CS 7880 Graduate Cryptography November 17, 2015

Lecture 18: Identification schemes

Lecturer: Daniel Wichs *Scribe: Hridam Basu*

1 Topic Covered

- Identification schemes

2 Identification schemes

We are going to construct identification schemes and see how to build digital signatures from them. Identification schemes helps to convince others about the identity of a particular person without others(read adversaries) being able to convince others about the identity of that particular person. For Eg:- I want to log in to many websites using the same password. But a secure identification scheme should prevent other websites from loging in to a particular website using my identity.
An identification scheme consists of three parts :-

- $(pk, sk) \leftarrow Gen(1^n)$ where n is a security parameter, pk is the public key and sk is the secret key.
- Protocol consisting of a Prover P and a Verifier V
  P(pk,sk) V(pk)

  $\leftarrow$
  $\rightarrow$
  $\leftarrow$
  $\downarrow 0/1$

  The protocol is randomised and V outputs either 0 or 1.

- Correctness:

$$\Pr[Output(P(pk, sk) \leftrightarrow V(pk)) = 1] = 1$$

  $(pk, sk) \leftarrow Gen(1^n)$
  The verifier should accept if we run the protocol correctly.

Honest Verifier Security:

- denote $View(P(pk, sk) \leftrightarrow V(pk)$
  The $View$ denotes everything the verifier sees when the protocol is happening. It is also sometimes referred to as the transcript of the protocol from the point of view of the verifier.

- $IDGame_A(1^n)$ where A is the adversary and n is the security parameter
- $(pk, sk) \leftarrow Gen(1^n)$ where n is a security parameter, pk is the public key and sk is the secret key.
- A gets pk, $tr_i \leftarrow View(P(pk, sk) \leftrightarrow V(pk))$ i=1, ..., t , t=poly(n)
- A interacts with $V(pk)$, output whatever V outputs
- Construction
  If we have a digital signature scheme, then it is simple to construct an identification scheme from it. The verifier V picks a random string and asks the prover P to sign it and then when that signature is transmitted to V, he can verify it.
  Construction from Discrete Log:

$P(pk, sk)$		$V(pk)$
	$\xrightarrow{\quad a \quad}$	
	$\xleftarrow{\quad c \quad}$	$c \leftarrow D_{pk}$
	$\xrightarrow{\quad z \quad}$	$V(pk, (a, c, z)) = 1$

  We want to look at restricted protocols called Σ-protocols.
  $(G, g, q) \leftarrow GroupGen(1^n)$ where G is a group with prime order q and g is the generator of the group
  Assumption: Discrete Log is hard for this group G.

$P(pk, sk)$		$V(pk)$
	$\xrightarrow{\quad a = g^r (r \leftarrow \mathbb{Z}_q) \quad}$	
	$\xleftarrow{\quad c \quad}$	$c \leftarrow \mathbb{Z}_q$
	$\xrightarrow{\quad z = r + c \cdot x \quad}$	$Ver(pk, (a, c, z)) = 1$

  $Ver(pk, (a, c, z))$: Check $a \cdot h^c = g^z$
  $a \cdot h^c = g^r \cdot g^{xc} = g^{r + c \cdot x} = g^z$

- Claim 1:
  $\exists$ PPT Sim,
  (a,c,z) $\leftarrow Sim(pk)$ such that $\forall (pk, sk) \leftarrow Gen(1^n)$ where Sim is basically a Simulator who simulates the transcripts himself without knowing the secret key.
  $View(P(pk, sk) \leftrightarrow V(pk)) \equiv Sim(pk)$
  Apparently, this is good for the security of the system because even if the adversary A sees many transcripts, he cannot do much because he could have generated the transcripts himself. He cannot learn anything more than the public key.

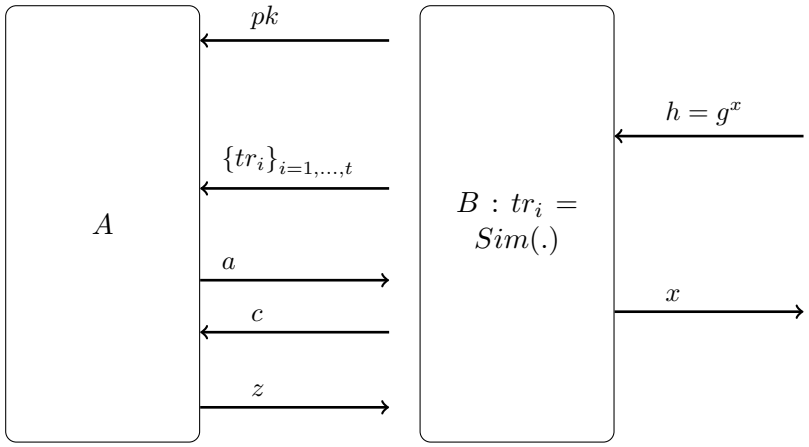
  Proof: If the Sim samples $c, z \leftarrow \mathbb{Z}_q$ and sets $a = \frac{g^z}{h^c}$ then the distribution of a given c,z is exactly the same as in View. In other words, the distribution of (a,c,z) is same in both the cases. The simulator basically generates the transcripts by running the protocol in a different order. $\square$

- Claim 2:
  $\exists$ PPT $Ext : sk = Ext(pk, a, (c, z), (c', z'))$ such that
  whenever c$\neq$ c' and $Ver(pk, (a, c, z)) = Ver(pk, (a, c', z')) = 1$, Then x'=x.
  Ext is a deterministic algorithm.

  Proof: $Ver(pk, (a, c, z)) = Ver(pk, (a, c', z')) = 1$
  $\iff a.h^c = g^z$
  $\iff a.h^{c'} = g^{z'}$
  $\Rightarrow h^{c-c'} = g^{z-z'}$
  $\Rightarrow h = g^{\frac{z-z'}{c-c'}}$ [c - c' $\neq$ 0]
  Ext outputs $x' = \frac{z-z'}{c-c'}$ □

The above two claims are somewhat contradictory. The 1st claim says that an adversary is able to create as many transcripts as he wants without knowing the secret key. On the other hand, the second claim says that if the adversary can verify two different transcripts with the same a, then he can break the system as he can recover the secret key.

Theorem 1 *Construction is a secure Identification scheme under Discrete Logarithm assumption.*



Proof: Assume $\exists$ PPT A s.t. $\Pr[IDGame_A(n) = 1] = \varepsilon(n) \neq negl(n)$
So A breaks the identification scheme and let us assume $\exists$ B that is a reduction that solves Discrete Logarithm. Here, a technique called rewinding is followed: B takes a snapshot upto the point it sees a. Then it rewinds and gives 2 c's and hopes that $Ver = 1$
Let w be the total state of A after it sends a.
Let $p_w = \Pr[W = w]$ $W \leftarrow$ random variable that ends up in the state w
$\delta_w = \Pr[IDGame = 1 | W = w]$
$\Pr[A \text{ succeeds}] = \sum_w p_w \cdot \delta_w = \varepsilon(n)$

$$\begin{aligned}
\Pr[B \text{ succeeds}] &= \sum_w p_w \cdot {\delta_w}^2 - \frac{1}{q} \qquad [\because Pr[c = c'] = \frac{1}{q}] \\
&= E[{\delta_w}^2] - \frac{1}{q} \\
&\geq E[\delta_w]^2 - \frac{1}{q} \qquad [\because \text{By Jensen's Inequality}] \\
&= \varepsilon^2(n) - \frac{1}{q} \neq negl(n) \qquad [\because \text{non-negl(n) - negl(n) = non-negl(n)}]
\end{aligned}$$

So, when A breaks the identification scheme, we are able to solve the Discrete Logarithm Problem with non-negligible probability, thus contradicting our assumption. Thus the construction is secure under the Discrete Logarithm Assumption.

□

This Identication based scheme is practically not useful because it cannot prevent man-in-the-middle attacks.