1 Day 5

1.1 Diffie-Hellman Problem

Fix a large prime p, the finite field $\mathbb{F}_p$, and a primitive root g. As we briefly mentioned before, the *discrete log problem* (DLP) is hard! i.e.

Discrete Logarithm Problem(DLP) Choose a secret $a \bmod p-1$. Even if Eve knows p, g, and $g^a \bmod p$, it is HARD for her to know what a is.

The Computational Diffie-Hellman Problem (CDHP) is simpler, meaning that if you know how to solve DLP you can solve DHP, but it's still hard to solve.

Computational Diffie Hellman Problem(CDHP) Choose secret $a, b \bmod p-1$. Even if Eve knows p, g and $g^a, g^b \bmod p$, it is HARD for her to know what g^{ab} is.

In fact, the original idea of Diffie-Hellman was to generate a secret key in a public channel, to be used in a symmetric cipher. (For example, Advanced Encryption Standard - AES, that we may talk about later in the course.)

1.2 Diffie-Hellman Key Exchange

Alice and Bob wants to use some symmetric ciphers, for example the Vigenere cipher. However, they did not have a chance to agree on a key beforehand. What can they do to decide on a key?

Here is one method, the Diffie-Hellman Key Exchange Process,

- Alice and Bob agree on a large prime p and a primitive root $g \bmod p$.
- Alice chooses a private key a, and Bob chooses a private key b.
- Alice sends Bob $A = g^a \bmod p$, and Bob sends alice $B = g^b \bmod p$.
- Alice now has $B = g^b \bmod p$ and her chosen key a. She can then compute $g^{ab} = B^a \bmod p$. Similarly, Bob can compute $g^{ab} = A^b \bmod p$.
- $K = g^{ab} \bmod p$ would be their shared secret key.

What if Eve intercepts in the middle? All she knows is p, g, $g^a \bmod p$, $g^b \bmod p$. If we believe in the hardness of DHP, then Eve would not be able to find $g^{ab} \bmod p$.

Example 1.1. Alice and Bob agrees to use the prime $p = 541$ and the primitive root $g = 10$. Alice chooses the secret key $a = 5$, and computes that $A = 456 \equiv 10^5 \bmod 541$ and sends it to Bob. Bob chooses the secret key $b = 7$, computes that $B = 156 \equiv 10^7 \bmod 541$, then sends it to Alice.

Bob $A = 456$ and $B = 156$ are sent over public channel, and may be intercepted by Eve.

Once Alice receives Bob's number $B = 156$, she computes $B^a = 156^5 \equiv 193 \bmod 541$. Similarly, Bob receives Alice's number $A = 456$, and computes $A^b = 456^7 \bmod 541 \equiv 193 \bmod 541$. Their shared secret key is now $193 \bmod 541$.

Example 1.2. Alice and Bob agrees to use the prime $p = 941$ and the primitive root $g = 627$. Alice chooses the secret key $a = 347$, and computes that $A = 390 \equiv 627^{347} \bmod 941$ then sends it to Bob. Bob chooses the secret key $b = 781$, computes that $B = 691 \equiv 627^{781} \bmod 941$, then sends it to Alice.

Both $A = 390$ and $B = 691$ are sent over public channel, and may be intercepted.

Once Alice receives Bob's number $B = 691$, she computes $B^a = 691^{347} \equiv 470 \bmod 941$. Similarly, Bob receives Alice's number $A = 390$, and computes $A^b = 390^{781} \equiv 470 \bmod 941$. Their shared secret key is then $470 \bmod 941$.

How may Eve attack such a system?

Brainstorm

Brutally trying g^c for all c takes forever, as long as p is large enough - we have to try all the possibilities $g, g^2, \cdots, g^{p-1} \mod p$, which is infeasible when p is large.

We will go over some algorithms to find the exponent/discrete log next week. To give some context of how large numbers are being used in life, reasonable sizes to use now is

- p is at least 2048 bits.
- g is at least 224 bits.
- a, b are random 256-bit integers.

Capabilities of current computer - As of two years ago, a method called Number field sieve can solve the discrete log problem for a random prime of size 530 bits.

Applications. Diffie-Hellman key exchange is common in modern day cryptography. It shows up in various Internet protocols: SSL/TLS, SSH, IPSec when we need to generate a shared secret key. This in turn is why we can send emails/use credit cards, etc. safely over the web.

1.3 The ElGamal Cryptosystem

How do we turn Diffie-Hellman's idea into a cryptosystem? If we want a symmetric-key algorithm, we saw that Diffie-Hellman key exchange gives you a private key for us to proceed. What about a public-key algorithm?

Egyptian cryptographer ElGamal proposed the following system in 1985. Suppose that Alice wants to send a message to Bob,

- Bob chooses a large prime p, primitive root $g \mod p$, and a private key $a \mod p$. He publishes p, g, and the public key $A = g^a \mod p$.
- Alice wants to send Bob the message m. She would choose a random key k, and compute $c_1 = g^k \mod p$, $c_2 = mA^k \mod p$.
- Bob computes $(c_1^a)^{-1} c_2 \mod p$ to get the message m.

Example 1.3. Bob uses the prime $p = 467$ and the primitive root $g = 2$. He chooses $a = 153$ to be his private key and computes his public key

$$A \equiv g^a \equiv 2^{153} \equiv 224 \mod 467$$

Alice wants to send Bob a message $m = 331$. She would then choose a random key, say $k = 197$, and she computes

$$c_1 \equiv g^k \equiv 2^{197} \equiv 87 \mod 467 \text{ and } c_2 \equiv mA^k \equiv 331 \cdot 224^{197} \equiv 57 \mod 467$$

Alice sends the pair $(c_1, c_2) = (87, 57)$ to Bob.

To decrypt the message, Bob uses his private key $a = 153$ and compute

$$x \equiv c_1^a \equiv 87^{153} \equiv 367 \mod 467 \text{ and } x^{-1} \equiv 14 \mod 467$$

Finally,

$$m \equiv x^{-1} c_2 \equiv 14 \cdot 57 \equiv 331 \mod 467$$

recovering the message.

Example 1.4. Bob uses the prime $p = 1549$ and the primitive root $g = 547$. He chooses $a = 107$ to be his private key and computes his public key

$$A \equiv g^a \equiv 547^{107} \equiv 721 \mod 1549$$

Alice wants to send Bob a message $m = 125$. She would then choose a random key, say $k = 132$, and she computes

$$c_1 \equiv g^k \equiv 547^{132} \equiv 17 \mod 1549 \text{ and } c_2 \equiv mA^k \equiv 125 \cdot 721^{132} \equiv 1272 \mod 467$$

Alice sends the pair $(c_1, c_2) = (17, 1272)$ to Bob.

To decrypt the message, Bob uses his private key $a = 107$ and compute

$$x \equiv c_1^a \equiv 17^{107} \equiv 1522 \mod 1549 \text{ and } x^{-1} \equiv 1090 \mod 1549$$

Finally,

$$m \equiv x^{-1} c_2 \equiv 1090 \cdot 1272 \equiv 125 \mod 1549$$

recovering the message.

Why does this work?

- To Bob: $c_1 = g^k \mod p$, and $c_2 = mA^k = mg^{ak} \mod p$. Thus,
$$(c_1^a)^{-1}c_2 = (g^{ak})^{-1}(mg^{ak}) \equiv m \mod p$$
- To Eve: The information she got were p, g, g^a (Public key) g^k, mg^{ak} (Alice's data). If she can figure out m somehow, then she can compute
$$g^{ak} \equiv m^{-1}(mg^{ak}) \mod p \equiv g^{ak} \mod p$$
which means that she can solve DHP, which we believe to be hard. So again, if we believe that DHP is hard, then Eve cannot figure m out.

1.4 Generalization of Diffie-Hellman

We saw that Diffie-Hellman is (essentially) dependent on the hardness of the discrete logarithm problem, for the group $\mathbb{F}_p^*$. A natural question is, can $\mathbb{F}_p^*$ be replaced by something else?

It turns out that the answer is yes, but your group has to be chosen carefully. For example, the additive cyclic group $\mathbb{Z}/m\mathbb{Z}$ would not work. But there are other mathematical objects where this works, such as $\mathbb{F}_{p^n}^*$ (units of a finite field), $E(\mathbb{F}_p)$ ($\mathbb{F}_p$-points of *elliptic curves*) (or even more generally, *abelian varieties*) where this would work. Again it is based on the hardness of the discrete log problem. We will briefly discuss elliptic curve Diffie-Hellman (or if someone wants to do a project on it) near the end of the class.

References

[1] *Cryptography*, by Simon Rubinstein-Salzedo
[2] *An Introduction to Mathematical Cryptography*, by Hoffstein, Pipher and Silverman