Today: Secret sharing

So far: 1. Authenticated encryption in the secret key setting

2. Signature schemes: Authenticity in the public key setting

Goal: Encryption in the public key setting

Intermediate goal: Alice and Bob agree on a secret key without ever meeting (by talking over a public channel)!

The importance of key agreement:

A key agreement protocol implies a public key encryption scheme! (Stay tuned...)

Definition: A key agreement protocol is a 2-message protocol between two parties, Alice and Bob, Defined via efficient functions F_1, F_2, G_1, G_2:

A		B
Choose r_1, $msg_1 = F_1(r_1)$	$\xrightarrow{msg_1}$	
		Choose r_2, $msg_2 = F(r_2, msg_2)$
	$\xleftarrow{msg_2}$	

The key defined by:

$$K = G_1(r_1, msg_1, msg_2) = G_2(r_2, msg_1, msg_2)$$

should satisfy the following security guarantee:

1. Strong security against passive attack:

Given (msg_1, msg_2), K should look random!

attacker can only listen, cannot modify!

2. Weak security against passive attack:

Given (msg_1, msg_2), it should be hard to find K,

except with negl probability.

Why are we limitting to passive attacks?

Can add a signature on top to ensure authenticity

(assuming Alice and Bob know each other's public keys).

If Alice and Bob sign their messages then the key agreement protocol

becomes secure against active attacks,

where the adversary is allowed to tamper with the messages.

Does there exist a (weak or strong) secure key agreement protocol?

Note: Any (weak or strong) secure key agreement protocol must rely on hardness assumptions!

An all powerful adversary can invert F_1, and find r_1 s.t. $F_1(r_1) = msg_1$,

and then deduce that $K = G_1(r_1, msg_1, msg_2)$

We do not know how to construct a key agreement protocol based on assumptions

such as hash functions or block ciphers!

We only know of constructions based on algebra and number theory.

$$\Downarrow$$

Less efficient!

Diffie-Hellman Key Exchange (1976):

The precursor to public key encryption

Simplified version: Weak security

Let p be a large prime of 2048 bits!

Let g be a random element in $\{1,\ldots,p\}$

A B

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

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

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

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

$$K = g^{x \cdot y} \bmod p$$

Question1: Can Alice and Bob execute this protocol efficiently?

Seems like they each need to do p multiplications -- too much!!

Answer: Yes! Compute g^x mod p efficiently by repeated squaring:

1. Comute $g_2 = g^2$ mod p

2. Compute $g_3 = g_2^2$ mod p

3. Compute $g_4 = g_3^2$ mod p....

Output $g^{x_1} \cdot g_2^{x_2} \cdot g_3^{x_3} \cdots g_n^{x_n}$ mod p, where $x = x_n \ldots x_3 x_2 x_1$.

Question2: Is this scheme secure??

Computational Diffie Hellman (CDH) Assumption:

Given g, g^x mod p, and g^y mod p, it is hard to predict $g^{x \cdot y}$ mod p,

assuming g, x, y are randomly chosen in $\{1,\ldots,p-1\}$.

Theorem: The above key exchange has weak security assuming the CDH assumption.

Remark: g does not need to be randomly chosen,

the only requirement is that the order of g is large,

where order$(g) = \{g^x \bmod p : x \in \{1,2,\ldots,p\}\}$

Why we believe the CDH assumption??

Discrete Log (DL) problem: Given $p, g, g^x \bmod p$, output x

Discrete Log (DL) Assumption: $f_{pg}(x) = g^x \bmod p$ is a one-way function:

1. Easy to compute (via repeated squaring).
2. Hard to invert.

Note: DL problem is harder than CDH problem.

If CDH assumption is true then DL Assumption is true!

There have been many attempts to try to break the discrete log assumption.

Best know alg: Number field sieve. Runs in time roughly $e^{\tilde{O}(\log p)^{1/3}}$

Giant-step Baby-step (GSBS) alg:

Runs in time roughly $p^{1/2}$.

Works for any group, not only $\mathbb{Z}_p^*$ (multiplication mod p).

GSBS(p,g,y):

1. Let $m = p$.
2. Let $L_1 = \{(i, g^{im} : i \in \{0,1,\ldots,m-1\}\}$
3. Let $L_2 = \{(j, y \cdot g^{-j} : j \in \{0,1,\ldots,m-1\}\}$

Inverses can be computed efficiently mod p! (Extended GCD algorithm)

4. Find (i,j,z) such that $(i,\underset{\substack{\parallel \\ g^{im}}}{z}) \in L_1$ and $(j,\underset{\substack{\parallel \\ y \cdot g^{-j}}}{z}) \in L_2$
5. Output $x = im+j$.

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

What about CDH??

Best known attacks for CDH is via breaking Discrete LOG.

Is weak security of key exchange sufficient? Note the key is not random only unpredictable!

YES! Simply use H(key) as the secret key, where H is a hash function.

Provides strong security in the Random Oracle Model!

The DH key exchage scheme has strong security if we assume the following stronger (but false) assumption:

Decisional Diffie Hellman (DDH) Assumption:

$(g, g^x \bmod p, g^y \bmod p, g^{xy} \bmod p) \cong (g, g^x \bmod p, g^y \bmod p, g^u \bmod p)$,

where x, y, u are randomly chosen in $\{1,\ldots,p-1\}$.

This assumption is false!

The reason is that it is easy to check if an element is a square (quadratic residue) mod p: i.e., if z is of the form $z = x^2 \bmod p$ for some x in $\{1,\ldots,p-1\}$. (Go to 6.875 or 6.857 for details.)

Note that if g is not a quadratic residue then $g^{xy} \bmod p$ is a quadratic residue with probability 3/4, whereas $g^u \bmod p$ is a quadratic residue with probability 1/2.

$\Longrightarrow$ The two are distinguishable

Let's choose g a quadratic residue!

We believe the DDH assumption is true if g is *any* quadratic residue (except 0,1) and p is a safe prime: i.e., $p = 2q+1$ for some prime q (q is called Sophie Germain prime).

The reason is that the the set $\{x^2 \bmod p:\ x \in \{1,\dots,p-1\}\}$ where p is a safe prime, is a group of prime order q (with multiplication mod p).

The fact that it is of prime order eliminates sub-group attacks.

Common group used in practice: Groups of prime order over eliptic curves.

1. DDH Assumption is beleived to be true in these groups!
2. No non-trivial attacks: Best known attacks are Giant-Step Baby-Step!

   This allows us to use shorter keys -- 256 bits!