Class 1: Diffie-Hellman Key Exchange, Collision-Resistant Hash Function, and Elgamal Encryption Scheme

The most efficient encryption schemes are symmetric (e.g., AES), so assume that both the sender and the receiver share a secret key. This is problematic in many cases since one often wants to interacts with users/servers that we have never interact with before (e.g., on the Internet). To solve this issue, one would like a protocol that allows to compute a secret shared key by communicating over a public, insecure channel. This is the purpose of public key exchange. This class is largely inspired by [2], Chapter 10.

1 2-Party Public Key Exchange

1.1 Key Exchange

In 2-party public key exchange, 2 users want to run a protocol over an insecure channel such that at the end of the protocol, both users know a shared secret key but eavedroppers that can see the messages that were sent over the channel during the execution of the protocol should not be able to recover this secret key.

Definition 1 (Key Exchange Protocol). *A key exchange protocol P is a pair of probabilistic machines (A, B) that take turns in sending messages to each other. At the end of the protocol, when both machines terminate, they both obtain the same value k. A protocol transcript T_P is the sequence of messages exchanged between the parties in one execution of the protocol. The protocol being randomized, the transcript is a random variable, which is a function of the random bits generated by A and B.*

Moreover, for security, an eavesdropping adversary $\mathcal{A}$ with knowledge of T_P should not be able to generate k. We define the following security game:

- *The challenger runs the protocol between A and B to generate a shared key k and sends the transcript T_P to the adversary $\mathcal{A}$.*
- *$\mathcal{A}$ outputs a guess k' for the shared key and wins if $k' = k$.*

We denote by $\mathsf{Adv}(\mathcal{A})$ the probability that $\mathcal{A}$ correctly guesses k and call it the advantage of the adversary. We say that P is secure if for all PPT adversary $\mathcal{A}$, its advantage is negligible.

A natural way to build a key exchange protocol is to rely on some functions E, F with the following properties:

1. E is easy to compute;
2. Given a and $E(b)$, it is easy to compute $F(a, b)$;
3. Given $E(a)$ and b, it is easy to compute $F(a, b)$;
4. Given $E(a)$ and $E(b)$, it is hard to compute $F(a, b)$.

In particular, for item 4 to not contradict items 2 and 3, E has to be hard to invert.

Given such functions, it is easy to define a secure key exchange protocol: A and B sample a and b at random respectively, and send to each other $E(a)$ and $E(b)$ respectively. Both can then compute $F(a, b)$ easily but an eavedropper that only knows $E(a)$ and $E(b)$ cannot.

1.2 Diffie-Hellman Key Exchange

Diffie and Hellman [1] propose a simple way to instantiate the above idea using a group $\mathbb{G}$ relying on the hardness of the discrete logarithm in $\mathbb{G}$. We first recall the discrete logarithm problem.

Definition 2 (Discrete Logarithm). *Let $\mathbb{G} = \langle g \rangle$ a cyclic group generated by g. The discrete logarithm (DL) problem in $\mathbb{G}$ consists, given a group element $h \in \mathbb{G}$, in finding x such that $h = g^x$. The DL assumption in $\mathbb{G}$ corresponds to the hardness of this problem in $\mathbb{G}$ (i.e., the advantage of any PPT adversary in solving DL is negligible).*

Today, the best known algorithms in cryptographic groups (e.g., elliptic curves) run in time $\sqrt{|\mathbb{G}|}$.

Definition 3 (Diffie-Hellman Key Exchange). *Consider a cyclic group $\mathbb{G}$ of prime order p and generated by g. The Diffie-Hellman protocol works as follows:*

- *A picks $a \xleftarrow{\$} \mathbb{Z}_p$ and sends g^a to B;*
- *B picks $b \xleftarrow{\$} \mathbb{Z}_p$ and sends g^b to A;*
- *A and B compute their shared key g^{ab} by computing $(g^b)^a$ and $(g^a)^b$ respectively.*

It is immediate to see that the security of the above protocol relies of the fact that given g, g^a, g^b, it is hard to compute g^{ab}. This problem corresponds to the Computational Diffie-Hellman (CDH) problem and its hardness to the so-called CDH assumption.

Definition 4 (CDH). *Let $\mathbb{G} = \langle g \rangle$ a cyclic group generated by g. The CDH problem in $\mathbb{G}$ consists, given a group elements g, g^a, g^b in computing g^{ab}. The CDH assumption in $\mathbb{G}$ corresponds to the hardness of this problem in $\mathbb{G}$ (i.e., the advantage of any PPT adversary in solving CDH is negligible).*

1.3 Hoping for the best

We have just seen that assuming that CDH holds in $\mathbb{G}$, the Diffie-Hellman protocol is secure against any PPT eavedropper. However, this is still a fairly weak security guarantee for the shared secret key g^{ab}. Indeed, even if it is hard to compute it, it could still be the case that it is easy to compute half of the bits of it.

Ideally, one would like the shared key to be as strong as a uniformly random key. This gives rise to another assumption, called the Decisional Diffie-Hellman assumption, which states that the shared key should be computationally indistinguishable from a uniformly random key.

Definition 5 (DDH). *Let $\mathbb{G} = \langle g \rangle$ a cyclic group generated by g. The DDH assumption holds in $\mathbb{G}$ if the advantage of any PPT adversary $\mathcal{A}$ in the following game is negligible. The game is defined as follows:*

- *The challenger samples $a, b, c \xleftarrow{\$} \mathbb{Z}_p$ and flips a coin $r \in \{0, 1\}$;*
- *If $r = 0$, it sends (g, g^a, g^b, g^{ab}) to the adversary, otherwise it sends (g, g^a, g^b, g^c);*
- *$\mathcal{A}$ outputs a guess r' for r and wins if $r' = r$.*

We define the advantage of $\mathcal{A}$ as $|2 \Pr[\mathcal{A} \text{ wins}] - 1|$.[1]

1.4 Summary of the assumptions

We have defined 3 assumptions that are now standard assumptions in cryptography: the Discrete Log assumption, the Computational Diffie-Hellman assumption, and the Decisional Diffie-Hellman assumption.

It is easy to see that, as assumptions, we have DDH $\Rightarrow$ CDH $\Rightarrow$ DL (and the reverse directions if we consider the problems, since solving DL is a way to solve CDH, and solving CDH is a way to solve DDH).

Remark 1. Note that all these problems are hard on average, since it is possible to randomize the instances by raising the group elements to a random (known) power. For instance, if one has an algorithm that solves DL on average, one can use it to solve any instance (g, g^a) by running is on $(g, (g^a)^\alpha)$ where $(g^a)^\alpha$ is now uniformly random in $\mathbb{G}$ and converting the solution x into a solution for the original instance by computing $x \times \alpha^{-1}$ over $\mathbb{Z}_p$ (which is easy). The same kind of reasoning applies to CDH and DDH.

In the following of this class, we construct cryptographic primitives based on these assumptions.

[1] In particular, an adversary that outputs a uniformly random guess r' has probability $1/2$ to win, and its advantage is 0.

2 Collision-Resistant Hash Function from DL

Definition 6 (Collision-Resistant Hash Function). *A collision-resistant hash function (CRH) is a pair of PPT algorithms* $(\mathsf{Setup}, \mathsf{H})$ *such that:*

- $\mathsf{Setup}(1^\lambda)$ *outputs public parameters* pp*;*
- $\mathsf{H}_{\mathsf{pp}} : \mathcal{X} \to \mathcal{Y}$ *is defined by public parameters* pp *such that* $|\mathcal{Y}| < |\mathcal{X}|$*.*

Moreover, we require that for any PPT adversary $\mathcal{A}$*, its advantage in winning in the following game is negligible (in* λ*):*

- *The challenger sends the public parameters* pp *defining the function to* $\mathcal{A}$*;*
- $\mathcal{A}$ *outputs a pair* (x_0, x_1) *and wins if* $x_0 \neq x_1$ *and* $\mathsf{H}(x_0) = \mathsf{H}(x_1)$*.*

Its advantage is defined as its probability of success in the above game.

Theorem 1. *Assuming DL holds in* $\mathbb{G} = \langle g \rangle$ *and let* h *be a random group element in* $\mathbb{G}$*. Then the function* $\mathsf{H} : \mathbb{Z}_p \times \mathbb{Z}_p \mapsto \mathbb{G}$ *defined as* $\mathsf{H}(a, b) = g^a h^b$*, where* g, h *are public parameters, is a collision-resistant hash function.*

Proof. Assume there exists an efficient adversary $\mathcal{A}$ against the above CRH, one builds an efficient solver $\mathcal{B}$ against the DL problem in $\mathbb{G}$ as follows. $\mathcal{B}$ gets a random DL instance (g, g^a) and sets $h = g^a$. It sends (g, h) as the description of H to $\mathcal{A}$. When $\mathcal{A}$ outputs $(a_0, b_0), (a_1, b_1)$, $\mathcal{B}$ outputs $(a_0 - a_1) \cdot (b_1 - b_0)^{-1}$. If $(a_0, b_0), (a_1, b_1)$ is a collision, then $g^{a_0} h^{b_0} = g^{a_1} h^{b_1}$, so $a_0 + ab_0 = a_1 + ab_1$, and then since $b_0 \neq b_1$ (otherwise necessarily $a_0 = a_1$ and it is not a collision), we have $a = (a_0 - a_1) \cdot (b_1 - b_0)^{-1}$. This concludes the proof.

3 DDH-based Public-Key Encryption

Definition 7 (Public Key Encryption). *A public key encryption (PKE) scheme is a tuple of PPT algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *such that:*

- $\mathsf{Gen}(1^\lambda)$ *outputs a pair* (pk, sk) *of public and secret keys;*
- $\mathsf{Enc}(pk, m)$ *on input a public key* pk *and a message* m *outputs a ciphertext* ct*;*
- $\mathsf{Dec}(sk, ct)$ *on input a secret key* sk *and a ciphertext* ct *outputs a message* m*.*

Moreover, we require that the following:
Correctness: *For any* m*,* $\Pr[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m) = m] \geq 1 - negl(\lambda)$ *where the probability is over* $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ *and the randomness of* Enc*.*
IND-CPA Security: *For any PPT adversary* $\mathcal{A}$*, its advantage in the following game is negligible. The game is defined as follows:*

- *The challenger picks* $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ *and sends* pk *to* $\mathcal{A}$*;*
- $\mathcal{A}$ *sends a pair of messages* (m_0, m_1) *to the challenger;*
- *The latter picks* $b \xleftarrow{\$} \{0, 1\}$ *and sends back* $\mathsf{Enc}(pk, m_b)$ *to* $\mathcal{A}$*;*
- $\mathcal{A}$ *outputs a guess* b' *for* b *and wins if* $b' = b$*.*

Its advantage is defined as $|2 \Pr[\mathcal{A} \text{ wins}] - 1|$*.*

It is very easy to see that the following construction by Elgamal [3] is an IND-CPA secure PKE scheme assuming DDH.

- $\mathsf{Gen}(1^\lambda)$: pick $s \xleftarrow{\$} \mathbb{Z}_p$ and output $pk = g^s$ and $sk = s$;
- $\mathsf{Enc}(pk, m)$: pick $r \xleftarrow{\$} \mathbb{Z}_p$ and output $ct = (ct_1, ct_2) = (g^r, (g^s)^r \cdot m)$;
- $\mathsf{Dec}(sk, ct)$: output ct_2 / ct_1^s.

References

1. W. Diffie, M. E. Hellman. New Directions in Cryptography. IEEE Trans. Information Theory, 1976.
2. D. Boneh, V. Shoup. A Graduate Course in Applied Cryptography.
3. T. Elgamal. A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. CRYPTO, 1984.