Recap: p - prime

$\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$ with operations $+_p$: addition mod p

$\times_p$: mult. mod p

Properties:

0 → If $a, b \in \mathbb{Z}_p$, then $a +_p b$, $a \times_p b$ are also in $\mathbb{Z}_p$.

1 → $+_p$, $\times_p$ are commutative, associative

2 → $(a +_p b) \times_p c = a \times_p c +_p b \times_p c$

3 → $\forall\, a \in \mathbb{Z}_p$, $\exists\, b \in \mathbb{Z}_p$ s.t. $a +_p b = 0$ — additive inverse exists

4 → $\forall\, a \in \mathbb{Z}_p \setminus \{0\}$, $\exists\, b \in \mathbb{Z}_p \setminus \{0\}$ s.t. $a \times_p b = 1$ — mult. inverse exists

$\forall\, a \in \mathbb{Z}_p$, $a \times_p 0 = 0$

Fermat's Little Theorem (FLT):

$$\forall\, a \in \mathbb{Z}_p \setminus \{0\}, \quad \underbrace{a \times_p a \times_p \cdots \times_p a}_{p-1 \text{ times}} = 1$$

To prove FLT, we used the following properties of $(\mathbb{Z}_p, +_p, \times_p)$:

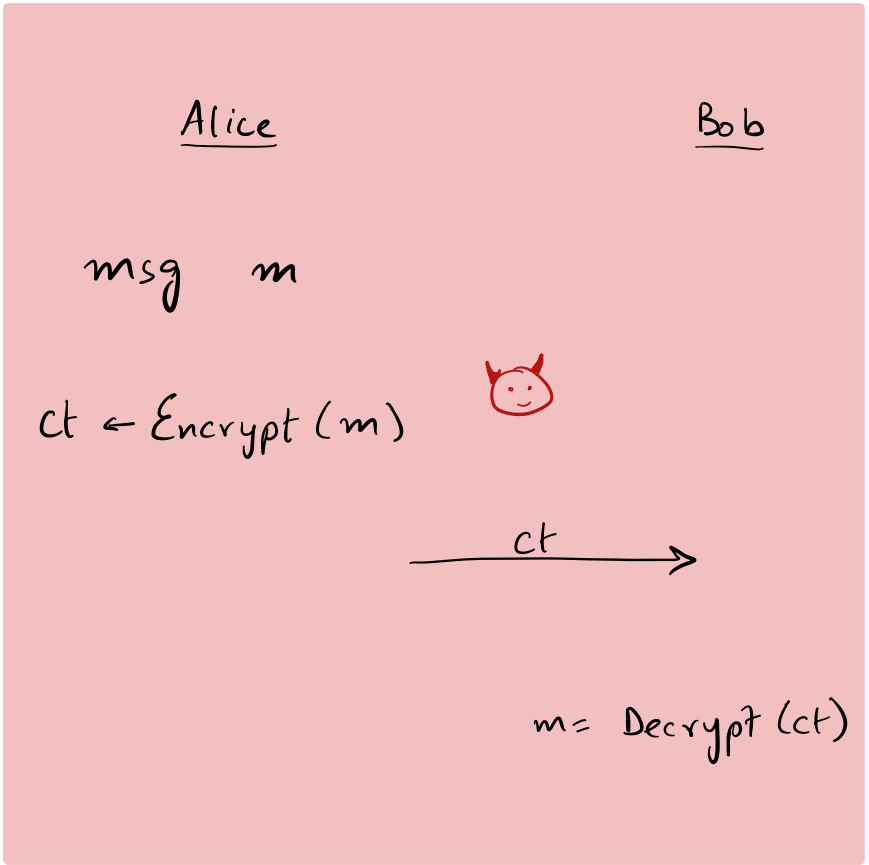
- $\times_p$ is associative and commutative
- for all $a, b \in \mathbb{Z}_p$, $a \times_p b$ is also in $\mathbb{Z}_p$.
- $|\mathbb{Z}_p \setminus \{0\}| = p-1$
- mult. inverse exists for all non-zero elements

Useful to abstract out the properties used. Results can be translated to other settings where these properties hold.

LECTURE 11 : PUBLIC KEY ENCRYPTION

Encryption : vital for secure communication

Scenario : Alice and Bob want to communicate over an insecure channel. Alice has a message m that she wants to send to Bob. However, there is an eavesdropping adversary who is watching every bit that passes through the insecure channel. Therefore, Alice must Encrypt the message, producing a ciphertext. Bob must Decrypt the ciphertext to recover the message.



Encrypt and Decrypt are efficient algorithms.

Kerchkoff Principle : We will always assume that the adversary knows the algorithms being used.

Hiding description of algorithms is hard.

However, if adversary also knows the decryption algorithm, then it can recover the message, just like Bob.

→ There must be some secret information that Bob knows, which the adversary does not know.

There are two kinds of encryption possible:

- Secret key encryption (aka symmetric key enc.)
- Public key encryption.

Let us first consider secret key encryption (SKE). SKE has been used for hundreds of years now.

$\mathcal{K}$: key space. Same key used for encryption and decryption.

Encrypt $(k, m) \rightarrow ct$

Decrypt $(k, ct) \rightarrow m$

Correctness : $\forall$ messages m, $\forall\ k \in \mathcal{K}$,

Decrypt $(k,$ Encrypt $(k, m)) = m$.

Security: If adversary does not know the key, then given the ciphertext, adversary does not learn anything about the underlying message.

This intuition for security can be formalised. However, we will not see the formalisation in this course. Discuss with me offline, or take COL759 in one of the upcoming semesters.

How did people use SKE:

1. Alice and Bob must meet and sample a common secret key k.

2. Alice can then use k to encrypt, and Bob can decrypt using k.

(1) is a huge limitation of SKE. What if Alice and Bob cannot meet physically to sample the common secret key k?

In 1960s and 70s, researchers started realizing that we need something more than SKE for Internet to reach the masses.

The critical object that we require is PUBLIC KEY CRYPTOGRAPHY, in particular PUBLIC KEY ENC.

Again, we have an Encrypt algorithm, and a Decrypt algorithm. Encrypt uses a public key pk, and Decrypt uses a secret key sk. Both pk and sk are sampled together using a Setup algorithm.

$$\text{Setup} \rightarrow pk, sk$$
$$\text{Encrypt}(pk, m) \rightarrow ct$$
$$\text{Decrypt}(sk, ct) \rightarrow m.$$

Correctness: For all (pk, sk) sampled by Setup,
For all messages m,
$\text{Decrypt}(sk, \text{Encrypt}(pk, m)) = m.$

Security: For any adversary that only has pk (but not sk), Encrypt(pk, m) hides the message m.

How is PKE used in practice?
Suppose Alice wants to send encrypted messages to Bob. Bob's public key pk_{Bob} is known to everyone, but secret key is only known to Bob. Alice encrypts her message

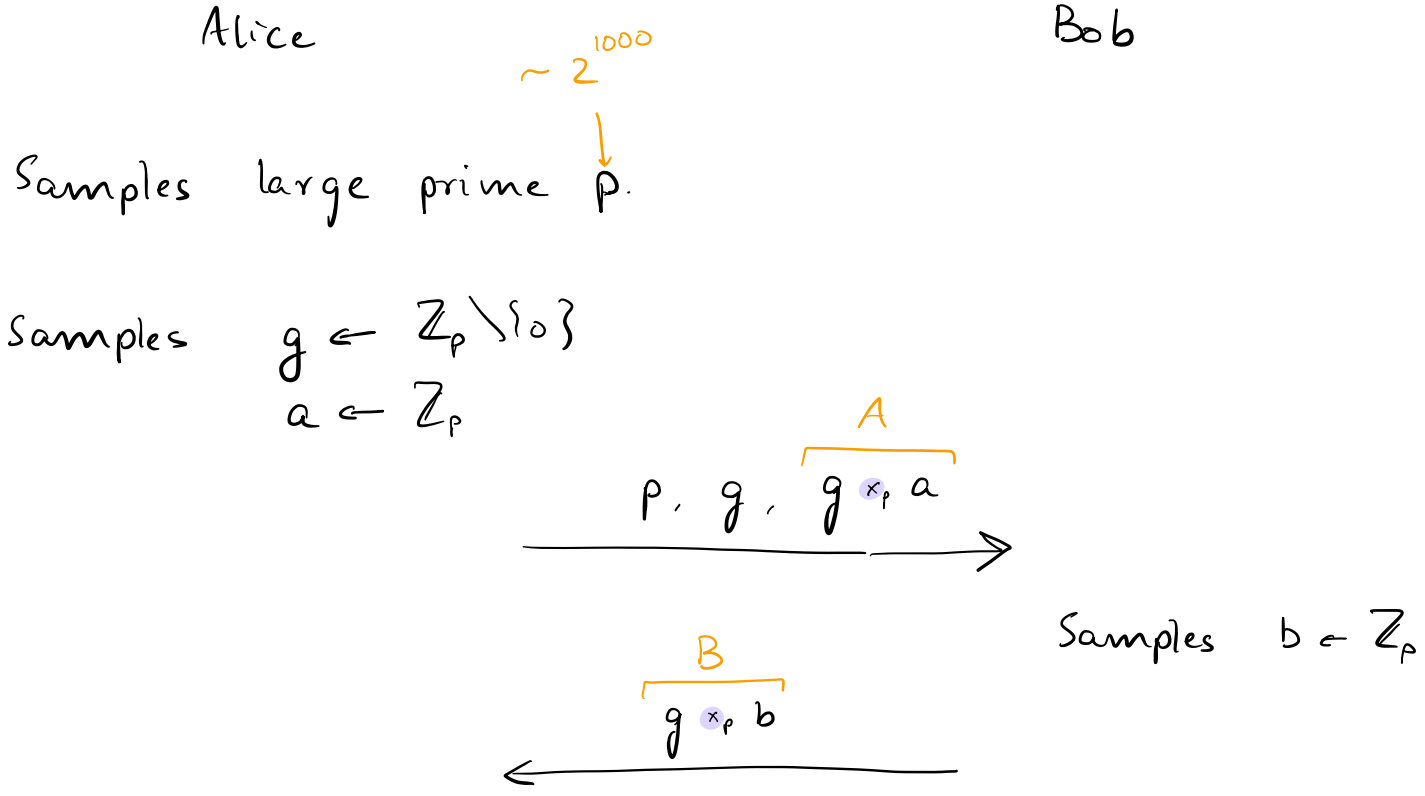
using pk_{Bob}. Adversary can see the ciphertext, it knows pk_{Bob}, but learns no info about the message. Bob can use his secret key to recover the message.

How to build PKE?

Building PKE is far more challenging than building SKE. The notion of PKE was proposed by two researchers at Stanford: Prof. Michael Hellman and his PhD student Whitfield Diffie. While Diffie and Hellman could not build PKE, they gave a secure key exchange protocol, which would suffice for some of the applications. Constructing secure key exchange is also a challenging problem, and until the work of Diffie & Hellman, we didn't have any constructions.

Secure Key Exchange:
Alice and Bob communicate over an insecure channel. Initially, they both don't have any common secrets. They exchange messages over insecure channel. At the end of their communication, Alice and Bob can derive a common key K. However, the adversary (listening to the insecure channel) does not learn anything about K.

Diffie and Hellman proposed a solution using $\mathbb{Z}_p$ arithmetic in 1976. Before seeing [DH 76], let us see a simpler attempt.



Both Alice and Bob have shared key $K = g \times_p a \times_p b$

Adversary sees p, g, A, B (and knows how the protocol works). Unfortunately, it can compute 'a' from A, and therefore compute $K = B \times_p a$.

DIFFIE-HELLMAN KEY EXCHANGE:

finding inverse

[DH 78] observed that while mult. and div. in $\mathbb{Z}_p$ are both efficient, there are some operations that are efficient in one direction, but computationally difficult to reverse.

One such operation is exponentiation.

Given $g \in \mathbb{Z}_p \setminus \{0\}$, $a \in \mathbb{Z}_p$, $\exp_p(g, a)$ can be computed in $O(\text{poly log}(p))$ steps (using repeated squaring).

However, computing DISCRETE LOGARITHM is believed to be hard.

DISCRETE LOG PROBLEM: Given unif. rand. $g \leftarrow \mathbb{Z}_p \setminus \{0\}$ and $A = \exp_p(g, a)$ for random 'a', compute a.

In case there are multiple numbers z s.t. $\exp_p(g, z) = A$, output any one of them.

DIFFIE - HELLMAN KEY EXCHANGE PROTOCOL :

Alice	Bob
Sample large prime p.	
Sample $g \leftarrow \mathbb{Z}_p \setminus \{0\}$, $a \leftarrow \mathbb{Z}_p$	
$A = \exp_p(g, a)$	
$\xrightarrow{p,\ g,\ A}$	
	Sample $b \leftarrow \mathbb{Z}_p$
	$B = \exp_p(g, b)$
$\xleftarrow{B}$	
Computes $\exp_p(B, a)$	Computes $\exp_p(A, b)$
$(g^a)^b = g^{ab} \pmod p$	$(g^b)^a = g^{a \cdot b} \pmod p$

Observation : Alice and Bob compute $K = g^{a \cdot b} \bmod p$.

How to prove that the Diffie-Hellman protocol is secure? There is no proof. Diffie-Hellman conjectured that the following problem is computationally hard:

COMPUTATIONAL DIFFIE HELLMAN PROBLEM (CDH):

Given g sampled uniformly at random from $\mathbb{Z}_p \setminus \{0\}$,
given $A = \exp_p(g, a)$ for uniformly random $a \leftarrow \mathbb{Z}_p$,
given $B = \exp_p(g, b)$ " " " $b \leftarrow \mathbb{Z}_p$,
compute $\exp_p(g, a \cdot b)$

↳ regular mult. of a and b.

CDH is a widely believed conjecture for certain choices of primes. It is used to build many other crypto primitives.

But what about public key encryption? Diffie & Hellman were keenly interested in building PKE, but the first construction came in 1978, given by Ron Rivest, Adi Shamir, Leonard Adleman.

↳ A very popular name in theoretical CS. The secret-sharing solution based on polynomials was also given by Shamir.

This is the most widely used PKE scheme in practice today.

Before seeing the RSA PKE scheme, let us see a simpler attempt that does not work. It is based on the following observation:

Observation: Suppose $a \in \mathbb{Z}_p$, d and e are numbers s.t. $d \cdot e \bmod (p-1) = 1$.

Then $a = \exp_p(a, d \cdot e)$

$= \exp_p(\exp_p(a, e), d)$

Proof: Using Fermat's Little Theorem,

$\exp_p(a, d \cdot e) = \exp_p(a, k \cdot (p-1) + 1)$

$= \exp_p(a, k \cdot (p-1)) \times_p \exp_p(a, 1)$

$= 1 \times_p a$

■

Using this observation, we build a PKE scheme as follows:

pk = (prime p, exponent e)

sk = d s.t. $e \cdot d \bmod p = 1$

Encrypt (pk = (p, e), $a \in \mathbb{Z}_p$) ← message is an element in $\mathbb{Z}_p$

$ct = \exp_p(a, e)$.

Even though prime p is very large $\sim 2^{1000}$, ct can be computed efficiently using repeated squaring.

Decrypt (sk = d , ct) :

Output $\exp_p(ct, d)$.

Correctness follows from the observation above, what about security? Recall, adversary sees pk = (p, e) and ciphertext. It must compute m.

The adversary can easily find 'd' s.t. $e \cdot d \bmod (p-1) = 1$, using Extd. Euclid's Algorithm. As a result, it can find m given ct.

To make this idea work, we need a set S, together with a multiplication operation $\times_S$ where something like FLT holds (that is, $\exp_S(a, |S|) = 1$), but given e, it is hard to find 'd' s.t. $e \cdot d \bmod |S| = 1$.

In particular, if $|S|$ is known, then one can always find such a d if it exists.

Therefore given an implicit description of S, it should be hard to find $|S|$.

For this, we will need to go beyond $\mathbb{Z}_p$.

Consider $\mathbb{Z}_N = \{0, 1, 2, \ldots, N-1\}$. Let $N = p \cdot q$

p, q are primes.

In $\mathbb{Z}_N$, there are certain elements which do not have mult. inverses eg. 0, multiples of p, multiples of q. Remove these elements from $\mathbb{Z}_N$, and let us call the remaining set $\mathbb{Z}_N^*$.

Def 11.1 $\mathbb{Z}_N^* = \{ a \in \mathbb{Z}_N \text{ s.t. } \gcd(a, N) = 1 \}$

$\times_N$: mult. mod N

→ $1 \in \mathbb{Z}_N^*$, $\times_N$ is assoc. and comm.

→ if $a, b \in \mathbb{Z}_N^*$, then $a \times_N b \in \mathbb{Z}_N$

$a \times_N b = a \cdot b - k \cdot N$

Suppose $\gcd(a \cdot b, N) \neq 1$. Then either p or q divides a or b. ⇒ either $a \notin \mathbb{Z}_N^*$ or $b \notin \mathbb{Z}_N^*$.

→ $\forall a \in \mathbb{Z}_N^*$, $\exists b \in \mathbb{Z}_N^*$ s.t. $a \times_N b = 1$. This b is unique.

Since $a \in \mathbb{Z}_N^*$, $\gcd(a, N) = 1$

⇒ $\exists s, t \in \mathbb{Z}$ s.t. $s \cdot a + t \cdot N = 1$ [Bezout's Lemma].

⇒ $s \cdot a \bmod N = 1$ [def. of mod operation]

⇒ $((s \bmod N) \cdot (a \bmod N)) \bmod N = 1$ [property of mod]

⇒ $(s \bmod N) \times_N a = 1$ [def^n. of $\times_N$]

$(\mathbb{Z}_N^*, \times_N)$ behaves like $(\mathbb{Z}_p \setminus \{0\}, \times_p)$

Therefore, an analogue of Fermat's Little Thm. exists.

[Euler's Thm]

Thm 11.1 $\forall\ a \in \mathbb{Z}_N^*,\quad \exp_N(a, |\mathbb{Z}_N^*|) = 1.$

The proof we saw for FLT in Lecture 10 also works for Euler's Theorem.

Note that if you used induction to prove FLT, then that proof does not work for Euler's theorem.

example : $N = 15$ $\quad |\mathbb{Z}_N^*| = 8$