## Today:

1. **Review:** DH key exchange
2. Definition of public key encryption
3. Construction

## Recall: Diffie-Hellman Key Exchange Protocol

A 2048-bit prime number $p$ is chosen and a generator $g \in Z_p^*$

Recall $g$ is a generator if $\{g, g^2, \dots, g^{p-1}\} = \{1,2, \dots, p-1\}$

(We will talk later how $p$ and $g$ are chosen.)

$\boldsymbol{A}$ $\boldsymbol{B}$

Choose at random
$x \in \{1, \dots, p-1\}$

$\xrightarrow{g^x \ mod \ p}$

$\xleftarrow{g^y \ mod \ p}$

Choose at random
y $\in \{1, \dots, p-1\}$

The key is defined by: $K = g^{x \cdot y} \ mod \ p$

**Theorem:** This scheme has **weak** security assuming the **CDH** Assumption.

**CDH Assumption:** For every $PPT$ adversary $A$ there exists a negligible function $\mu$ such that for any $n \in N$, any $n$-bit prime $p$ and generator $g \in Z_p^*$,

$$\Pr[A(g^x, g^y) = g^{xy}] \leq \mu(n)$$

where the probability is over randomly chosen $x, y \leftarrow \{1, \dots, p-1\}$.

**Theorem:** This scheme has **strong** security assuming the (false) **DDH** Assumption.

**DDH Assumption:** For every $PPT$ adversary $A$ there exists a negligible function $\mu$ such that for any $n \in N$, any $n$-bit prime $p$ and generator $g \in Z_p^*$ ,

$$| \Pr[A(g^x, g^y, g^{xy}) = 1] - \Pr[A(g^x, g^y, g^u) = 1]| \leq \frac{1}{2} + \mu(n)$$

where the probability is over randomly chosen $x, y, u \leftarrow \{1, \dots, p-1\}$.

**Remark:** These assumptions can be made w.r.t. any group $G$, not only $Z_p^*$

Why do we believe that the CDH assumption is true?

It is stronger than the well studied Discrete Log (DL) assumption.

**DL Assumption:** The function $f_{p,g}: Z_p^* \to Z_p^*$, defined by

$$f_{p,g}(x) = g^x \, mod \, p,$$

is a OWF.

The best algorithms we have for breaking the CDH assumption is via breaking the DL Assumption.

DDH is known to be broken (only) via subgroup attacks.

**Best known algorithm for DL:** Number Field Sieve.

runs in time roughly $2^{\tilde{O}(\log p)^{1/3}}$

This algorithm is quite complicated and will not be covered in this class. Instead, we will see a simple algorithm that runs in time roughly $\sqrt{p}$.

**Giant-Step Baby-Step (GSBS) algorithm:**

This algorithm works for any group, not only for $Z_p^*$

$\boldsymbol{GSBS(p, g, y)}$:

1. Let $m = \sqrt{p}$.
2. Let $L_1 = \left\{\left(i, g^{i \cdot m} \; mod \; p\right) : i \in \{0,1, \dots, m\}\right\}$
3. Let $L_2 \left\{\left(j, y \cdot g^{-j} \; mod \; p\right) : j \in \{0,1, \dots, m\}\right\}$
4. Find $(i, j, z)$ such that $(i, z) \in L_1$ and $(j, z) \in L_2$

   (Note that $z = g^{i \cdot m} = y \cdot g^{-j}$)
5. Output $x = i \cdot m + j$

**Note:** Inverses can be computed efficiently!

Either by the extended GCD algorithm or by using Fermat's theorem.

**Fermat's theorem:** For any $g \in Z_p^*$ we have $g^{p-1} = 1 \; mod \; p$

Thus, $x^{-1} \; mod \; p = x^{p-2} \; mod \; p$

Discrete Log assumption is broken with quantum computers but is believed to be hard classically.

**Is weak security of key exchange sufficient?**

Note that the key is not random only unpredictable!

For encryption we need our secret key to be random.

We can get by with weak security by using $H(K)$ as the secret key, where $H$ is a hash function.

**DH key exchange where the output is $H(K)$ has strong security in the Random Oracle Model!**

**Remark:** DH key exchange have strong security without using $H$ due to subgroup attacks. Indeed, the DDH assumption is known to be false in $Z_p^*$ (and in other groups on non-prime order).

**The DDH Assumption is believed to be true in prime order groups!**

These are groups with no (non-trivial) subgroups.

**Common groups used in practice:**

Groups of prime order over elliptic curves.

1. DDH Assumption is believed to be true in these groups.
2. No non-trivial attacks: Best known attack is the Giant-Step-Baby-Step.

   This allows us to use shorter keys – 256 bits!

One can also use a prime sub-group of $Z_p^*$

**Idea:** Choose $p$ to be a safe prime; i.e., $p = 2q + 1$.

($q$ is called Sophie Germain prime).

Choose $g \in Z_p^*$ to be any quadratic residue s.t. g $\neq 1$ .

Namely, choose any $x \in Z_p^*$ s.t. $x \notin \{1, -1\}$ and let $g = x^2 \; mod \; p$.

Then $g$ is a generator of the Quadratic Residues subgroup of $Z_p^*$ ,

$$\{x^2 \; mod \; p : x \in Z_p^*\}$$

which is a group of prime order $q$.

The reason is that the order of any subgroup divides the order of the groups.

## Public Key Cryptography:

**Idea:** Key agreement can be used to share a key over an insecure channel and then we can use it to encrypt messages.

This results in an interactive process.

**Idea: Interaction can be replaced with a "public key"!**

Each user will publish their own DH message $g^x \; mod \; p$ as their "public key".

If I want to encrypt a message $m$ to a user with public key $pk = g^x mod\ p$, I will simply choose a random $y \leftarrow \{1, \dots, p-1\}$ and send $g^y\ mod\ p$ together with $H(g^{xy}) \oplus m$.

Namely, I will use the secret $H(g^{xy})$ as a one-time pad.

**Note:** The fact that the first message $g^x$ mod p is reused is not a Problem for security! Namely, seeing many pairs $g^{y_i}, g^{x \cdot y_i}$ (for random $y_1, y_2, \dots$) does not harm security since these could be simulated.

**This scheme is known as El-Gamal encryption scheme!**

**Definition:** A **public key encryption scheme** consists of three PPT algorithms $(Gen, Enc, Dec)$ and a messages space $M$:

- $Gen$ generates a pair $(pk, sk)$.
- $Enc$ takes as input $pk$ and message $m \in M$ and outputs $ct$.
- $Dec$ takes as input $sk$ and $ct$ and outputs $m$

**Correctness:** $\forall m \in M$ and $\forall (pk, sk) \leftarrow Gen$,

$$\Pr\left[Dec\big(sk, Enc(pk, m)\big) = m\right] = 1$$

**CPA security:** For every $m_1, m_2 \in M$

$$\left(pk, Enc(pk, m_1)\right) \approx (pk, Enc(pk, m_2))$$

**Note:** This definition is simpler than the one given in the symmetric key setting since an adversary can generate encryptions of any messages of its choice on his own!

**El-Gamal Encryption Scheme:**

It is associated with public parameters $p, g$ (as in DH key exchange)

$\boldsymbol{Gen}$: Choose at random $x \leftarrow \{1, \dots, p-1\}$ and output

$$(pk, sk) = (g^x \ mod\ p\ , x)$$

$\boldsymbol{Enc(pk, m)}$: Choose at random y $\leftarrow \{1, \dots, p-1\}$ and output

$$ct = (g^y \ mod\ p, H(pk^y \ mod\ p) \oplus m)$$

$\boldsymbol{Dec(sk, ct)}$: Parse $ct = (ct_1, ct_2)$, and output

$$m = H\left(ct_1^{sk} \ mod\ p\right) \oplus ct_2$$

**Security:** follows immediately from the security of the DH key exchange!