14 Diffie-Hellman Key Agreement

14.1 Cyclic Groups

Definition 14.1 *Let $g \in \mathbb{Z}_n^*$. Define $\langle g \rangle_n = \{g^i \,\%\, n \mid i \in \mathbb{Z}\}$, the set of all powers of g reduced mod n. Then g is called a **generator** of $\langle g \rangle_n$, and $\langle g \rangle_n$ is called the **cyclic group generated by** g **mod** n.*

*If $\langle g \rangle_n = \mathbb{Z}_n^*$, then we say that g is a **primitive root mod** n.*

The definition allows the generator g to be raised to a *negative* integer. Since $g \in \mathbb{Z}_n^*$, it is guaranteed that g has a multiplicative inverse mod n, which we can call g^{-1}. Then g^{-i} can be defined as $g^{-i} \stackrel{\text{def}}{=} (g^{-1})^i$. All of the usual laws of exponents hold with respect to this definition of negative exponents.

Example *Taking $n = 13$, we have:*

$$\langle 1 \rangle_{13} = \{1\}$$
$$\langle 2 \rangle_{13} = \{1, 2, 4, 8, 3, 6, 12, 11, 9, 5, 10, 7\} = \mathbb{Z}_{13}^*$$
$$\langle 3 \rangle_{13} = \{1, 3, 9\}$$

Thus 2 is a primitive root modulo 13. Each of the groups $\{1\}$, $\mathbb{Z}_{13}^*$, $\{1, 3, 9\}$ is a cyclic group under multiplication mod 13.

A cyclic group may have more than one generator, for example:

$$\langle 3 \rangle_{13} = \langle 9 \rangle_{13} = \{1, 3, 9\}$$

Similarly, there are four primitive roots modulo 13 (equivalently, $\mathbb{Z}_{13}^*$ has four different generators); they are 2, 6, 7, and 11.

Not every integer has a primitive root. For example, there is no primitive root modulo 15. However, when p is a prime, there is always a primitive root modulo p (and so $\mathbb{Z}_p^*$ is a cyclic group).

Let us write $\mathbb{G} = \langle g \rangle = \{g^i \mid i \in \mathbb{Z}\}$ to denote an unspecified cyclic group generated by g. The defining property of $\mathbb{G}$ is that each of its elements can be written as a power of g. From this we can conclude that:

- Any cyclic group is closed under multiplication. That is, take any $X, Y \in \mathbb{G}$; then it must be possible to write $X = g^x$ and $Y = g^y$ for some integers x, y. Using the multiplication operation of $\mathbb{G}$, the product is $XY = g^{x+y}$, which is also in $\mathbb{G}$.
- Any cyclic group is closed under inverses. Take any $X \in \mathbb{G}$; then it must be possible to write $X = g^x$ for some integer x. We can then see that $g^{-x} \in \mathbb{G}$ by definition, and $g^{-x}X = g^{-x+x} = g^0$ is the identity element. So X has a multiplicative inverse (g^{-x}) in $\mathbb{G}$.

These facts demonstrate that $\mathbb{G}$ is indeed a *group* in the terminology of abstract algebra.

Discrete Logarithms

It is typically easy to compute the value of g^x in a cyclic group, given g and x. For example, when using a cyclic group of the form $\mathbb{Z}_n^*$, we can easily compute the modular exponentiation $g^x \% n$ using repeated squaring.

The inverse operation in a cyclic group is called the discrete logarithm problem:

Definition 14.2 (Discrete Log) *The **discrete logarithm problem** is: given $X \in \langle g \rangle$, determine a number x such that $g^x = X$. Here the exponentiation is with respect to the multiplication operation in $\mathbb{G} = \langle g \rangle$.*

The discrete logarithm problem is conjectured to hard (that is, no polynomial-time algorithm exists for the problem) in certain kinds of cyclic groups.

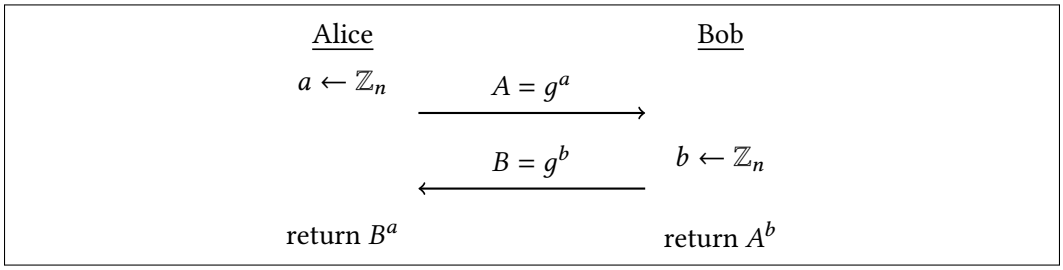
14.2 Diffie-Hellman Key Agreement

Key agreement refers to the problem of establishing a private channel using public communication. Suppose Alice & Bob have never spoken before and have no shared secrets. By exchanging *public* messages (*i.e.*, that can be seen by any external observer), they would like to establish a secret that is known only to the two of them.

The **Diffie-Hellman** protocol is such a key-agreement protocol, and it was the first published instance of public-key cryptography:

Construction 14.3 (Diffie-Hellman) *Both parties agree (publicly) on a cyclic group $\mathbb{G}$ with generator g. Let $n = |\mathbb{G}|$. All exponentiations are with respect to the group operation in $\mathbb{G}$.*

1. *Alice chooses $a \leftarrow \mathbb{Z}_n$. She sends $A = g^a$ to Bob.*
2. *Bob chooses $b \leftarrow \mathbb{Z}_n$. He sends $B = g^b$ to Alice.*
3. *Bob locally outputs $K := A^b$. Alice locally outputs $K := B^a$.*



By substituting and applying standard rules of exponents, we see that both parties output a common value, namely $K = g^{ab} \in \mathbb{G}$.

Defining Security for Key Agreement

Executing a key agreement protocol leaves two artifacts behind. First, we have the collection of messages that are exchanged between the two parties. We call this collection a **transcript.** We envision two parties executing a key agreement protocol in the presence of an *eavesdropper,* and hence we imagine that the transcript is public. Second, we have the **key** that is output by the parties, which is private.

To define security of key agreement, we would like to require that the transcript leaks no (useful) information to the eavesdropper about the key. There are a few ways to approach the definition:

- We could require that it is hard to compute the key given the transcript. However, this turns out to be a rather weak definition. For example, it does not rule out the possibility that an eavesdropper could guess the *first half* of the bits of the key.
- We could require that the key is *pseudorandom* given the transcript. This is a better definition, and the one we use. To formalize this idea, we define two libraries. In both libraries the adversary / calling program can obtain the transcript of an execution of the key agreement protocol. In one library the adversary obtains the key that resulted from the protocol execution, while in the other library the adversary obtains a totally unrelated key (chosen uniformly from the set $\Sigma.\mathcal{K}$ of possible keys).

Definition 14.4 (KA security)

*Let Σ be a key-agreement protocol. We write $\Sigma.\mathcal{K}$ for the keyspace of the protocol (*i.e.*, the set of possible keys it produces). We write $(t,K) \leftarrow \textsc{execprot}(\Sigma)$ to denote the process of executing the protocol between two honest parties, where t denotes the resulting transcript, and K is resulting key. Note that this process is randomized, and that K is presumably correlated to t.*

*We say that Σ is **secure** if $\mathcal{L}^{\Sigma}_{\text{ka-real}} \approx\!\!\approx \mathcal{L}^{\Sigma}_{\text{ka-rand}}$, where:*

$\mathcal{L}^{\Sigma}_{\text{ka-real}}$

```
QUERY():
  (t,K) ← EXECPROT(Σ)
  return (t,K)
```

$\mathcal{L}^{\Sigma}_{\text{ka-rand}}$

```
QUERY():
  (t,K) ← EXECPROT(Σ)
  K' ← Σ.𝒦
  return (t,K')
```

14.3 Decisional Diffie-Hellman Problem

The Diffie Hellman protocol is parameterized by the choice of cyclic group $\mathbb{G}$ (and generator g). Transcripts in the protocol consist of (g^a, g^b), where a and b are chosen uniformly. The key corresponding to such a transcript is g^{ab}. The set of possible keys is the cyclic group $\mathbb{G}$.

Let us substitute the details of the Diffie-Hellman protocol into the KA security libraries. After simplifying, we see that the security of the Diffie Hellman protocol is equivalent to the following statement:

$\mathcal{L}^{\mathbb{G}}_{\text{dh-real}}$

```
QUERY():
  a,b ← ℤ_n
  return (g^a, g^b, g^ab)
```

$\approx\!\!\approx$

$\mathcal{L}^{\mathbb{G}}_{\text{dh-rand}}$

```
QUERY():
  a,b,c ← ℤ_n
  return (g^a, g^b, g^c)
```

We have renamed the libraries to $\mathcal{L}_{\text{dh-real}}$ and $\mathcal{L}_{\text{dh-rand}}$. In $\mathcal{L}_{\text{dh-real}}$ the response to QUERY corresponds to a DHKA transcript (g^a, g^b) along with the corresponding "correct" key

g^{ab}. The response in $\mathcal{L}_{\text{dh-rand}}$ corresponds to a DHKA transcript along with a completely independent random key g^c.

Definition 14.5 (DDH) *The **decisional Diffie-Hellman (DDH) assumption** in a cyclic group $\mathbb{G}$ is that $\mathcal{L}^{\mathbb{G}}_{\text{dh-real}} \approx \mathcal{L}^{\mathbb{G}}_{\text{dh-rand}}$ (libraries defined above).*

Since we have defined the DDH assumption by simply renaming the security definition for DHKA, we immediately have:

Claim 14.6 *The DHKA protocol is a secure KA protocol **if and only if** the DDH assumption is true for the choice of $\mathbb{G}$ used in the protocol.*

For Which Groups does the DDH Assumption Hold?

So far our only example of a cyclic group is $\mathbb{Z}^*_p$, where p is a prime. Although *many* textbooks describe DHKA in terms of this cyclic group, it is not a good choice because the DDH assumption is *demonstrably false* in $\mathbb{Z}^*_p$. To see why, we introduce a new concept:

Claim 14.7 (Euler criterion) *If p is a prime and $X = g^x \in \mathbb{Z}^*_p$, then $X^{\frac{p-1}{2}} \equiv_p (-1)^x$.*

Note that $(-1)^x$ is 1 if x is even and -1 if x is odd. So, while in general it is hard to determine x given g^x, Euler's criterion says that it is possible to determine the *parity of x* (*i.e.*, whether x is even or odd) given g^x.

To see how these observations lead to an attack against the Diffie-Hellman protocol, consider the following attack:

$\mathcal{A}$:
$(A, B, C) \leftarrow$ QUERY()
return $1 \stackrel{?}{\equiv}_p C^{\frac{p-1}{2}}$

Roughly speaking, the adversary returns true whenever C can be written as g raised to an *even* exponent. When linked to $\mathcal{L}_{\text{dh-real}}$, $C = g^{ab}$ where a and b are chosen uniformly. Hence ab will be even with probability 3/4. When linked to $\mathcal{L}_{\text{dh-rand}}$, $C = g^c$ for an independent random c. So c is even only with probability 1/2. Hence the adversary distinguishes the libraries with advantage 1/4.

Concretely, with this choice of group, the key g^{ab} will never be uniformly distributed. See the exercises for a slightly better attack which correlates the key to the transcript.

Quadratic Residues. Several better choices of cyclic groups have been proposed in the literature. Arguably the simplest one is based on the following definition:

Definition 14.8 *A number $X \in \mathbb{Z}^*_n$ is a **quadratic residue modulo** n if there exists some integer Y such that $Y^2 \equiv_n X$. That is, if X can be obtained by squaring a number mod n. Let $\mathbb{QR}^*_n \subseteq \mathbb{Z}^*_n$ denote the set of quadratic residues mod n.*

For our purposes it is enough to know that, when p is prime, $\mathbb{QR}^*_p$ is a cyclic group with $(p-1)/2$ elements (see the exercises). When both p and $(p-1)/2$ are prime, we call p a **safe prime** (and call $(p-1)/2$ a *Sophie Germain prime*). To the best of our knowledge the DDH assumption is true in $\mathbb{QR}^*_p$ when p is a safe prime.

Exercises

14.1. Let p be an odd prime, as usual. Recall that $\mathbb{QR}_p^*$ is the set of quadratic residues mod p — that is, $\mathbb{QR}_p^* = \{x \in \mathbb{Z}_p^* \mid \exists y : x \equiv_p y^2\}$. Show that if g is a primitive root of $\mathbb{Z}_p^*$ then $\langle g^2 \rangle = \mathbb{QR}_p^*$.

Note: This means that $g^a \in \mathbb{QR}_p^*$ if and only if a is even — and in particular, the choice of generator g doesn't matter.

14.2. Suppose $N = pq$ where p and q are distinct primes. Show that $|\mathbb{QR}_N^*| = |\mathbb{QR}_p^*| \cdot |\mathbb{QR}_q^*|$.

Hint: Chinese remainder theorem.

14.3. Suppose you are given $X \in \langle g \rangle$. You are allowed to choose any $X' \neq X$ and learn the discrete log of X' (with respect to base g). Show that you can use this ability to learn the discrete log of X.

14.4. Let $\langle g \rangle$ be a cyclic group with n elements and generator g. Show that for all integers a, it is true that $g^a = g^{a\%n}$.

Note: As a result, $\langle g \rangle$ is isomorphic to the additive group $\mathbb{Z}_n$.

14.5. Let g be a primitive root of $\mathbb{Z}_n^*$. Recall that $\mathbb{Z}_n^*$ has $\phi(n)$ elements. Show that g^a is a primitive root of $\mathbb{Z}_n^*$ if and only if $\gcd(a, \phi(n)) = 1$.

Note: It follows that, for every n, there are either 0 or $\phi(\phi(n))$ primitive roots mod n.

14.6. Let $\langle g \rangle$ be a cyclic group with n elements. Show that for all $x, y \in \langle g \rangle$, it is true that $x^n = y^n$.

Hint: every $x \in \langle g \rangle$ can be written as $x = g^a$ for some appropriate a. What is $(g^a)^n$?

14.7. (a) Prove the following variant of Lemma 4.9: Suppose you fix a value $x \in \mathbb{Z}_N$. Then when sampling $q = \sqrt{2N}$ values $r_1, \ldots, r_q$ uniformly from $\mathbb{Z}_N$, with probability at least 0.6 there exist $i \neq j$ with $r_i \equiv_N r_j + x$.

Hint: This is extremely similar to Exercise **??**.

(b) Let g be a primitive root of $\mathbb{Z}_p^*$ (for some prime p). Consider the problem of computing the discrete log of $X \in \mathbb{Z}_p^*$ with respect to g — that is, finding x such that $X \equiv_p g^x$. Argue that if one can find integers r and s such that $g^r \equiv_p X \cdot g^s$ then one can compute the discrete log of X.

(c) Combine the above two observations to describe a $O(\sqrt{p})$-time algorithm for the discrete logarithm problem in $\mathbb{Z}_p^*$.

14.8. In an execution of DHKA, the eavesdropper observes the following values:

$p = 461733370363$	$A = 114088419126$
$g = 2$	$B = 276312808197$

What will be Alice & Bob's shared key?

14.9. Explain what is wrong in the following argument:

In Diffie-Hellman key agreement, Alice sends $A = g^a$ and Bob sends $B = g^b$. Their shared key is g^{ab}. To break the scheme, the eavesdropper can simply compute $A \cdot B = (g^a)(g^b) = g^{ab}$.

14.10. Let $\mathbb{G}$ be a cyclic group with n elements and generator g. Consider the following algorithm:

```
RAND(A,B,C):
  r,s,t ← Z_n
  A' := A^t g^r
  B' := B g^s
  C' := C^t B^r A^{st} g^{rs}
  return (A',B',C')
```

Let $DH = \{(g^a, g^b, g^{ab}) \in \mathbb{G}^3 \mid a, b, \in \mathbb{Z}_n\}$.

(a) Suppose $(A,B,C) \in DH$. Show that the output distribution of RAND(A,B,C) is the uniform distribution over DH

(b) Suppose $(A,B,C) \notin DH$. Show that the output distribution of RAND(A,B,C) is the uniform distribution over $\mathbb{G}^3$.

★ (c) Consider the problem of determining whether a given triple (A,B,C) is in the set DH. Suppose you have an algorithm $\mathcal{A}$ that solves this problem on average slightly better than chance. That is:

$$\Pr[\mathcal{A}(A,B,C) = 1] > 0.51 \text{ when } (A,B,C) \text{ chosen uniformly in } DH$$
$$\Pr[\mathcal{A}(A,B,C) = 0] > 0.51 \text{ when } (A,B,C) \text{ chosen uniformly in } \mathbb{G}^3$$

The algorithm $\mathcal{A}$ does not seem very useful if you have a *particular* triple (A,B,C) and you really want to know whether it is in DH. You might have one of the triples for which $\mathcal{A}$ gives the wrong answer, and there's no real way to know.

Show how to construct a randomized algorithm $\mathcal{A}'$ such that: for every $(A,B,C) \in \mathbb{G}^3$:

$$\Pr\left[\mathcal{A}'(A,B,C) = [(A,B,C) \stackrel{?}{\in} DH]\right] > 0.99$$

Here the input A,B,C is fixed and the probability is over the internal randomness in $\mathcal{A}'$. So on *every* possible input, $\mathcal{A}'$ gives a very reliable answer.

to-do *better attack against $\mathbb{Z}_p^*$ instantiation of DHKA*