G22.3220-001/G63.2180 Advanced Cryptography September 23, 2009

Lecture 3

Lecturer: Yevgeniy Dodis *Scribe: Adriana Lopez*

Last time we covered Schnorr's ID scheme:

- example of Σ-protocol
- passively secure
- 2 proofs of proof of knowledge
  - "easy": $\Pr[\text{EXT succeeds}] \geq \epsilon^2 - \frac{\epsilon^2}{q}$, time $= 2$
  - "hard": $\Pr[\text{EXT succeeds}] \geq \frac{1}{2}$, time $= O(\frac{1}{\epsilon})$

Today we'll cover:

- a review of Σ-protocols and their properties
- the Guillou-Quisquater protocol and proof that it is a Σ-protocol
- Witness Indistinguishability
- Construction of passively secure ID schemes from SPR functions
- the Okamoto protocol

1 Σ-protocols

Definition 1 *A* Σ-protocol *for language* $L = \{y \mid \exists x \text{ such that } R(x, y) = 1\}$ *(i.e. all* y *such that there exists a witness* x*), is a 3-round "*(a, c, z)*"-protocol where* c *is chosen at random from some challenge space* S *(where* $|S| = q$*) and which has* "special soundness" *and* "special HVZK" *properties.*

Properties:

- Special HVZK: $\exists$ efficient SIM* such that $\forall c \in S$, $(a^*, z^*) \leftarrow \text{SIM}^*(c)$:

$$(a^*, z^*) \equiv (a, z) \text{ in real } P(x) \text{ transcript for challenge } c$$

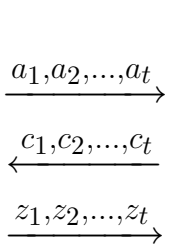
- Strong Special HVZK: Same as Special HVZK except that simulator is required to sample z uniformly at random: $\exists$ efficient SIM** such that $\forall c \in S$, samples z^{**} uniformly at random, computes a^{**} from c and z^{**} and:

$$(a^{**}, z^{**}) \equiv (a, z) \text{ in real } P(x) \text{ transcript for challenge } c$$

- Special Soundness: For all valid transcripts (a, c, z), (a, c', z'), $\exists$ EXT* such that for $c \neq c'$, $x' \leftarrow \text{EXT}^*(a, c, z, c', z')$: $R(x', y) = 1$.
- Proof of Knowledge: If P^* succeeds with probability $\geq \epsilon$, then $\exists$ EXT such that for $x' \leftarrow \text{EXT}^{P^*}$, $R(x', y) = 1$ with probability $\geq poly(\epsilon)$.

Lemma 2 *Σ-protocols satisfy the proof of knowledge property if $q = |S|$ is "large".*

This motivates the design of Σ-protocols with large q. The good news is that we can run Σ-protocols in parallel to amplify q.



Lemma 3 *The t-wise direct product of Σ-protocols is a Σ-protocol with challenge space $S^t (|S^t| = q^t)$.*

Proof: EXERCISE. Hint: Run SIM*, SIM**, EXT, EXT*, t times in parallel. □

Observation 4 *In Schnorr, S is already large so there is no need to amplify q.*

Remark: We can also define *computational Σ-protocols* by making the following changes:

- In special HVZK, replace $\equiv$ (perfect indistinguishability) by $\approx$ (computational indistinguishability).
- In special soundness, replace "$\forall$ (a, c, z, c', z') EXT succeeds" by "it is hard to find (a, c, z, c', z') for which EXT fails".

Later we will prove that if OWFs exist, then for all NP languages there exists a computational Σ-protocol .

Theorem 5 *If L is a hard relation (i.e. for $(x, y) \leftarrow Gen(1^\lambda)$, given y it is hard to find x) and L has a Σ-protocol with large q, then this Σ-protocol gives a passively secure ID scheme.*

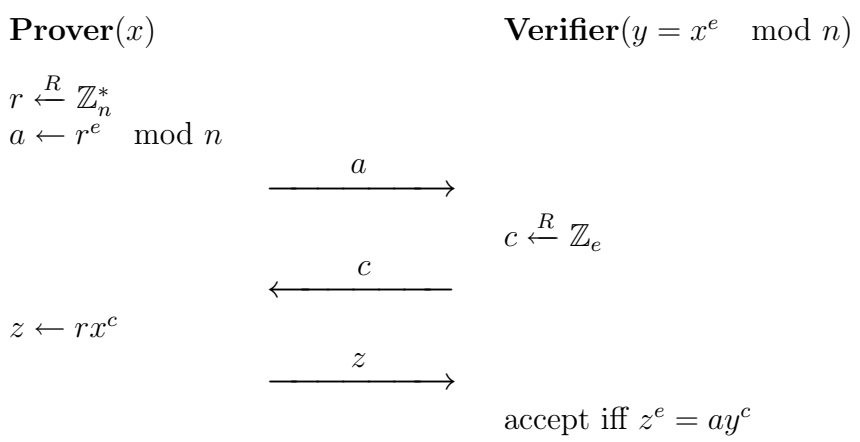
Notice that if f is a OWF, then $L_f = \{y \mid \exists x \text{ such that } y = f(x)\}$ is a hard relation.

Corollary 6 *A Σ-protocol for a OWF gives a passively secure ID scheme. In particular, if OWFs exist then there exists passively secure ID schemes.*

The latter part follows from the (unproven) claim we made earlier that if OWFs exist, then any NP statement has a (computational) Σ-protocol, and the language L_f above forms such an NP statement.

2 Guillou-Quisquater Protocol

The Guillou-Quisquater Protocol considers the (RSA) function $f(x) = x^e \mod n$ where e is a large prime and $\gcd(e, \varphi(n)) = 1$.



Theorem 7 *The Guillou-Quisquater Protocol described above is a Σ-protocol.*

Proof:

- **Correctness**: Note that $z^e = (rx^c)^e = r^e x^{ec} = r^e (x^e)^c = ay^c$.
- **Strong Special HVZK:** Define $\text{SIM}^*(c)$:

  1. $z \xleftarrow{R} \mathbb{Z}_n^*$
  2. $a \leftarrow \frac{z^e}{y^c} \mod n$

  The distribution of z is uniform over $\mathbb{Z}_n^*$ in both the real transcript and when output by SIM^*. And in both cases, $a = \frac{z^e}{y^c} \mod n$.

- **Special Soundness:** To prove special soundness we will use Shamir's trick.

  Theorem 8 (Shamir's Trick) *Assume $g^\alpha = y^\beta \mod n$, for $\alpha, \beta > 0$. Let $\gamma = \gcd(\alpha, \beta)$. There exists an efficient algorithm that computes the αth root on h^γ, i.e. there exists an efficient algorithm that computes f such that $f^\alpha = y^\gamma \mod n$.*

  Proof: Using the extended Euclidean algorithm, we can find $u, v \in \mathbb{Z}$ such that $\alpha u + \beta v = \gamma$. This means that $g^{\alpha v} = y^{\beta v} = y^{\gamma - \alpha u}$ and $y^\gamma = g^{\alpha v} y^{\alpha u} = (g^v y^u)^\alpha$. Thus, $f = g^v y^u$. □

  To prove special soundness, construct $\text{SIM}^{**}(a, c, z, c', z')$. Since (a, c, z) and (a, c', z') are valid transcripts, we have that $z^e = ay^c$ and $(z')^e = ay^{c'}$. Therefore,

$$\left(\frac{z}{z'}\right)^e = y^{c-c'}$$

Using Shamir's trick, SIM** computes f such that $f^e = y^{\gcd(e,c-c')}$. Notice that since e is prime, $c \neq c'$ and $c - c' < e$ (since $c, c' \in \mathbb{Z}_e$), we have that $y^{\gcd(e,c-c')} = y$. Thus, SIM** computes f such that $f^e = y \mod n$.

□

3 Leakage-Resilient Passive ID Schemes

Recall that in the leakage-resiliance model, Eve can learn ℓ bits of information about x. We can then easily generalize Corollary 6 to the leakage-resilient case as follows.

Lemma 9 *If f is an ℓ-LR-OWF and Σ is a Σ-protocol for L_f, then Σ is an ℓ-LR ID scheme.*

Recall that if f is SPR (second pre-image resistant), then f is a LR-OWF. This means that we can get passively secure LR-ID schemes from Σ-protocols for such SPR functions. As it turns out, the resulting protocols are in fact *actively secure* (and leakage-resilient). To see why, we first need to introduce the concept of witness indistinguishability (WI).

4 Witness Indistinguishability (WI)

Definition 10 *An interactive protocol $(P(x), V(y))$ is* witness indistinguishable *for relation R if $\forall(y, x, x')$ such that $R(x, y) = R(x', y) = 1$ and for any malicious verifier V^*, the view of $P(x) \leftrightarrow V^*(y)$ is perfectly indistinguishable ($\equiv$) from the view of $P(x') \leftrightarrow V^*(y)$. We also define* computational witness indistinguishability *by requiring the views to be computationally indistinguishable ($\approx$) instead of perfectly indistinguishable ($\equiv$).*

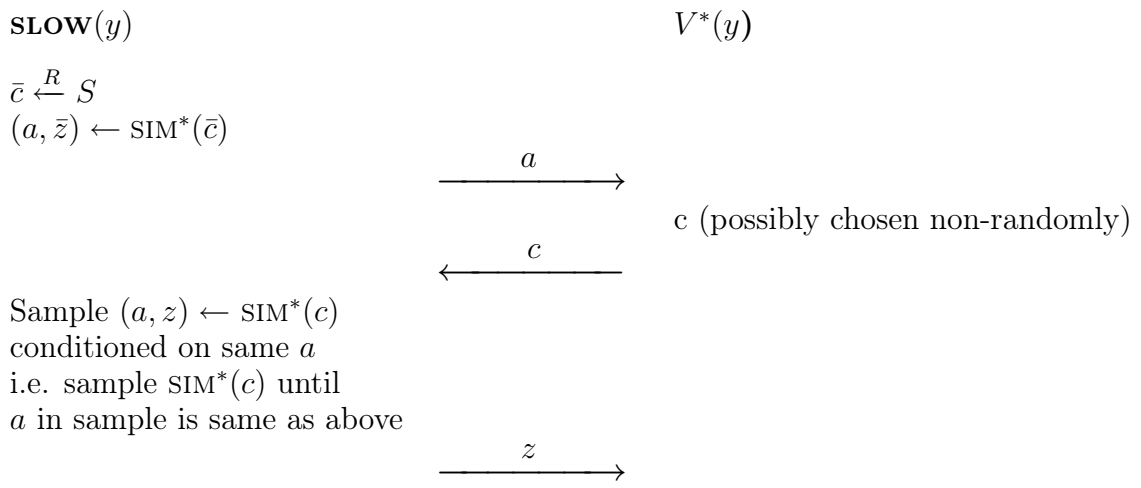
Lemma 11 *Σ-protocols are witness indistinguishable.*

Proof: By perfect HVZK, we know there exists SIM* such that $\forall c, (a^*, z^*) \leftarrow \text{SIM}^*(c)$: $(a^*, z^*) \equiv (a, z)$ from real transcript with challenge c. We claim that for all V^*, there exists an exponential-time algorithm SLOW(y) such that $\forall x$ such that $R(x, y) = 1$:

$$[\text{SLOW}(y) \leftrightarrow V^*(y)] \equiv [P(x) \leftrightarrow V^*(y)]$$

Notice that the left hand side is independent of x and that the statement holds $\forall x$ such that $R(x, y) = 1$. This means that for x, x' such that $R(x, y) = R(x', y) = 1$, we have $[P(x) \leftrightarrow V^*(y)] \equiv [P(x') \leftrightarrow V^*(y)]$, which is exactly what we want to prove.

We show the construction of SLOW(y):



We note that SLOW terminates because since the honest verifier could choose to send c with non-zero probability $\frac{1}{q}$, there must exist z such that $(a, z) \leftarrow \text{SIM}^*(c)$. This is because sampled z must be distributed in the same way as in real transcript with P, and in particular is independent of x (from perfect HVZK). Algorithm SLOW terminates but it is not efficient since the sampling step can take exponential time. But notice that the running time of SLOW is not important, since it is only an aid to prove that $[P(x) \leftrightarrow V^*(y)] \equiv [P(x') \leftrightarrow V^*(y)]$. □

Observation 12 *Witness indistinguishability is only interesting if there exists more than one witness. Otherwise, WI follows directly from HVZK.*

5 Actively Secure ID Schemes from SPR functions

Recall that a function $f : \{0,1\}^n \to \{0,1\}^k$ with $n > k$ is called *second pre-image resistant (SPR)* if $\forall x \in \{0,1\}^n$, it is hard to find $x' \neq x$ such that $f(x) = f(x')$.

Theorem 13 *Assume Σ is a Σ-protocol for SPR function $f : \{0,1\}^n \to \{0,1\}^k$. Then Σ is a actively secure ID scheme if either of these conditions hold:*

- *f is regular (i.e. f is such that every $y \in \{0,1\}^k$ has the same number of pre-images in $\{0,1\}^n$) and $n > k$ as required for an SPR function.*
- *$n > k + \omega(\log \lambda)$.*

Proof: We assume there exists algorithm A that breaks the ID scheme, and construct B (running A on $f(x)$) that breaks the SPR property of f.

Case 1: f is a regular function.
In learning stage, $B(x)$ runs $[P(x) \leftrightarrow A(y)]$. By witness indistinguishability, A doesn't know anything more about x aside from $y = f(x)$. In impersonation stage, B uses the algorithm

EXT from the proof of knowledge property to extract a valid witness x' from A. If $x' = \perp$ (i.e. if EXT fails), then repeat stages; otherwise output x'.

We construct B' and show that $\Pr[B \text{ succeeds}] = \Pr[B' \text{ succeeds}] = \delta\left(1 - \frac{2^k}{2^n}\right) \geq \frac{\delta}{2}$, where δ is the probability that EXT succeeds in extracting witness x'. B' selects y at random from $\{0,1\}^k$ and runs $[\text{SLOW}(y) \leftrightarrow A(y)]$ in learning stage. In impersonation stage, B uses the algorithm EXT to extract a valid witness x' from A. If $x' = \perp$ (i.e. if EXT fails), then repeat stages; otherwise, once $x' \neq \perp$, B chooses x uniformly at random from the pre-images of y. The following illustrates the constructions of B and B'.

B :	B' :
repeat until $x' \neq \perp$:	repeat until $x' \neq \perp$:
	$y \xleftarrow{R} \{0,1\}^k$
$P(x) \leftrightarrow A(y)$	$\text{SLOW}(y) \leftrightarrow A(y)$
$x' \leftarrow [A(y) \leftrightarrow \text{EXT}(y)]$	$x' \leftarrow [A(y) \leftrightarrow \text{EXT}(y)]$
	$x \xleftarrow{R} f^{-1}(y)$
B succeeds iff $x \neq x'$	B' succeeds iff $x \neq x'$

We have that:

$$\begin{aligned} \Pr[B' \text{succeeds}] &= \Pr[\text{EXT succeeds and } x' \neq x] \\ &= \Pr[\text{EXT succeeds}] \Pr[x' \neq x | \text{EXT succeeds}] \\ &= \delta\left(1 - \frac{1}{2^{n-k}}\right) = \delta\left(1 - \frac{2^k}{2^n}\right) \end{aligned}$$

This is because x' is already fixed when B' chooses x uniformly at random from the set of 2^{n-k} pre-images of y.

Case 2: for general f, assuming $n > k + \omega(\log lambda)$.
The above argument does not work for the case when f is not regular, since y could have a very small number of pre-images. We will thus change the construction of B and B'. B is constructed as in Case 1 with one difference: B does not repeat stages if $x' = \perp$. B only runs EXT once and returns the output x' whether EXT is successful or not. The same applies to B'. . Below are the constructions of B and B':

B :	B' :
	$y \xleftarrow{R}$ right distribution
$P(x) \leftrightarrow A(y)$	$\text{SLOW}(y) \leftrightarrow A(y)$
$x' \leftarrow [A(y) \leftrightarrow \text{EXT}(y)]$	$x' \leftarrow [A(y) \leftrightarrow \text{EXT}(y)]$
	$x \xleftarrow{R} f^{-1}(y)$
B succeeds iff $x \neq x'$	B' succeeds iff $x \neq x'$

We have that:

$$
\begin{aligned}
\Pr\left[B' \text{succeeds}\right] &= \Pr\left[\text{EXT succeeds and } x' \neq x\right] \\
&= \Pr\left[\text{EXT succeeds}\right] - \Pr\left[\text{EXT succeeds and } x' = x\right] \\
&\geq \delta - 2^{-H_\infty(X|Y)} \\
&\geq \delta - 2^{-(n-k)} \qquad \text{(by lemma from lecture 1)} \\
&= \delta - \frac{1}{2^{\omega(\log \lambda)}} \\
&= \delta - \mathsf{negl}(\lambda)
\end{aligned}
$$

□

One question is whether we can have the same result in the leakage resilient model, where Eve can learn ℓ bits. The answer is yes, as stated by the theorem below:

Theorem 14 *Assume Σ is a Σ-protocol for a ℓ-LR function $f : \{0,1\}^n \to \{0,1\}^k$. Then Σ is a actively secure ID scheme if either of these conditions hold:*

- *f is regular and $n > k + \ell$.*
- *$n > k + \ell + \omega(\log \lambda)$.*

6 Okamoto's Protocol

All that remains is to give some examples of Σ-ptrotocols for some natural SPR functions. Okamoto's protocol provides such an example. It uses the function $f(x_1, x_2) = g_1^{x_1} g_2^{x_2}$, where g_1, g_2 are generators of some group G. The Σ-protocol follows.

Prover(x_1, x_2)		**Verifier**$(y = g_1^{x_1} g_2^{x_2})$
$r_1, r_2 \xleftarrow{R} \mathbb{Z}_p^*$		
$a \leftarrow g_1^{r_1} g_2^{r_2}$		
	$\xrightarrow{a}$	
		$c \xleftarrow{R} \mathbb{Z}_p$
	$\xleftarrow{c}$	
$z_1 \leftarrow r_1 + cx_1$		
$z_2 \leftarrow r_2 + cx_2$		
$z \leftarrow (z_1, z_2)$		
	$\xrightarrow{z}$	
		accept iff $g_1^{z_1} g_2^{z_2} = a y^c$

Lemma 15 *Okamoto's protocol is a Σ-protocol for $L_{\text{SPR}} = \{y \mid \exists (x_1, x_2)$ such that $y = g_1^{x_1} g_2^{x_2}\}$.*

Proof:

- **Correctness:** Note that $g_1^{z_1} g_2^{z_2} = g_1^{r_1+cx_1} g_2^{r_2+cx_2} = g_1^{r_1} g_1^{cx_1} g_2^{r_2} g_2^{cx_2} = (g_1^{r_1} g_2^{r_2})(g_1^{x_1} g_2^{x_2})^c = ay^c$.

- **Strong Special HVZK:** Construct SIM*:

  1. $z_1, z_2 \xleftarrow{R} \mathbb{Z}_p$
  2. $a \leftarrow \frac{g_1^{z_1} g_2^{z_2}}{y^c}$

  The distribution of $z = (z_1, z_2)$ is uniform over $\mathbb{Z}_p \times \mathbb{Z}_p$ in both the real transcript and when output by SIM*. And in both cases, $a = \frac{g_1^{z_1} g_2^{z_2}}{y^c}$.

- **Special Soundness:** Construct SIM$^{**}(a, c, z, c', z')$. Since (a, c, z) and (a, c', z') are valid transcripts, we have that $g_1^{z_1} g_2^{z_2} = ay^c$ and $g_1^{z_1'} g_2^{z_2'} = ay^{c'}$. Therefore,

$$y^{c-c'} = g_1^{z_1 - z_1'} g_2^{z_2 - z_2'}$$

$$y = g_1^{\frac{z_1 - z_1'}{c-c'}} g_2^{\frac{z_2 - z_2'}{c-c'}}$$

  Therefore, $\left(\frac{z_1 - z_1'}{c-c'}, \frac{z_2 - z_2'}{c-c'}\right)$ is a witness and SIM** can compute it by performing simple group operations.

□

In particular, since f above is regular, the Okamoto's protocol is ℓ-LR for $\ell = n-k-1 \approx n/2$.

Remark: There is a Okamoto-GQ variant that uses the function: $f(x, \alpha) = x^e g^\alpha \mod n$, where g is a random element of $\mathbb{Z}_n^*$.