EE 720: An Introduction to Number Theory and Cryptography (Spring 2019)

Lecture 24 — April 14, 2019

Instructor: Saravanan Vijayakumaran *Scribe: Saravanan Vijayakumaran*

1 Lecture Plan

- Schnorr Signature Scheme

2 Schnorr Identification Scheme

- The Schnorr signature scheme is based on the Schnorr identification scheme. We will discuss the latter first and show how to build the former from it.
- An identification scheme is an interactive protocol that allows one party (*the prover*) to prove its identity to another (*the verifier*).
- In the public-key setting, the verifier knows only the prover's public key pk. The prover wants to convince the verifier that it knows the secret key sk corresponding to the public key pk. So the prover's identity here is "the party which knows sk".
- Let $\mathcal{G}$ be a polynomial-time group generation algorithm. On input 1^n, it outputs a description of a cyclic group G, its order q (with $\|q\| = n$), and a generator $g \in G$.
- The Schnorr identification scheme
  - Prover runs $\mathcal{G}(1^n)$ to obtain (G, q, g).
  - Prover chooses x uniformly from $\mathbb{Z}_q$ and sets $y = g^x$.
  - Prover's public key is $pk = (G, q, g, y)$ and the secret key is $sk = x$.
  - Prover picks $k \leftarrow \mathbb{Z}_q$ and sends initial message $I = g^k$
  - Verifier sends a challenge $r \leftarrow \mathbb{Z}_q$
  - Prover sends $s = rx + k \bmod q$
  - Verifier checks $g^s \cdot h^{-r} \stackrel{?}{=} I$
- We want to argue two points regarding the protocol construction:
  - The verifier does not gain any knowledge about the secret key x.
  - A prover who does not know x cannot convince a verifier except with a negligible probability.
- How to quantify knowledge? This is difficult in general but we will say that some value Y does not contain more knowledge than X if Y can be efficiently computed from X. By efficient computation, we mean PPT algorithms.

- **Example:** Suppose $N = pq$ where p and q are n-bit primes. A party who knows $\{p, q\}$ has more knowledge than a party who only knows N. Since multiplication can be done in time which is polynomial in n, N can be efficiently computed from $\{p, q\}$. But there are no known PPT algorithms which can compute $\{p, q\}$ from N.

- Verifier does not gain any knowledge about x from the protocol transcript.
  - (I, r) is uniform on $G \times \mathbb{Z}_q$ and $s = \log_g(I \cdot y^r)$
  - Transcripts with same distribution can be simulated without knowing x
  - Choose r', s' uniformly from $\mathbb{Z}_q$ and set $I' = g^{s'} \cdot h^{-r'}$

- **Exercise:** Suppose G is a cyclic group of order q with generator g. Let $x \in \mathbb{Z}_q$ and $h = g^x$. Show that (I, r, s) and (I', r', s') have the same distribution where
  - $k \leftarrow \mathbb{Z}_q$, $I = g^k$, $r \leftarrow \mathbb{Z}_q$, and $s = rx + k \bmod q$
  - $r' \leftarrow \mathbb{Z}_q, s' \leftarrow \mathbb{Z}_q, I' = g^{s'} h^{-r'}$

- Suppose a malicious prover does not know x corresponding to $y = g^x$. Informally, if this prover is able to give correct responses with high probability then it must be able to generate responses s_1, s_2 to at least two different challenges $r_1, r_2 \in \mathbb{Z}_q$ for the same initial message I. This implies that

$$\begin{aligned} g^{s_1} \cdot y^{-r_1} = I = g^{s_2} \cdot y^{-r_2} &\implies y^{r_1 - r_2} = g^{s_1 - s_2} \\ &\implies x(r_1 - r_2) = s_1 - s_2 \\ &\implies x = (r_1 - r_2)^{-1} (s_1 - s_2) \end{aligned}$$

  So the prover can efficiently calculate the discrete logarithm of y with respect to g.

- **Theorem:** If the discrete-logarithm problem is hard relative to $\mathcal{G}$, then the Schnorr identification scheme is secure.[1]

3 Schnorr Signature Scheme

- The *Fiat-Shamir transform* provides a way to convert any interactive identification scheme into a non-interactive signature scheme. The idea is for the signer to act as a prover and use the cryptographic hash of the initial message I and the message to be signed m as the challenge r. In other words, the challenge r is set to $H(I, m)$ where the comma between I and m denotes concatentation, i.e. $H(I, m) = H(I \| m)$.

- The Schnorr signature scheme
  - **Gen**: Run $\mathcal{G}(1^n)$ to obtain (G, q, g). Choose a uniform $x \in \mathbb{Z}_q$ and set $y = g^x$. The private key is x and the public key is (G, q, g, y). As part of the key generation, a function $H : \{0, 1\}^* \mapsto \mathbb{Z}_q$ is specified.

[1] Note that we have not defined security of identification schemes formally. It is defined in Definition 12.8 of KL. It essentially prevents a malicious prover (i.e. a prover who does not know the secret key) from convincing a verifier with a non-negligible probability.

- **Sign**: On input private key x and message $m \in \{0,1\}^*$, choose k uniformly from $\mathbb{Z}_q$ and set $I = g^k$. Then compute $r = H(I, m)$, followed by $s = rx + k \bmod q$. Output the signature (r, s).
- **Vrfy**: On input public key (G, q, g, y), a message m, and a signature (r, s), compute $I = g^s \cdot y^{-r}$ and output 1 if $H(I, m) = r$.

4 References and Additional Reading

- Sections 12.5.1 from Katz/Lindell