1 Diffie-Hellman Key Exchange Protocol

In 1976, Whitefield Diffie and Martin Hellman published their paper *New Directions in Cryptography*, revolutionizing modern cryptography. Prior to this publication, all significant cryptographic techniques relied on some pre-agreed upon key. In their paper however, Diffie and Hellman proposed a protocol that enabled two parties, having no prior communication, to jointly establish a secret key over an insecure channel. Here we will introduce the concrete key exchange protocol and examine its security in the presence of both passive and active adversaries.

1.1 The Diffie-Hellman Protocol

Figure 1 illustrates the concrete Diffie-Hellman key exchange protocol. To begin, two parties, Alice and Bob, choose the values x_A and x_B respectively. These can be determined using the coin flipping techniques discussed in Section **??**. Neither party discloses their value to the other.

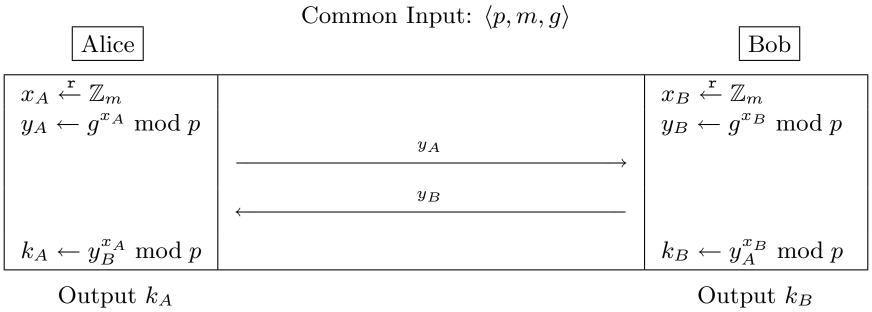
Common Input: $\langle p, m, g\rangle$
Alice
Bob
$x_A \overset{r}{\leftarrow} \mathbb{Z}_m$
$y_A \leftarrow g^{x_A} \bmod p$
$x_B \overset{r}{\leftarrow} \mathbb{Z}_m$
$y_B \leftarrow g^{x_B} \bmod p$
$\xrightarrow{y_A}$
$\xleftarrow{y_B}$
$k_A \leftarrow y_B^{x_A} \bmod p$
$k_B \leftarrow y_A^{x_B} \bmod p$
Output k_A
Output k_B

Figure 1: The Diffie-Hellman key exchange protocol, where p is a large prime and g is a generator of the group $\mathbb{Z}_p^*$ of order m.

The notation $x \overset{r}{\leftarrow} \mathbb{Z}_m$ means that x is sampled according to the uniform over $\mathbb{Z}_m$. Observe that $y_B^{x_A} = y_A^{x_B} \bmod p$, so $k_A = k_B$ and both parties compute the same value in $\mathbb{Z}_p^*$.

In Section **??** we mentioned our interest in the goals, designs, primitives, models, and proofs of cryptography. The goal of a key exchange protocol is to establish a key in the presence of an eavesdropper. Our design of interest is the Diffie-Hellman protocol, whose primitives rely on the protocols for sampling random elements. Continuing with this theme, we now naturally want to know how to model the security of the key exchange protocol and investigate the underlying assumptions required for the Diffie-Hellman key exchange to be provably secure.

1.2 Related Number-Theoretical Problems

Here we introduce several potentially hard number theory problems that allow the Diffie-Hellman protocol to reduce. In the following sections, we examine the proper security definition and reduce the security of the protocol to an appropriate number-theoretical assumption.

Definition 1.2.1. For a suitable cyclic group $G = \langle g\rangle$, take $y \in G$ of order m. The ***discrete logarithm problem*** (DL) is to find an integer $x \in \mathbb{Z}_m$ such that $g^x = y$.

We have no proof that this problem is hard. To the best of our knowledge, the number of steps necessary to find a solution is super-polynomial in the size of the group element, assuming the group is chosen appropriately.

Definition 1.2.2. Given a cyclic group $G = \langle g \rangle$ of order m, g^a and g^b where $a, b \xleftarrow{r} \mathbb{Z}_m$, the ***computational Diffie-Hellman problem*** (CDH) is to compute g^{ab}.

An adversary attacking the Diffie-Hellman protocol does not specifically care about DL. His objective is to solve CDH. It is clear however, that if an adversary could solve DL and derive x from g^x, he could solve CDH with a single exponentiation. This therefore establishes a reduction between the discrete logarithm problem and the computational Diffie-Hellman problem: CDH $\leq$ DL.

Lemma 1.2.1. *The computational Diffie-Hellman problem is no harder than the discrete logarithm problem.*

It is unknown if the converse holds.

Definition 1.2.3. The ***decisional Diffie-Hellman problem*** (DDH) is as follows: given a group $G = \langle g \rangle$ of order m and g^a, g^b, g^c, where $a, b, c \xleftarrow{r} \mathbb{Z}_m$, decide if $c = ab$ or $c \xleftarrow{r} \mathbb{Z}_m$.

This is a very weak problem since it only asks an adversary to determine whether or not c is randomly generated. If an adversary could solve CDH, he could solve DDH by computing g^{ab} and comparing it to g^c; thus, DDH $\leq$ CDH.

Lemma 1.2.2. *The decisional Diffie-Hellman problem is no harder than the computational Diffie-Hellman problem.*

Moreover, this last problem is no harder than the discrete logarithm problem.

So far we have been conveniently vague in our choice of a group; in fact, we have carefully chosen our parameters to ensure that the underlying problems are indeed hard. The next example demonstrates this by showing that the discrete logarithm problem is solvable in polynomial-time when we choose an inappropriate group.

Example. Consider $\mathbb{Z}_p^*$ for a large prime p. By a theorem of Euler, $\mathbb{Z}_p^*$ has order $p-1$. For this example, consider the case where $p-1$ factors into small primes q_i: $p - 1 = q_1 q_2 \cdots q_s$. Then there is a subgroup G_i of order q_i.[1] Define the group homomorphism $f_i \colon \mathbb{Z}_p^* \longrightarrow G_i$ by $x \mapsto x^{p-1/q_i}$ and let $g_i = g^{p-1/q_i}$ for some fixed generator g of $\mathbb{Z}_p^*$. Note that g_i has order q_i.

Take some $y = g^x \bmod p$. Raising both sides to the $p-1/q_i$ power, we have $y^{p-1/q_i} \equiv (g^{p-1/q_i})^x \equiv g_i^{x \bmod q_i} \bmod p$ where $1 \leq i \leq s$. Because q_i is a small prime, we can use ***brute force*** to solve the discrete logarithm problem; that is, we can perform an exhaustive search to find the set of congruences $x_i \equiv x \bmod q_i$. We can then compute x using the Chinese Remainder Theorem.

To avoid this type of attack, we can select $\mathbb{Z}_p^*$ such that it contains a large subgroup. For example, if $p = 2q + 1$ and q is prime, there is a subgroup of size q, called the quadratic residue of $\mathbb{Z}_p^*$.

Definition 1.2.4. The ***quadratic residue*** of G is the subgroup of all $y \in G$ such that there is an $x \in G$ with $x^2 = y$.

When $G = \mathbb{Z}_n^*$, we write the quadratic residue as $QR(n)$. In the particular case $G = \mathbb{Z}_p^*$ for a prime p, $QR(p) = \langle g^2 \rangle$ for a generator g of G. $QR(p)$ is exactly half the elements of G. This is the largest proper subgroup of $\mathbb{Z}_p^*$.

The mapping $x \mapsto x^{\frac{p-1}{2}}$ is particularly useful in this context. It is easy to see that the image of the map is $\{1, -1\}$.

We prove the following useful result regarding quadratic residues.

Lemma 1.2.3. *Consider some $a \in \mathbb{Z}$. It holds that $a^{\frac{p-1}{2}} = 1 \bmod p$ if and only if $a \in QR(p)$.*

Proof. For the forward direction, suppose that $a^{\frac{p-1}{2}} = 1 \bmod p$. Let $y = a^{\frac{p+1}{4}} \bmod p$. Then we have

$$y^2 = a^{\frac{p+1}{2}} = a^{\frac{p-1}{2}} \cdot a = a \bmod p$$

[1] The existence of such a subgroup is guaranteed by Cauchy's Theorem.

Given that $y^2 = a \bmod p$ we obtain $a \in QR(p)$.

For the other direction, if $a \in QR(p)$, i.e., we have $y^2 = a \bmod p$ we have that $a^{\frac{p-1}{2}} = y^{p-1} = 1 \bmod p$. ■

Observe that the proof of the lemma provides a way to construct the roots of a quadratic residue modulo p. Indeed, given a the two roots of a modulo p are calculated as $\pm a^{\frac{p+1}{4}} \bmod p$.

1.3 Group Generators

Definition 1.3.1. A ***group generator*** GGen is a probabilistic algorithm that produces a description of a finite group G when given a length λ. At a minimum, the description contains a group element, the group operation, and a group membership test.

Example. Take $\mathbb{Z}_p$ to be our group for some prime p of length λ. GGen returns an element g of order m, where m is some function of λ and p. The group operation is multiplication modulo p, and if an integer is between 0 and $p-1$, it passes the group membership test.

1.4 The Decisional Diffie-Hellman Assumption

Informally, DDH assumes that it is difficult to distinguish between tuples of the form $\langle g, g^a, g^b, g^{ab}\rangle$ and $\langle g, g^a, g^b, g^c\rangle$, where g belongs to a multiplicative group and a, b, and c are randomly chosen exponents.

Definition 1.4.1. The group generator GGen is said to satisfy the ***decisional Diffie-Hellman assumption*** provided the following probability ensembles $\{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\{\mathcal{R}_\lambda\}_{\lambda\in\mathbb{N}}$ are computationally indistinguishable:

$$\mathrm{D}_\lambda := \Big\{\langle G, m, g\rangle \leftarrow \mathsf{GGen}(1^\lambda);\ a, b \xleftarrow{\mathtt{r}} \mathbb{Z}_m \colon (G, m, g^a, g^b, g^{ab})\Big\}$$

$$\mathrm{R}_\lambda := \Big\{\langle G, m, g\rangle \leftarrow \mathsf{GGen}(1^\lambda);\ a, b, c \xleftarrow{\mathtt{r}} \mathbb{Z}_m \colon (G, m, g^a, g^b, g^c)\Big\}$$

where $m = \mathsf{ord}(g)$.

Equivalently, if $\mathcal{A}$ is a statistical test bounded by probabilistic polynomial-time (PPT), it holds that

$$\mathsf{Adv}^{\mathcal{A}}(\lambda) = \left|\underset{\gamma\leftarrow\mathcal{D}_\lambda}{\mathsf{Prob}}[\mathcal{A}(\gamma) = 1] - \underset{\gamma\leftarrow\mathcal{R}_\lambda}{\mathsf{Prob}}[\mathcal{A}(\gamma) = 1]\right|$$

is negligible in λ. $\mathsf{Adv}^{\mathcal{A}}$ is called the ***advantage*** of $\mathcal{A}$.

1.5 Modeling Security against Passive Adversaries

When defining security, it is important to keep in mind the anticipated adversary. In this section, we focus on passive adversaries. A passive adversary eavesdrops on the communication channel and attempts to extract information about the key without interfering. Before we examine the security definitions, we establish some common notation.

Let $\mathtt{trans}_{A,B}(1^\lambda)$ be the distribution of the transcripts of the interactions between two players A and B. In the Diffie-Hellman protocol, the transcript includes the common input and any exchange of information. The common key produced at the end of a transcript τ is denoted $\mathit{key}(\tau)$. Finally, a predicate V is an algorithm whose only outputs are 1 and 0 (`True` and `False`).

Security Model 1

The most obvious security model for any key exchange defines the protocol to be secure if an adversary cannot obtain any part of the key. More specifically, for all PPT adversaries[2] $\mathcal{A}$,

$$\underset{\tau \leftarrow \mathtt{trans}_{A,B}(1^\lambda)}{\mathsf{Prob}}[\mathcal{A}(\tau) = key(\tau)]$$

is a negligible function in λ. Under this model, it is plausible for an adversary to obtain all but a small amount of information about the key; it is therefore inadequate. The number of bits protected by this model can be as few as $\log^2(\lambda)$.

Security Model 2

For all PPT adversaries $\mathcal{A}$ and predicates V, we define a key exchange to be secure if

$$\underset{\tau \leftarrow \mathtt{trans}_{A,B}(1^\lambda)}{\mathsf{Prob}}[\mathcal{A}(\tau) = V(key(\tau))] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

for some negligible function $\mathsf{negl}(\lambda)$. This model is ideal in that, if our protocol is secure, an adversary cannot identify any information about the key space. Unfortunately, this is also unrealistic.

Assume this model does define security and there is a PPT adversary $\mathcal{A}$ capable of breaking the key exchange protocol. Then there is a predicate V such that

$$\underset{\tau \leftarrow \mathtt{trans}_{A,B}(1^\lambda)}{\mathsf{Prob}}[\mathcal{A}(\tau) = V(key(\tau))] \geq \frac{1}{2} + \alpha,$$

where α is nonnegligible. Let $\mathcal{B}$ be a DDH distinguisher such that, given $\gamma = \langle G, m, g, a, b, c\rangle$, $\mathcal{B}$ uses γ to form a transcript $\tau_\gamma = \langle G, m, g, a, b\rangle$. $\mathcal{B}$ then simulates $\mathcal{A}$ on τ_γ to obtain its output S. $\mathcal{B}$ will return 1 if $V(c) = S$ and 0 if $V(c) \neq S$. When c is a random element of the cyclic group G, let $\mathsf{Prob}[V(c) = 1] = \delta$.

1. If $\gamma \leftarrow \mathcal{D}_\lambda$, then $c = key(\tau_\gamma)$ and $\underset{\gamma \leftarrow \mathcal{D}_\lambda}{\mathsf{Prob}}[\mathcal{B}(\gamma) = 1] \geq \frac{1}{2} + \alpha$.

2. If $\gamma \leftarrow \mathcal{R}_\lambda$, then $c \overset{\mathtt{r}}{\leftarrow} G$ and

$$\underset{\gamma \leftarrow \mathcal{R}_\lambda}{\mathsf{Prob}}[\mathcal{B}(\gamma) = 1]$$

$$= \underset{\langle G,m,g,a,b,c\rangle \leftarrow \mathcal{R}_\lambda}{\mathsf{Prob}}[\mathcal{A}(G, m, g, a, b) = V(c)]$$

$$= \text{Prob}[\text{A}(\tau_\gamma) = V(c)]$$

$$= \text{Prob}[\text{A}(\tau_\gamma) = V(c) \mid V(c) = 1] \cdot \mathsf{Prob}[V(c) = 1] + \ldots$$

$$\ldots + \text{Prob}[\text{A}(\tau_\gamma) = V(c) \mid V(c) = 0] \cdot \mathsf{Prob}[V(c) = 0]$$

$$= \text{Prob}[\text{A}(\tau_\gamma) = 1] \cdot \mathsf{Prob}[V(c) = 1] + \mathsf{Prob}[\mathcal{A}(\tau_\gamma) = 0] \cdot \mathsf{Prob}[V(c) = 0]$$

[2] We say adversary to mean any PPT algorithm.

In the special case where $\delta = 1/2$, we see

$$\underset{\gamma \leftarrow \mathcal{R}_\lambda}{\mathsf{Prob}}[\mathcal{B}(\gamma) = 1] = (\mathsf{Prob}[\mathcal{A}(\tau_\gamma) = 1] + \mathsf{Prob}[\mathcal{A}(\tau_\gamma) = 0])\frac{1}{2} = \frac{1}{2}.$$

Looking at the DDH assumption,

$$\mathsf{Adv}^{\mathcal{B}} \geq \left(\frac{1}{2} + \alpha\right) - \frac{1}{2} = \alpha.$$

Because α is nonnegligible, $\mathcal{B}$ can break the DDH assumption when it has $\mathcal{A}$ and $\delta = 1/2$. When $\delta \neq 1/2$ however, it is easy to find a V that the adversary can guess with probability better than $1/2$ (e.g., V can be the "or" of the first two bits of c). As a result, all schemes fail under this unreasonably strong model.

Security Model 3

Finally, we explore a model under which the security of the key exchange protocol can be proven. This will define passive security.

We have to acknowledge that an adversary can distinguish some part of the key, so let

$$\underset{key \leftarrow \mathsf{Key}(1^\lambda)}{\mathsf{Prob}}[V(key) = 1] = \delta,$$

where $\mathsf{Key}(1^\lambda)$ is the key space probability distribution for the protocol with parameter 1^λ (i.e., the random variable $key(\mathtt{trans}_{A,B}(1^\lambda))$. We now define the key exchange protocol is secure provided that

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

Assume

$$\underset{\gamma \leftarrow \mathcal{D}_\lambda}{\mathsf{Prob}}[\mathcal{B}(\gamma) = 1] \geq \max\{\delta, 1-\delta\} + \alpha$$

for nonnegligible α. Using this, we can show $\underset{\gamma \leftarrow \mathcal{R}_\lambda}{\mathsf{Prob}}[\mathcal{B}(\gamma) = 1] \leq \max\{\delta, 1-\delta\}$:

$$\begin{aligned}
&\underset{\gamma \leftarrow \mathcal{R}_\lambda}{\mathsf{Prob}}[\mathcal{B}(\gamma) = 1] \\
&= \text{Prob}[\text{A}(\tau_\gamma) = 1] \cdot \mathsf{Prob}[V(c) = 1] + \mathsf{Prob}[\mathcal{A}(\tau_\gamma) = 0] \cdot \mathsf{Prob}[V(c) = 0] \\
&= \text{Prob}[\text{A}(\tau_\gamma) = 1]\delta + \mathsf{Prob}[\mathcal{A}(\tau_\gamma) = 0](1-\delta) \\
&\leq \mathsf{Prob}[\mathcal{A}(\tau_\gamma) = 1](\max\{\delta, 1-\delta\}) + \mathsf{Prob}[\mathcal{A}(\tau_\gamma) = 0](\max\{\delta, 1-\delta\}) \\
&= (\text{Prob}[\text{A}(\tau_\gamma) = 1] + \mathsf{Prob}[\mathcal{A}(\tau_\gamma) = 0]) \max\{\delta, 1-\delta\} \\
&= \max\{\delta, 1-\delta\}.
\end{aligned}$$

Based on the above, we have proved the following theorem.

Theorem 1.5.1. *If the DDH assumption is true, the Diffie-Hellman key exchange protocol is secure against passive adversaries under Security Model 3.*

1.6 Suitable Group Generators for the DDH Assumption

In this section, we examine the DDH assumption over two groups.

First consider $\langle g \rangle = \mathbb{Z}_p^*$ for a large prime p. This group is potentially a poor choice; in fact, we can construct a PPT algorithm $\mathcal{A}$ as in Figure 2 that breaks the DDH assumption.

```
Algorithm $\mathcal{A}(p, m, g, a, b, c)$
        if $(a^{m/2} = 1 \lor b^{m/2} = 1) \land (c^{m/2} = 1)$
            then output 1
            else output 0
```

Figure 2: A PPT algorithm that breaks the DDH assumption when $\langle g \rangle = \mathbb{Z}_p^*$, $a, b, c \in \langle g \rangle$, and $m = \mathsf{ord}(g)$ is even.

By Euler's Theorem, $\mathbb{Z}_p^*$ has order $m = p - 1$. Since p is odd for all primes greater than 2, m is even for any nontrivial group.

Let $\gamma = \langle p, m, g, a, b, c \rangle$ where $a = g^x, b = g^y$, and $c = g^{xy}$. If x is even, write $x = 2k$ for some $k \in \mathbb{Z}$. Then $a^{m/2} = (g^x)^{m/2} = g^{km} = 1$. If x is odd, write $x = 2j + 1$ for some $j \in \mathbb{Z}$. Then $a^{m/2} = (g^{2j+1})^{m/2} = g^{m/2} = -1$.

The same result holds for g^y depending on if y is even or odd. The parity of xy clearly depends on the parity of x and y, so $c^{m/2} = (g^{xy})^{m/2} = 1$ as long as one of x or y is even. Thus,

$$\underset{\gamma \leftarrow \mathcal{D}}{\mathsf{Prob}}[\mathcal{A}(\gamma) = 1] = \frac{3}{4}.$$

If instead $\gamma \leftarrow \mathcal{R}$, so $c = g^z$ for a randomly chosen z, there is an equal probability that z will be even or odd. So

$$\underset{\gamma \leftarrow \mathcal{R}}{\mathsf{Prob}}[\mathcal{A}(\gamma) = 1] = \frac{3}{8}.$$

Based on this information,

$$\mathsf{Adv}^{\mathcal{A}} = \frac{3}{4} - \frac{3}{8} = \frac{3}{8}.$$

In an ideal situation, both probabilities are close to 1/2, so their difference is negligible. Since $\mathsf{Adv}^{\mathcal{A}} = 3/8$, $\mathcal{A}$ can distinguish between the two tuples. It is therefore ineffective to build a key exchange over $\mathbb{Z}_p^*$.

One group we can build a key exchange over is the quadratic residue $QR(p)$ of $\mathbb{Z}_p^*$. For example, if $p = 2q + 1$ for a prime q, $QR(p)$ has order q. To the best of our knowledge, this is an adequate group. Recall that $QR(p) = \langle g^2 \rangle$ for a generator g of $\mathbb{Z}_p^*$, so $QR(p)$ is a cyclic group of odd order.

1.7 Modified Diffie-Hellman Protocol

Under the DDH assumption, the generated key is a random element from a group whose structure we typically know very little about. This becomes problematic when using the key in cryptographic applications. Here we look at how to extract a random integer from a random group element. This is useful in that we do understand the structure of integers.

One approach is to define a predicate V such that $\mathsf{Prob}_{x \leftarrow \langle g \rangle}[V(x) = 1] = 1/2$. V then defines one unpredictable bit from the adversary's point of view. It is unclear however, how to find even one such predicate. One must completely understand the structure of the group in oder to discern a random bit. Instead, take $p = 2m + 1$ and define the map

$$H\colon \mathbb{Z}_m \longrightarrow QR(p)$$

by $x \mapsto (x+1)^2 \bmod p$. This is a bijection. To show it is injective, assume $H(x) = H(y)$ for some $x, y \in \mathbb{Z}_m$. Then

(x+1)$^2 \equiv (y+1)^2 \bmod p$

(x+1)$^2 - (y+1)^2 \equiv 0 \bmod p$

x$^2 + 2x - 2y - y^2 \equiv 0 \bmod p$

(x-y)(x+y+2) $\equiv 0 \bmod p$.

So either $x - y \equiv 0 \bmod p$ or $x + y + 2 \equiv 0 \bmod p$. Since $x, y \in \mathbb{Z}_m$, we have $0 \leq x, y \leq m - 1$. Then

x+y+2 $\leq 2(m-1) + 2 = 2m$

¡2m+1 $\equiv 0 \bmod p$.

Thus $x + y + 2 \not\equiv 0 \bmod p$, which leaves only $x - y \equiv 0 \bmod p$, or equivalently $x \equiv y \bmod p$. Since $x, y \in \mathbb{Z}_m \subset \mathbb{Z}_p$, it holds that $x = y$, showing H is injective. H is surjective by the following pre-image of any $y \in QR(p)$,

H$^{-1}(y) =$
$$\begin{cases} y^{p+1/4} \bmod p - 1, & \text{if } y^{p+1/4} \bmod p \in \{1, 2, \ldots, m\} \\ p - y^{p+1/4} \bmod p - 1, & \text{otherwise.} \end{cases}$$

Using this, we can modify the key exchange protocol as is seen in Figure 3.

Under the modified Diffie-Hellman key exchange protocol, we can now use the bijection H to pass from a random element from a group whose structure we do not fully understand to a random integer modulo m.

Exercise: We have shown how to derive a random element from $\mathbb{Z}_m$. This enables us to access cryptographic applications requiring a random integer modulo m as a key. Most applications however, necessitate that the key be a bit string. Determine how to extract the longest possible bit string from an integer modulo m.

It is interesting to note that in a λ-bit key, the probability that the least significant bit is 1 is very close to 1/2, while the probability that the most significant bit is 1 can be far from 1/2.

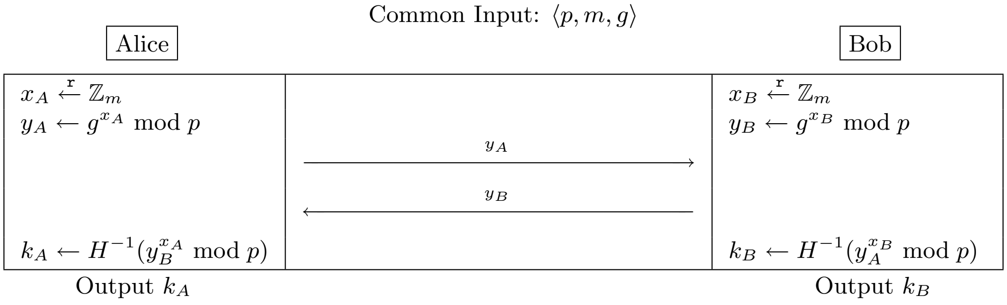
Common Input: $\langle p, m, g \rangle$
Alice
Bob
$x_A \xleftarrow{r} \mathbb{Z}_m$
$y_A \leftarrow g^{x_A} \bmod p$
$x_B \xleftarrow{r} \mathbb{Z}_m$
$y_B \leftarrow g^{x_B} \bmod p$
y_A
y_B
$k_A \leftarrow H^{-1}(y_B^{x_A} \bmod p)$
$k_B \leftarrow H^{-1}(y_A^{x_B} \bmod p)$
Output k_A
Output k_B

Figure 3: The modified Diffie-Hellman key exchange protocol where p is a large prime, g generates the group $QR(p)$ of order m, and $H\colon \mathbb{Z}_m \longrightarrow QR(p)$ by $x \mapsto (x+1)^2 \bmod p$.

1.8 Stronger Adversaries

While the Diffie-Hellman key exchange protocol, as given in Section 1.7, is secure against an eavesdropper, it does not remain so against a more active adversary. In Figure 3, we show the ***man-in-the-middle attack*** in which the adversary, Malorie, participates in the exchange of information between Alice and Bob. The adversary is now the communication channel itself. Malorie can inject messages into the conversation and impersonate the identity of each party to the other. In doing so, Malorie creates two keys, one to share with Alice and one to share with Bob.

This attack exemplifies the need to authenticate and verify authentication on each exchange. Next we introduce a digital signature, which is an important cryptographic primitive, essential in defending against tactics like the man-in-the-middle attack.

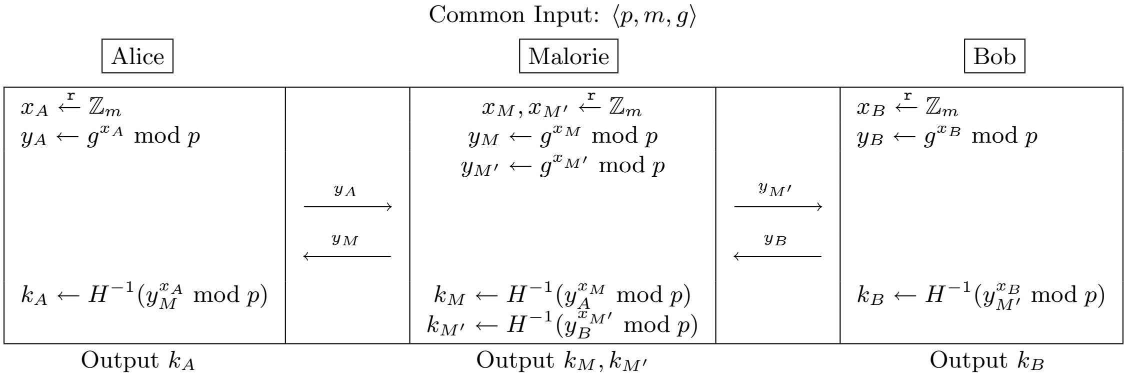
Common Input: $\langle p, m, g \rangle$
Alice
Malorie
Bob
$x_A \xleftarrow{r} \mathbb{Z}_m$
$y_A \leftarrow g^{x_A} \bmod p$
$x_M, x_{M'} \xleftarrow{r} \mathbb{Z}_m$
$y_M \leftarrow g^{x_M} \bmod p$
$y_{M'} \leftarrow g^{x_{M'}} \bmod p$
$x_B \xleftarrow{r} \mathbb{Z}_m$
$y_B \leftarrow g^{x_B} \bmod p$
y_A
y_M
$y_{M'}$
y_B
$k_A \leftarrow H^{-1}(y_M^{x_A} \bmod p)$
$k_M \leftarrow H^{-1}(y_A^{x_M} \bmod p)$
$k_{M'} \leftarrow H^{-1}(y_B^{x_{M'}} \bmod p)$
$k_B \leftarrow H^{-1}(y_{M'}^{x_B} \bmod p)$
Output k_A
Output $k_M, k_{M'}$
Output k_B

Figure 4: The "man-in-the-middle" attack on the Diffie-Hellman key exchange protocol.