Collision Resistant Hash Functions (CRHF)
One-Time signatures
Fiat-Shamir heuristics
Zero Knowledge (ZK)

1 CRHF

We want to find a function $H$ where it is hard to find $m \neq m'$ such that $H(m) = H(m')$.

First, recall the properties of a $\Sigma$-protocol:

- **Special soundness**: Given valid $(a, c, z, c', z')$ with $c \neq c'$, we can extract a witness $x'$ such that $R(x', y) = 1$
- **Special HVZK**: $\exists \text{ sim}^*(c)$ that gives $(a, z)$

For CRHF, we will need a slightly stronger version of special HVZK:

- **Strong Special HVZK**: $\exists \text{ sim}^{**}(c)$ such that $z$ is chosen at random and $a = f(c, z, y)$.

Example: for Schnorr's protocol, $a = g^z y^c \text{ mod}(p)$.

In most $\Sigma$-protocols, $|c| + |z| > |a|$.

**Construction** of $H_{PK}$:

1. $Gen(1^\lambda) : (x, y)$ taken from relation $R$ and set $PK = y$
2. $H_y(c, z) = f(c, z, y)$

**Lemma 1** If $|c| + |z| > |a|$ and $\Sigma$ is a $\Sigma$-protocol for a hard relation with special soundness, strong special HVZK and the extra property that given $(a, c)$, $\exists$ a unique accepting $z$, then the above $H$ is a CRHF.

**Proof**: If $H(c, z) = H(c', z') = a$ and $(c, z) \neq (c', z')$ then $c \neq c'$ by assumption. By special soundness, we can then extract $x$, contradicting the hard relation.

Example: for Schnorr, $H(c, z) = g^z y^c \text{ mod}(p)$. If $g \rightarrow g_1, y \rightarrow g_2, z \rightarrow z_1$ and $y \rightarrow z_2$ then we recover the more familiar $H = g_1^{z_1} g_2^{z_2} \text{ mod}(p)$. 

L5-1
2 One-Time Signatures

One-time signatures \((Gen, Sig, Ver)\) are like regular signatures, except we have only one query to the signing oracle (i.e., it is only safe to sign one message). In theory, they are easier and faster than full signatures and using the “top down tree,” we can build full signatures from one-time signatures.

We will need a \(\Sigma\)-protocol for an SPR function \(f : \{0,1\}^n \to \{0,1\}^k\). Recall, an SPR function is one where given a random \(x\), it is hard to find \(x \neq x'\) such that \(f(x) = f(x')\).

**Construction:**

\(Gen(1^\lambda)\):
- \(x \leftarrow \{0,1\}^n\), let \(y = f(x)\)
- pick \(r\) at random and let \(a = \text{first-flow}(x, r)\)
- \(Pk = (a, y), Sk = (x, r)\)

\(Sig(c)\):
- Compute \(z = \text{third-flow}(x, r, c)\)
- Output \(z\).

**When is it secure?**

If \(F\) asked \(Sig(c) \rightarrow z\) and forged \((z' = Sig(c'), c \neq c')\) then we have \((a, c, c', z, z')\) needed by special soundess to extract \(x'\) such that \(f(x') = y = f(x)\), which would imply \(x = x'\).

The point is, \(z\) is independent if \(x\) by WI. Using the proof from lecture 1 we get:

**Theorem 2** If \(f\) is SPR, then the above is a one-time signature if either

- \(f\) is regular (and \(n > k\), or
- \(n > k + \omega(\log \lambda)\)

Exercise: Write the one-time signature for Okamoto.

Question: How secure/insecure is it for Schnorr and GQ?

2.1 Leakage-resilience?

Say the forger can learn \(\ell\) bit of \(x\).

**Theorem 3** If \(f\) is SPR and \(\Sigma\) has strong special HVZK, then the above is leakage-resilient if either

- \(f\) is regular and \(n > k + \ell\), or
• $n > k + \ell + \omega(\log \lambda)$

**Observation 4** We need to argue that $z = \text{Sig}(c)$ doesn’t give more information about $x$ beyond the leakage.

## 3 Fiat-Shamir Signatures

Given the transcript

<table>
<thead>
<tr>
<th>Prover</th>
<th>Verifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td></td>
</tr>
<tr>
<td>$c$</td>
<td></td>
</tr>
<tr>
<td>$z$</td>
<td></td>
</tr>
</tbody>
</table>

instead of a random $c$, set $c = H(a,m)$. We can then use $c$ to create a signature $\text{Sig}(m) = (a,z)$.

More generally, we can build a signature scheme from any three-round public-coin passive ID scheme (e.g. $\Sigma$-protocols):

**Gen($1^\lambda$):**
- $(x,y) \leftarrow R(1^\lambda)$
- $SK = x$
- $PK = (y,H)$

**Sig($m$):**
- produce $a$ honestly using $x$
- set $c = H(a,m)$
- produce $z$ honestly for this $c$
- output $\sigma = (a,z)$

**Ver($m,(a,z)$):**
- $c = H(a,m)$
- check $(a,c,z)$ is valid.
Example: Schnorr’s signature
\[
a = g^r, \ c = H(a, m)
\]
Signer: \( z = r + xc, \ \sigma = (a, z) \)
Verifier: \( c \leftarrow H(a, m), \ g^z = ay^c \)

**What is H?**
Practice: \( H = \text{SHA2} \)
Theory: \( H \) is random oracle. \( H : \{0, 1\}^* \rightarrow \{0, 1\}^t \) (assume \( \{0, 1\}^t = \text{challenge space} \))

Random Oracle Model (ROM): Assume \( H \) above is a random function.

**Theorem 5** Assume \((P \leftrightarrow V)\) is a 3-round public-coin passively secure ID scheme where
1. \( |c| = t = \omega(\log \lambda) \)
2. \( H_{\infty}(a|y) = \omega(\log \lambda) \)

then Fiat-Shamir signatures from \((P \leftrightarrow V)\) is secure (UF-CMA) in ROM.

**Proof:** Assume \( \exists \) Forger \( F^{Sig(H)} \) forging \((m^*, (a^*, z^*))\) with \( \Pr > \epsilon \). Construct prover \((P^*)^{Tran(\cdot)}\) who breaks ID scheme.
Assume \( F \) makes \( q_s \) signing queries \( \{m_i|i = 1, \ldots, q_s\} \) and \( q_h \) hash queries \( \{H(\hat{a}_i, \hat{m}_i)|i = 1, \ldots, q_h\} \).
WLOG, assume \( F \) calls \( H(a^*, m^*) \), so \((a^*, m^*) = (\hat{a}_{i^*}, \hat{m}_{i^*})\) for some \( i^* \).

\((P^*)^{Tran(\cdot)} : j \leftarrow \{1, \ldots, q_h\} \) (hope \( j = i^* \))
get \((a_1, c_1, z_1), \ldots, (a_{q_h}, c_{q_h}, z_{q_h})\) from \( Tran(\cdot) \)

Simulate \( F \) until \( H(\hat{a}_j, \hat{m}_j) \) as follows:
1. to answer \( H(\hat{a}_i, \hat{m}_j) \):
   - if defined, give it
   - else, define it as random and give it
2. to answer \( Sig(m_i) \):
   - define \( H(a_i, c_i) = c_i \)
   - (problem: if already defined \( \Rightarrow \) fail)
   - output \((a_i, z_i)\) as \( Sig(m_i) \)
     By assumption on \( H_{\infty}(a|y) \), \( \Pr(\text{fail}) = \text{negl.} \)
3. to answer \( H(\hat{a}_j, \hat{m}_j) \):
   - give \( \hat{a}_j \) to \( V \). \( V \) returns random \( c^* \). Define \( H(\hat{a}_j, \hat{m}_j) = c^* \)
4. answer remaining \( H \) queries as in (1).
Finally, if $F$ forges $(m^*, (a^*, z^*))$, fail if $(a^*, m^*) \neq (\hat{a}_j, \hat{m}_j)$. Otherwise, send $z^*$ to $V$.

\[
\Pr[P^* \text{succeeds}] \geq \Pr[F \text{succeeds and } j = i^*] - \Pr[\text{failure in step (2)}] \\
\geq \frac{\epsilon}{q_h} - \text{negl}
\]

\[\square\]

**Corollary 6** All natural $\Sigma$-protocols give secure signature schemes in ROM.

Can we instantiate $H$ in standard model? Evidence suggests it’s hard to prove from “nice” assumption even if it’s true.

**Observation 7** $\exists$ artificial 3-round public-coin passively secure ID scheme such that Fiat-Shamir heuristics gives insecure signatures $\forall H$

## 4 Zero Knowledge (ZK)

Recall, $\Sigma$-protocols are only HVZK: $\exists \text{sim}(y) \rightarrow (a, c, z) \equiv \text{real} (a, c, z)$

What about malicious $V^*$?

**Definition 8** ($P(x) \leftrightarrow V(y)$) is a (black-box, auxiliary input) ZK protocol with soundness $s$ for a polynomial time relation $R(x, y)$ (defines $L = \{y|\exists x \text{ such that } R(x, y) = 1\}$) if

- **completeness**: If $R(x, y) = 1$, then $\Pr(P(x) \leftrightarrow V(y) \text{ accepts}) = 1$

- **soundness**: If $y \notin L(R(x', y) = 0 \forall x')$, then $\forall P^*$ (even exponential time), $\Pr(P^*(y) \leftrightarrow V(y) \text{ accepts}) \leq s$

- **ZK**: $\exists$ efficient simulator $S$ such that $\forall (x, y) \in R, \forall \text{aux}$, and $\forall$ efficient $V^*$, we have $\text{View}_{V^*}(P(x) \leftrightarrow V^*(y, \text{aux})) \approx S^{V^*}(y, \text{aux})$