CSE 5852: Lecture 25

Jonathan Huang

December 15, 2016

1 Symmetric Identification/Authentication

In a symmetric setting, authentication either comes in the form of either a message authentication code or a stateful protocol between parties. The security requirement for a message authentication code is that an adversary cannot create new valid pairs, but can use previous ones. There needs to be some method to identify stale pairs (time is an easy one). Remember that message authentication is orthogonal is protecting confidentiality using encryption.

In this class we'll talk about identification and authentication in the public-key setting.

2 Public-Key Identification/Authentication

We'll start with the Schnorr identification scheme [Sch91]. The security of Schnorr is based on the hardness of finding discrete logs. Recall this assumption:

Assumption 1. *For any PPT $\mathcal{A}$, there exists a negligible ϵ such that for a random n-bit p and its generator and select a random $x \in \mathbb{Z}_p^*$,*

$$\Pr[\mathcal{A}(1^n, p, g, g^x \mod p) = x] \leq \epsilon(n).$$

The protocol is as follows:

1. Alice generates (p, g, x)
$pk = (p, g, g^x)$
$sk = (p, g, x)$
$y \leftarrow \{0, ..., p-2\}$
2. Sends g^y to Bob.
3. Bob creates an $r \leftarrow \{0, ..., r-2\}$ and sends this value to Alice.
4. Sends $s = rx + y \mod p-1$ to B
5. B checks that s has the form $rx + y \mod p-1$ by checking that:
$g^s \cdot (g^x)^{-r} = g^{rx+y-rx} = g^y$.

Our goal in showing security is two-fold:

Alice security Alice wants to make sure that she does not reveal any information about her public key. That is it should be completely hidden from Bob after execution of the protocol.

Bob security Bob should be convinced that only an Alice that actually knows the value x is reliably able to send correctly formed values s.

Security:

Alice To show that the security of Alice is preserved we argue that is possible to create an accepting transcript without knowledge of the private key x simply by selecting messages in a different order. Given pk, it is easy to produce triples of the form (g^y, r, s) with exactly the same distribution as given by executions of the protocol (w/ honest A). The idea is to reverse the order of the steps, choose uniform and independent $r, s \in \mathbb{Z}_p$ and then set $g^y = g^s \cdot g^{-r}$.

Bob Here we show an adversary I who can make B accept (w/ non-negl. pr) can be used to extract discrete log. This shows that I must know the value of x. We prove this claim below.

Proof. Claim: suppose $I(pk = (p, g, g^x))$ can make B accept with

$$Pr[a * \log(p)] \geq \frac{1}{poly(n)}.$$

Then we can use I to extract x. Let I be as follows

1. Take (p, g, g^x) as input
2. Send some $g^y = h \in$ zp*
3. Receive some $r \leftarrow 0, ..., p-2$
4. Return $s = rx + y$

We can run two I twice with inputs $I(p, g, g^x), r_1$ and $I(p, g, g^x), r_2$. This causes I to create two values s_1, s_2. We can then find x using: $\frac{s_1 - s_2}{r_1 - r_2}$ □

Note, that security is not guaranteed after a pair g^y, r has been used. The scheme is compromised afterwards. Thus this scheme is vulnerable to parallel attack (lots of authentication sessions with B).

References

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174, 1991.