}, G)$ and $x \in \{0,1\}^*$, define

$$\mathrm{sig}_K(x) = f^{-1}(G(x)).$$

A signature $y = (y_1, \ldots, y_k) \in \{0,1\}^k$ on the message x is verified as follows:

$$\mathrm{ver}_K(x, y) = \mathrm{true} \Leftrightarrow f(y) = G(x).$$



Algorithm 7.2: FDH-INVERT(z_0, q_h)

```
external f
procedure SIMG(x)
  if j > q_h
    then return ("failure")
    else if j = j_0
    then z ← z_0
    else let z ∈ {0,1}^k be chosen at random
  j ← j + 1
  return (z)

main
  choose j_0 ∈ {1, ..., q_h} at random
  j ← 1
  insert the code for FDH-FORGE(f) here
  if FDH-FORGE(f) = (x, y)
           ⎧ if f(y) = z_0
    then   ⎨   then return (y)
           ⎩   else return ("failure")
```

THEOREM 7.2 *Suppose there exists an algorithm* FDH-FORGE *that will output an existential forgery for Full Domain Hash with probability* $\epsilon > 2^{-k}$, *using a key-only attack. Then there exists an algorithm* FDH-INVERT *that will find inverses of random elements* $z_0 \in \{0, 1\}^k$ *with probability at least* $(\epsilon - 2^{-k})/q_h$.

Undeniable Signatures

Cryptosystem 7.8: *Chaum-van Antwerpen Signature Scheme*

Let $p = 2q + 1$ be a prime such that q is prime and the discrete log problem in $\mathbb{Z}_p$ is intractable. Let $\alpha \in \mathbb{Z}_p{}^*$ be an element of order q. Let $1 \leq a \leq q - 1$ and define $\beta = \alpha^a \bmod p$. Let G denote the multiplicative subgroup of $\mathbb{Z}_p{}^*$ of order q (G consists of the quadratic residues modulo p). Let $\mathcal{P} = \mathcal{A} = G$, and define

$$\mathcal{K} = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{p}\}.$$

The values p, α and β are the public key, and a is the private key.

For $K = (p, \alpha, a, \beta)$ and $x \in G$, define

$$y = \mathrm{sig}_K(x) = x^a \bmod p.$$

For $x, y \in G$, verification is done by executing the following protocol:

1. Bob chooses e_1, e_2 at random, $e_1, e_2 \in \mathbb{Z}_q^*$.
2. Bob computes $c = y^{e_1}\beta^{e_2} \bmod p$ and sends it to Alice.
3. Alice computes $d = c^{a^{-1} \bmod q} \bmod p$ and sends it to Bob.
4. Bob accepts y as a valid signature if and only if

$$d \equiv x^{e_1}\alpha^{e_2} \pmod{p}.$$

Signer <u>must</u> cooperate to verify

Disavowal of forgery $\Rightarrow$

Undeniable

Algorithm 7.3: DISAVOWAL

1. Bob chooses e_1, e_2 at random, $e_1, e_2 \in \mathbb{Z}_q$
2. Bob computes $c = y^{e_1}\beta^{e_2} \bmod p$ and sends it to Alice
3. Alice computes $d = c^{a^{-1} \bmod q} \bmod p$ and sends it to Bob
4. Bob verifies that $d \not\equiv x^{e_1}\alpha^{e_2} \pmod p$
5. Bob chooses f_1, f_2 at random, $f_1, f_2 \in \mathbb{Z}_q$
6. Bob computes $C = y^{f_1}\beta^{f_2} \bmod p$ and sends it to Alice
7. Alice computes $D = C^{a^{-1} \bmod q} \bmod p$ and sends it to Bob
8. Bob verifies that $D \not\equiv x^{f_1}\alpha^{f_2} \pmod p$
9. Bob concludes that y is a forgery if and only if

$$(d\alpha^{-e_2})^{f_1} \equiv (D\alpha^{-f_2})^{e_1} \pmod p.$$

run verify twice

THEOREM 7.3 *If $y \not\equiv x^a \pmod p$, then Bob will accept y as a valid signature for x with probability $1/q$.*

Proof:

write $c = \alpha^i$, $d = \alpha^j$, $x = \alpha^k$,

$y = \alpha^\ell$, where $i, j, k, \ell \in \mathbb{Z}_q$ and all arithmetic is modulo p.

$\forall c \ \exists q$ ways for e_1, e_2

$$c \equiv y^{e_1}\beta^{e_2} \pmod p$$
$$d \equiv x^{e_1}\alpha^{e_2} \pmod p.$$

This system is equivalent to the following system:

$$i \equiv \ell e_1 + a e_2 \pmod q$$
$$j \equiv k e_1 + e_2 \pmod q.$$

Now, we are assuming that

$$y \not\equiv x^a \pmod p,$$

so it follows that

$$\ell \not\equiv ak \pmod q.$$

$\det\begin{bmatrix} \ell & a \\ k & 1 \end{bmatrix} \neq 0$

so d solves to (e_1, e_2)

Alice can cheat with $1/q$ chance $\square$