1 Diffie-Hellman Key Exchange Protocol

In 1976, Whitefield Diffie and Martin Hellman published their paper *New Directions in Cryptography*. This is a revolution in modern cryptography. In this lecture, we introduce the concrete key-exchange protocol in the seminal paper. We also investigate its security in the presence of passive adversaries.

1.1 Concrete Diffie-Hellman Protocol

We give the concrete Diffie-Hellman key-exchange protocol in Figure 1. Observe that, \( y_B^x = y_A^x \equiv k_A \). So the two parties have computed the same value, which is an element in \( \mathbb{Z}_p^* \).

![Diagram of the Diffie-Hellman key exchange protocol](image)

**Common Input:** \( (g, p, m) \)

**Alice**

\[
\begin{align*}
&x_A \leftarrow \mathbb{Z}_m \\
y_A \leftarrow g^{x_A} \mod p \\
k_A \leftarrow y_A^{x_B} \mod p \\
\text{output } k_A
\end{align*}
\]

**Bob**

\[
\begin{align*}
&x_B \leftarrow \mathbb{Z}_m \\
y_B \leftarrow g^{x_B} \mod p \\
k_B \leftarrow y_B^{x_A} \mod p \\
\text{output } k_B
\end{align*}
\]

Figure 1: Diffie-Hellman key-exchange protocol. Here \( p \) is large prime, \( g \) generator in group \( \mathbb{Z}_p^* \) of order \( m \).

Recall that the five steps, “Goal-Design-Primitive-Model-Proof”, mentioned in Lecture 1. Naturally, we will ask: how can we model the security for Diffie-Hellman protocol? what are the underlying assumptions required for Diffie-Hellman key exchange to be secure?

1.2 Related Number-Theoretic Problems

Here we introduce some potential number-theoretic hard problems which allow Diffie-Hellman protocol to reduce. In the next subsections, we search the proper security definition for the protocol and reduce the protocol to a proper number-theoretic assumption.

**Definition 1 (Discrete Logarithm Problem)** The discrete logarithm (DL) problem is following: given a group \( G \) with order \( m \), a generator \( g \), and \( y \in G \), find integer \( x \in \mathbb{Z}_m \) such that \( g^x = y \).
The DL problem is in many groups notoriously hard, for instance in $\mathbb{Z}_p^*$. A related problem is computational Diffie-Hellman (CDH) problem.

**Definition 2 (Computational Diffie-Hellman Problem)** The computational Diffie-Hellman (CDH) problem is following: given a group $G$ with order $m$, a generator $g$, and $g^{x_A}, g^{x_B}$ where $x_A, x_B \in \mathbb{Z}_m$, compute $g^{x_A x_B}$.

Clearly, if we find $x$ from $g^x$, we could solve CDH problem by a single exponentiation, so

**Lemma 3** The CDH problem is no harder than the DL problem.

However, we do not know if the opposite direction holds.

**Definition 4 (Decisional Diffie-Hellman Problem)** The decisional Diffie-Hellman (DDH) problem is following: given a group $G$, a generator $g$, and $g^{x_A}, g^{x_B}, g^{x_C}$, where $x_A, x_B \in \mathbb{Z}_m$; and where $x_C$ is chosen either as $x_C = x_A x_B$, or uniformly random from $\mathbb{Z}_m$; decide which one of the two is the case.

If we can solve CDH problem, then we can solve DDH problem, by computing $g^{x_A x_B}$ and comparing this to $g^{x_C}$. So we have:

**Lemma 5** The DDH problem is no harder than the CDH problem.

Moreover, either of these two problems are no harder than the discrete-logarithm problem.

In the above formulation of the problem, the choice of the group is “conveniently” abstracted out. In fact we have to be very careful in our choice of the proper parameters to make sure the underlying problems are indeed hard. To demonstrate this, we give an example below that shows that the DL problem can be solved in polynomial time if we choose a “weak” group.

**Example** Choose group $\mathbb{Z}_p^*$, where $g$ is a generator, $p$ is a large prime, and the order $p \equiv 1 = q_1 \cdot q_2 \cdot \cdots q_s$ such that $q_i$’s are “small” primes, $G_i$ is subgroup of $\mathbb{Z}_p^*$ of order $q_i$. Observe that

$$f_i : \mathbb{Z}_p^* \to G_i \quad x \mapsto x^{rac{p-1}{q_i}}$$

is a group homomorphism. Let $g_i = g^{\frac{p-1}{q_i}}$. Then the order of $g_i$ is $q_i$. Because $q_i$ is small prime, we can **brute force** the discrete logarithm problem inside $G_i$.

Now given $y = g^x \mod p$, we can compute $x$ by two steps: first, from $y^{\frac{p-1}{q_i}} = g_i^x \mod q_i \mod p$, we brute force $x_i = x \mod q_i$, where $i = 1, 2, \ldots, s$; then we use Chinese Remainder Theorem to solve $x$ from the equations $x = x_i \mod q_i, i = 1, 2, \ldots, s$.

To avoid this kind of attack in the above setting, we can select $\mathbb{Z}_p^*$ with a large subgroup. For example, if $p = 2q + 1$, $q$ is a prime. In this case, $p$ is a large prime, so is $q$, and $\mathbb{Z}_p^*$ will have a large subgroup with size $q$, $G_q = QR(p)$.
1.3 The Formal Definition of DDH Assumption

Informally, the DDH assumption is that it is hard to distinguish between tuples of the form \((g, g^x, g^y, g^{xy})\) and \((g, g^x, g^y, g^z)\), where \(g\) is a generator and \(x, y, z\) are random. More formally, \(Gen\) satisfies the DDH assumption if the DDH problem is hard for \(Gen\), where this is defined as follows:

**Definition 6 (DDH Assumption)** The DDH problem is hard for \(Gen\) if the following two probability distributions \(D_\lambda\) and \(R_\lambda\) are computationally indistinguishable:

\[
D_\lambda \overset{\text{def}}{=} \{ G, m, g \leftarrow Gen(1^\lambda); x, y \overset{\$}{\leftarrow} \mathbb{Z}_m : (G, m, g, g^x, g^y, g^{xy}) \}
\]

\[
R_\lambda \overset{\text{def}}{=} \{ G, m, g \leftarrow Gen(1^\lambda); x, y, z \overset{\$}{\leftarrow} \mathbb{Z}_m : (G, m, g, g^x, g^y, g^z) \}
\]

where \(m = \text{ord}(g)\), i.e. for any PPT \(A\), it holds that

\[
\text{Adv}^A(\lambda) = \left| \Pr_{\gamma \leftarrow D_\lambda}[A(\gamma) = 1] - \Pr_{\gamma \leftarrow R_\lambda}[A(\gamma) = 1] \right| \leq \text{negl}(\lambda)
\]

where \(\text{negl}()\) is negligible in \(\lambda\).

1.4 Modeling Security against Passive Adversaries

Consider the setting of Diffie-Hellman key-exchange protocol we discussed above. Now we define \(\text{trans}_{A,B}(1^\lambda)\) as a random variable that contains a full interaction of the two players \(A\) and \(B\) over the channel. And define \(key(\tau)\) as the key that the two players compute at the end of the protocol with transcript \(\tau\).

1.4.1 A First Attempt for Defining Security

Consider the definition below: for all PPT adversary \(A\)

\[
\Pr_{\tau \leftarrow \text{trans}_{A,B}(1^\lambda)}[A(\tau) = key(\tau)] \leq \text{negl}(\lambda)
\]

This definition is rather weak because we ask too much from the adversary. The number of bits protected is about \(\Omega(\log^2(\lambda))\).

1.4.2 The Second Definition

Consider another definition: for all PPT adversary \(A\), and for all predicate \(V\)

\[
\Pr_{\tau \leftarrow \text{trans}_{A,B}(1^\lambda)}[A(\tau) = V(key(\tau))] \leq \frac{1}{2} + \text{negl}(\lambda)
\]

We ask too little from the adversary and this is a very strong definition. We will see why the definition is not reasonable in a moment.

Now we try to prove the key-exchange protocol is secure in this definition. Assume there is a PPT adversary \(A\) can break the key-exchange protocol, and let \(A\) and predicate \(V\) such that

\[
\Pr_{\tau \leftarrow \text{trans}_{A,B}(1^\lambda)}[A(\tau) = V(key(\tau))] \geq \frac{1}{2} + \alpha
\]
where $\alpha$ is non-negligible. Let $B$ be a distinguisher for DDH defined as follow: given $\sigma = \langle G, m, g, a, b, c \rangle$, $B$ uses $\sigma$ to form a transcript $\tau_\sigma = \langle G, m, g, a, b \rangle$; then $B$ will simulate $A$ on $\tau_\sigma$ and obtain its output $S$; finally, $B$ will return $V(c) \leftarrow S$. Let $\text{Prob}[V(c) = 1] = \gamma$.

- if $\sigma \leftarrow D_\lambda$, then $c = \text{key}(\tau_\sigma)$, and $\text{Prob}_{\sigma \leftarrow D_\lambda}[B(\sigma) = 1] \geq \frac{1}{2} + \alpha$;

- if $\sigma \leftarrow R_\lambda$, then $c \leftarrow G$, and

$$\text{Prob}_{\sigma \leftarrow R_\lambda}[B(\sigma) = 1] = \text{Prob}_{\langle G, m, g, a, b, c \rangle \leftarrow R_\lambda}[A(G, m, g, a) = V(c)] = \text{Prob}_{\langle \tau_\sigma \rangle \leftarrow R_\lambda}[A(\tau_\sigma) = V(c)] = \text{Prob}_{\langle \tau_\sigma \rangle \leftarrow R_\lambda}[A(\tau_\sigma) = 1 | V(c) = 1] \cdot \text{Prob}[V(c) = 1] + \text{Prob}[A(\tau_\sigma) = 0 | V(c) = 0] \cdot \text{Prob}[V(c) = 0] = \text{Prob}[A(\tau_\sigma) = 1] \cdot \text{Prob}[V(c) = 1] + \text{Prob}[A(\tau_\sigma) = 0] \cdot \text{Prob}[V(c) = 0].$$

$$\gamma \geq \frac{1}{2} \left( \text{Prob}[A(\tau_\sigma) = 1] + \text{Prob}[A(\tau_\sigma) = 0] \right) \cdot \frac{1}{2} = \frac{1}{2}.$$

We can see that $B$ can break the DDH assumption when $\gamma = \frac{1}{2}$ given $A$ can break the key-exchange protocol. However, when $\gamma \neq \frac{1}{2}$, it is very easy to find $V$ which the adversary can guess easily. As a result, all schemes will fail in this unreasonably strong definition.

### 1.4.3 The Final Version of the Definition

In this section we develop a new version of the definition in which the key-exchange protocol can be proven to be secure. And this is the definition of passive security for key-exchange protocol.

Define $\text{Prob}_{\text{key} \leftarrow \text{KEY}(1^\lambda)}[V(\text{key}) = 1] = \gamma$, where $\text{KEY}(1^\lambda)$ is the key space for the key-exchange protocol on parameter $1^\lambda$. And we get a new definition below:

$$\text{Prob}_{\tau \leftarrow \text{trans}_{A,B}(1^\lambda)}[A(\tau) = V(\text{key}(\tau))] \leq \max \{ \gamma, 1 - \gamma \} + \text{negl}(\lambda).$$

By the similar way above, assume $\text{Prob}_{\sigma \leftarrow D_\lambda}[B(\sigma) = 1] \geq \max \{ \gamma, 1 - \gamma \} + \text{nonnegl}(\lambda)$. Now we can compute $\text{Prob}_{\sigma \leftarrow R_\lambda}[B(\sigma) = 1] \leq \max \{ \gamma, 1 - \gamma \}$ as below:

$$\text{Prob}_{\sigma \leftarrow R_\lambda}[B(\sigma) = 1] = \text{Prob}[A(\tau_\sigma) = 1] \cdot \text{Prob}[V(c) = 1] + \text{Prob}[A(\tau_\sigma) = 0] \cdot \text{Prob}[V(c) = 0] = \text{Prob}[A(\tau_\sigma) = 1] \cdot \gamma + \text{Prob}[A(\tau_\sigma) = 0] \cdot (1 - \gamma) \leq \text{Prob}[A(\tau_\sigma) = 1] \cdot \max \{ \gamma, 1 - \gamma \} + \text{Prob}[A(\tau_\sigma) = 0] \cdot \max \{ \gamma, 1 - \gamma \} = (\text{Prob}[A(\tau_\sigma) = 1] + \text{Prob}[A(\tau_\sigma) = 0]) \cdot \max \{ \gamma, 1 - \gamma \} = \max \{ \gamma, 1 - \gamma \}.$$

So the PPT distinguisher $B$ can obtain non-negligible advantage, which obviously breaks the DDH assumption. In other words, under the DDH assumption, the DH key-exchange protocol we discussed above can be proven in the passive security definition given in this sub-section.