CS 7880 Graduate Cryptography October 25th, 2017

Lecture 14: Public Key Encryption

Lecturer: Daniel Wichs *Scribe: Schuyler Rosefield*

1 Topic Covered

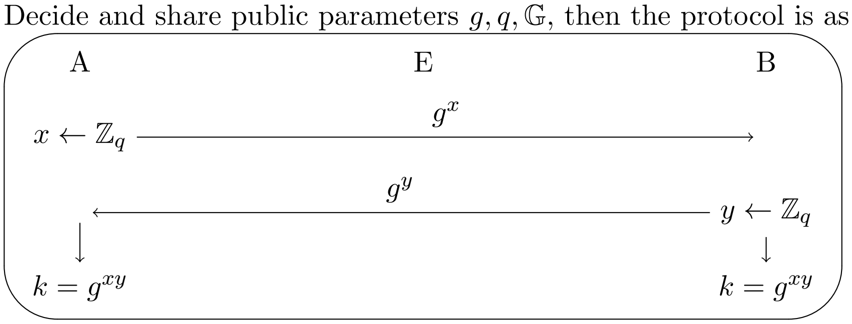
- Discrete Log Assumptions
- Public Key Encryption from Discrete Log Assumptions

DEFINITION 1 Discrete Log Assumption (DL)
$Pr[A(g^x) = x : x \leftarrow \mathbb{Z}_q] = negl(n)$

DEFINITION 2 Computational Diffie-Hellman (CDH)
$Pr[A(g^x, g^y) = g^{xy} : x, y \leftarrow \mathbb{Z}_q] = negl(n)$

DEFINITION 3 Decisional Diffie-Hellman (DDH)
$(g^x, g^y, g^{xy}) \approx (g^x, g^y, g^z) | x, y, z \leftarrow \mathbb{Z}_q$

DEFINITION 4 Diffie-Hellman Key Exchange
Decide and share public parameters $g, q, \mathbb{G}$, then the protocol is as follows



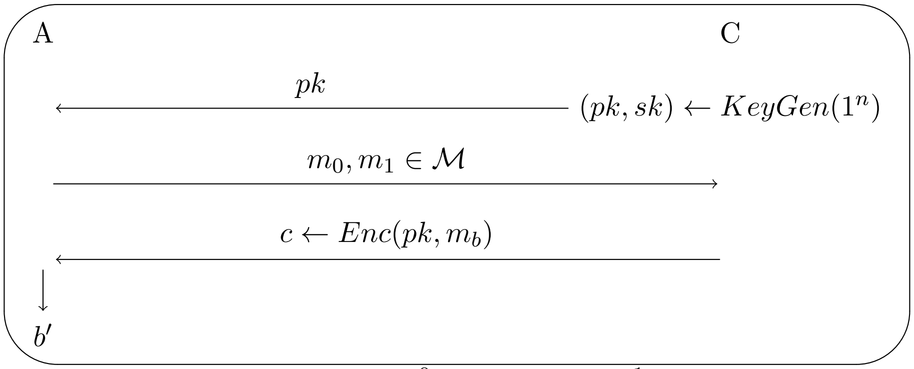
If DDH isn't assumed, so g^{xy} can be distinguished from random, but is still hard to compute.

In the RO model, each party could instead take $K = RO(g^{xy})$. In the standard model, instead the protocol can be run n times for an n bit key, and in each iteration take $hc(g^{xy})$ which is a uniform random bit.

DEFINITION 5 Public Key Encryption
Three functions

$$(pk, sk) \leftarrow Gen(1^n)$$
$$c \leftarrow Enc(pk, m)$$
$$m \leftarrow Dec(sk, c)$$

Correctness: $\forall m, Pr[Dec(sk, Enc(pk, m)) = m | (pk, sk) \leftarrow Gen(1^n)] = 1$
Security: Denote game $PKCPASec^b$ as follows:



We have security if $PKCPASec^0 \approx PKCPASec^1$

DEFINITION 6 ElGamal Encryption Scheme

$$
\begin{aligned}
(pk, sk) &\leftarrow Gen(1^n) = g^x, x; x \leftarrow \mathbb{Z}_q \\
c &\leftarrow Enc(pk, m) = (g^y, pk^y \cdot m); y \leftarrow \mathbb{Z}_q \\
m &\leftarrow Dec(sk, c) = h_2/h_1^{sk}; (h1, h2) \leftarrow c
\end{aligned}
$$

Proof:

Correctness: This from the definition exactly

Security Using the DDH assumption.

Define hybrids

$$
\begin{aligned}
H_1 &: c \equiv (g^y, g^z \cdot m_0) | z \leftarrow \mathbb{Z}_q \\
H_2 &: c \equiv (h_1, h_2) | h_1, h_2 \leftarrow G \\
H_3 &: c \equiv (g^y, g^z \cdot m_1) | z \leftarrow \mathbb{Z}_q
\end{aligned}
$$

then $PKCPASec^0 \approx H_1 \approx H_2 \approx H_3 \approx PKCPASec^1$ □

DEFINITION 7 CRHF from DL

Modify the definition of CRHF to specify the seed as $s \leftarrow Gen(1^n)$. The CRHF is this the combination of Gen, H_s

$\forall PPTA, Pr[H_s(x) = H_s(x'), x \neq x' : s \leftarrow Gen(1^n), (x, x') \leftarrow A(s)] = negl(n)$

We define the construction as follows:

Assume that q is prime, and so G is a prime-ordered group.

$s = (g, h) = G^2$

$H_s : \mathbb{Z}_q^2 \rightarrow G$, $H_s(x_1, x_2) = g^{x_1} h^{x_2}$

Theorem 1 *The above is a CRHF under DL.*

Proof Sketch: *The attacker A generates $(x_1, x_2) \neq (x'_1, x'_2)$ s.t. $g^{x_1} h^{x_2} = g^{x'_1} h^{x'_2}$ with non-negligible probability.*

This is equivalent to saying $h^{x_2 - x'_2} = g^{x_1 - x'_2}$

As G is a field, then we can find inverses so then we can find $h = g^{(x_1 - x'_1)/(x_2 - x'_2)}$ mod q. *Since $(x_1, x_2) \neq (x'_1, x'_2) \Rightarrow x_2 \neq x'_2$.*

This is equivalent to saying finding the discrete log of $h = g^z$. □

Definition 8 *PRG from DDH*

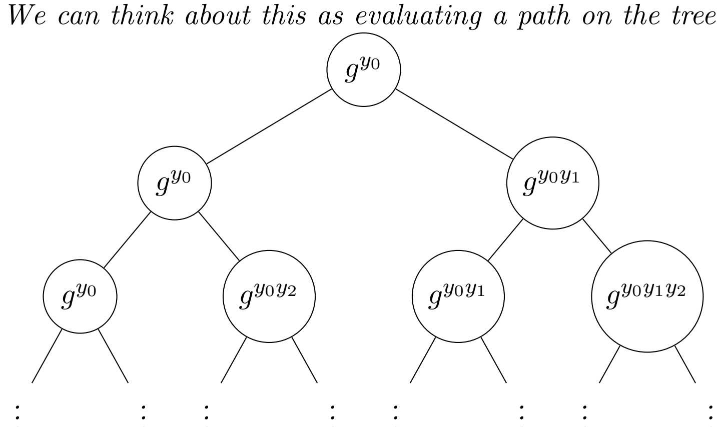
Similarly to our change in the CRHF, we also slightly change the definition of the PRG.
$G : \mathbb{Z}_q^2 \to G^3$, $G(x, y) = (g^x, g^y, g^{xy})$
This follows immediately from DDH.

Note 1 *As a generalization of the above, we can take*
$G : \mathbb{Z}_q^{l+1} \to G^{2l+1}$, $G(x, y_1, \ldots y_l) = (g^x, g^{y_1}, g^{xy_1}, g^{y_2}, g^{xy_2}, \ldots)$

Proof Sketch: *Define hybrids* $H_0 = G, H_i = (g^x, \mathit{uniform}, g^{y_{i+1}}, g^{xy_{i+1}}, \ldots), H_l = (\mathit{uniform})$
Wish to find that $H_i \approx H_{i+1}$*, breaking this would be equivalent to breaking DDH.* □

Definition 9 *Naor-Reingold PRF*

$F_{k=(y_0,\ldots y_l)}(x \in \{0,1\}^l) = g^{y_0 \prod_{i;x_i=1} y_i}$
We can think about this as evaluating a path on the tree



Proof Sketch: *First, consider this tree as a PRG. Instead of outputting a single leaf, output all* 2^l *elements on level* l *in the tree. If we take hybrids where we replace level* l *with uniform values, then the elements on level* $l+1$ *are equivalent to the results in the above PRG.*

To prove as a PRF, the a similar argument to the GGM construction can be made. □

Definition 10 *Distributed decryption for ElGamal*

We can use additive secret sharing to distribute $x_1, \ldots x_n$ *s.t.* $\sum_i x_i = x$ *to* n *computers.*

Then to recover the message from an encryption, each computer can generate g^{yx_i}*, and taking* $\prod_i g^{yx_i} = g^{xy}$ *allows us to get th emessage.*