Lecture 17: More Constructions

Instructor: Omkant Pandey

Spring 2017 (CSE 594)

Today

- Some more constructions based on Discrete Log
- Specifically:
  - Collection of OWFs
  - CRHFs
  - Diffie-Hellman Key Exchange
- Scribe notes volunteers?

Discrete Log Based Collection of OWFs

- Consider the following collection $DL = \{f_i : D_i \to R_i\}$:
  - $I = \{(q, g) | q \in \Pi_n, g \in \mathsf{Gen}_{G_q}\}$
  - $D_i = \{x | x \in \mathbb{Z}_q\}$
  - $R_i = G_q$
  - $f_{q,g}(x) = g^x \in G_q$.
- The function is easy to compute, and elements are also easy to sample from the domain. From the DL Assumption, it also follows that $f_{q,g}$ is hard to invert.
- The only issue is sampling the **index**, namely (q, g) such that g is a generator. In general it is not known, however, in special cases such as G_q being a subgroup of $\mathbb{Z}_p^*$ for a safe prime p, it is easy. So we have to assume that the G_q comes with an algorithm to sample from I.

CRHFs based on Discrete Log

- Hash function for compressing **1-bit** only:
  - **DL Problem:** for a large random prime p, given $(g, p, y = g^x \mod p)$, find x. (hard)
  - $H = \{h_i\}_i$ where h_i is defined by $i = (p, g, y)$ as follows:
    The input is $x \| b$ where b is a bit and $x \in \mathbb{Z}_p^*$.
    The output is:
    $$h_i(x \| b) = h_{p,g,y}(x, b) = g^x \cdot y^b \mod p.$$

Proving Collision-Resistance

- Recall the function: $h_i(x\|b) = h_{p,g,y}(x,b) = g^x \cdot y^b \mod p$
- Proof of collision-resistance:
  - Suppose A finds $x\|b \neq x'\|b'$ s.t. $h_i(x\|b) = h_i(x'\|b')$.
  - I.e., $g^x \cdot y^b \mod p = g^{x'} \cdot y^{b'} \mod p$
  - If $b = b'$, then $g^x = g^{x'} \mod p \Rightarrow x = x'$.
  - Therefore, $b \neq b'$. Suppose $b = 0, b' = 1$.
  - We have: $g^x = g^{x'} \cdot y \mod p \Rightarrow y = g^{x-x'} \mod p$.
  - $x - x'$ is the discrete log of y.
  - Therefore, A is solving the DL instance (p, g, y).
  - This is hard and hence a contradiction (QED)

More efficient construction

- Construction from based on prime order groups: we work with prime order groups where discrete log is hard. (For example, $p = 2q + 1$ where p, q are both primes and g generates a prime order sub-group G_q of $\mathbb{Z}_p^*$).
  - $H = \{h_i\}_i$ where $i = (p, g, y)$ is defined by a safe prime p and a prime-order generator g and h_i is defined as follows: input is a pair of elements $x_1 \| x_2$ where $x_1, x_2 \in \mathbb{Z}_q$; and output is:

$$h_i(x_1 \| x_2) = h_{p,g,y}(x_1 \| x_2) = g^{x_1} \cdot y^{x_2} \mod p.$$

- Proof of collision resistance:
  - If A finds $x_1 \| x_2 \neq x_1' \| x_2'$ s.t. $h_i(x_1 \| x_2) = h_i(x_1' \| x_2')$.
    $\implies y^{x_2 - x_2'} = g^{(x_1 - x_1')} \mod p$.
    Since g generates an order q subgroup, the DL of y w.r.t. g is:

$$(x_1 - x_1') \times (x_2 - x_2')^{-1} \mod q$$

Note that inverse always exists in this case.

Key Exchange: Definition

- Alice picks a local randomness r_A
- Bob picks a local randomness r_B
- Alice and Bob engage in a protocol and generate the transcript τ
- Alice's view $V_A = (r_A, \tau)$ and Bob's view $V_B = (r_B, \tau)$
- Eavesdropper's view $V_E = \tau$
- Alice outputs k_A as a function of V_A and Bob outputs k_B as a function of V_B
- Correctness: $\Pr_{r_A, r_B}[k_A = k_B] \approx 1$
- Security: $(k_A, V_E) \equiv (k_B, \tau) \approx (r, \tau)$

Diffie-Hellman Key Exchange

- Protocol is based on discrete-logarithms
- The Diffie-Hellman Key-Exchange Protocol:
  - Let p be a large safe prime, i.e., $p = 2q + 1$ for prime q.
  - Let g be a generator of order q subgroup G_q of $\mathbb{Z}_p^*$.
  - Alice picks $x \leftarrow \mathbb{Z}_p^*$ and sends $X = g^x \mod p$ to Bob.
  - Bob picks $y \leftarrow \mathbb{Z}_p^*$ and sends $Y = g^y \mod p$ to Alice.
  - Alice and Bob both can compute $K = g^{xy} \mod p$ as follows:

Alice	Bob
$K = Y^x \mod p$	$K = X^y \mod p$
$= (g^y \mod p)^x \mod p$	$= (g^x \mod p)^y \mod p$
$= g^{xy} \mod p$	$= g^{xy} \mod p$

- Why is this secure?

2-round Key Exchange $\implies$ PKE

- In general: we do not know if every Key Exchange protocol can be used to construct a public-key encryption scheme.
- However, if the protocol has only 2 rounds: i.e., one message from each party, we can build PKE from it.
- Idea: use the key as a (computational) one-time pad