Summary

One Time $\ell$-bit Signatures construction from any OWF (end proof of security). Collision Resistant-Hash Functions definition. Birthday attack.

1 One Time $\ell$-bit Signatures

Construction 1 $\textsc{Gen}(1^k)$: Choose $x_0^1, x_1^1, x_0^2, x_1^2, \ldots, x_0^\ell, x_1^\ell$ : $\ell$-pairs of $\kappa$-bit integers chosen uniformly at random from $\{0,1\}^\kappa$. Set $y_0^i = f(x_0^i)$, $y_1^i = f(x_1^i)$, for $i = 1,2,\ldots,\ell$.

$\textsc{Output}$: $sk = all$ $x$'s and $pk = all$ $y$'s.

$\textsc{Sign}_{sk}(m)$: where $m \in \{0,1\}^\ell$ if $m = m_1,\ldots,m_\ell : m_i \in \{0,1\}$ set $\sigma_i = x_{m_i}^i$, for $i \in \{1,\ldots,\ell\}$ and output $\sigma = (\sigma_1,\ldots,\sigma_\ell) : \sigma_i \in \{0,1\}^\kappa$

e.g. $m = 110$
$\sigma = x_1^1 x_1^2 x_0^3$

$\textsc{Verify}_{pk}(m, \sigma)$ : check that $f(\sigma_i) = y_{m_i}^i$

Theorem 1 If $f$ is a OWF then this is a secure one-time signature scheme

Proof: Suppose $\exists$ ppt $A$ that forges with non negligible probability $\delta(k)$. Construct $B$ that inverts $f$ with non-negligible probability.

$B(y)$:
1) choose $j \in \{1,\ldots,\ell\}$ at random $c \in \{0,1\}$ at random and set $y_c^j = y$
2) for all \((i,b) \neq (j,c)\) choose \(x_b^i \in \{0,1\}^k\) at random and set \(y_b^i = f(x_b^i)\)

3) Run \(A(pk)\) where \(pk = \) all \(y\)’s.

4) When (if) \(A\) asks for a signature on some \(m = m_i, \ldots, m_t \in \{0,1\}^\ell\) then:

if \(m_j = c\) output \textbf{fail}

else \(B\) provides \(A\) with a \textit{correct signature} on \(m\). (note \(A\) knows inverses of all \(y\)’s except for \(y_c^i = y\))

if \(A\) outputs a valid \((m', \sigma')\) where \(m \neq m'\) and if \(m_j' = c\) then output \(\sigma_j'\) (note this is the inverse of \(y\)) else output \textbf{fail}.

Intuition: \(B\) will be able to forge all signatures except those where bit \(j = c\). Now \(A\) asks for a signature on msg \(m\) and forges a signature on message \(m' \neq m\). \(m'\) and \(m\) differ on at least one location. If \(m_j' = c\) and \(m_j \neq c\), then \(B\) will succeed in inverting \(f\).

\textbf{Example:}

Assume \(B\) chooses its unknown \(y_c^j\) to be \(y_0^2\). So \(B(y)\) runs \(A\) on \((y_0^1, y_1^1, y, y_1^2, y_0^3, y_1^3)\) and \(B\) knows the corresponding inverses \((x_0^1, x_1^1, ?, x_1^2, x_0^3, x_1^3)\) and wants to find the inverse of \(y\).

If \(A\) asks for a signature for \(m = 110\) then \(B\) can provide \(A\) with a signature \(\sigma = x_1^1x_1^2x_0^3\). Otherwise, for example if \(m = 100, B\) cannot provide \(A\) the signature it wants to continue working, so it fails.

\(B\) hopes that if \(A\) asks for a signature that it can provide (namely with second bit \(\neq 0\)). \(B\) further hopes that \(A\) will output a different message with a valid signature that will differ from \(m\)’s signature in the position \(x_0^2\). Then \(B\) will know that this \(x_0^2\) is the inverse of \(y_0^2\) \((y_c^2)\).

What is the probability that \(B\) succeeds inverting \(f\)?

Note \(pk\) on which \(A\) is ran is distributed exactly the same as \(pk\) generated by \textbf{GEN} (independently of \(j,c\)).

Thus, \(B\) succeeds with probability \(\geq \frac{\delta(c)}{2^t}\) (since a good \((c,j)\) was chosen with probability at least \(\frac{1}{2^t}\) which is \textbf{non-negligible}.

\[\blacksquare\]

\textbf{PROBLEMS:}
1) for length $\ell$ messages we need keys of size $2\ell \kappa$ (not very efficient)
2) Still only secure for one time signature

We will define CRHF which will help us fix both of these problems.

## 2 Collision Resistant Hash Functions

**Requirements:** (intuitive)

1) $h(x) \ll x$
2) $h$ easy to evaluate
3) hard to find collisions i.e. $x \neq x'\ h(x) = h(x')$.

We want CRHF for which we can securely use the Hash and Sign paradigm: to sign long message $m$, first hash it, then sign.

**Definition 1** $H = \{ h_i : D_i \rightarrow R_i \}_{i \in I}$ is a family of collision free hash functions if:

1) $|R_i| < |D_i|$ (hashing)
2) there is a ppt GEN such that: $GEN(1^n)$ outputs $i \in I$
3) (easy to evaluate) Given $x$, i easy to compute $h_i(x)$ in poly time
4) (hard to find collisions) $\forall$ ppt $A \exists$ a negligible function $\varepsilon$ such that:

$$\text{Prob}[ A(1^n,i) = (x,x') \text{ such that } x \neq x' \text{ and } h(x) = h(x')] \leq \varepsilon(\kappa).$$

Probability taken over $i \leftarrow GEN(1^n)$ and coin tosses of $A$.

**Note:** if $h$ has range $\{0,1\}^n$

**Exponential time attacks**

1) Exhaustive search attack: check $h_i(x)$ for all inputs until you find collision. For general $h$ this may take $\sim O(2^k)$ evaluations of $h$.

2) Birthday attack: Choose at random $x$’s until you get a collision. If $t$ messages are chosen, there are $\binom{t}{2} \approx \frac{t^2}{2}$ pairs. For each pair the probability of collision is roughly $\frac{1}{2^k}$ so overall expected number of collisions $\approx \frac{t^2}{2^{k+1}}$. Taking $t = \Theta(2^{\frac{k}{2}})$ (which is a square root of exhaustive search) gives collision with high probability.
This is known as birthday attack, after the known ”birthday paradox”: among t people the expected number of people who share birthday with someone else (”collision”) is \( \approx \frac{t^2}{365} \) (assuming uniform distribution). When the number of people is around \( \sqrt{365} \approx 20 \) (the expected number of collisions = 1,5), with probability \( \approx \frac{1}{2} \), 2 people will have the same birthday (to be exact 23 is the number of people that the probability of collision is > \( \frac{1}{2} \)). If you have 30 people, the probability that two have the same birthday is extremely high.