Public key Encryption I

Knapsack cryptography
Diffie-Hellman key Exchange
El Gamal Encryption

Public key Crypto

Message + public key = Ciphertext
Ciphertext + private key = Message

Two keys need to be linked in a mathematical way.
Knowing the public key should tell you nothing about the private key.
**Knapsack Cryptography**

Given a pile of \( n \) items, each with different weights \( w_i \), is it possible to put items in a knapsack such that we get a specific weight \( S \)? \( b_i \in \{0, 1\} \)

\[
S = b_1 w_1 + b_2 w_2 + \ldots + b_n w_n
\]

NP-complete problem in general case.

Super-increasing knapsacks: linear time solvable

\( w_j \geq \sum_{i=1}^{j-1} w_i \) \( \{2, 3, 6, 13, 27, 52\} \)

**Merkle-Hellman Cryptosystem**

Private key \( \rightarrow \) super-increasing knapsack problem

\[ \text{PRIVATE TRANSFORM} \]

Public key \( \leftarrow \) "hard" general knapsack problem

Transform: two private integers \( N, M \) s.t. \( \gcd(N, M) = 1 \)

Multiply all values in the sequence by \( N \) and then mod \( M \)
Merkle-Hellman Example

\[ N = 31, \quad M = 105 \quad \text{private key} = \{2, 3, 6, 13, 27, 52\} \]
\[ \text{public key} = \{62, 93, 81, 88, 102, 37\} \]

Message = \[ 011000 \quad 110101 \quad 101110 \]

Ciphertext:
\[ 011000 \quad 93 + 81 = 174 \]
\[ 110101 \quad 62 + 93 + 88 + 37 = 280 \]
\[ 101110 \quad 62 + 81 + 88 + 102 = 333 \]
\[ = 174, 280, 333 \]

Recipient knows \[ N = 31, \quad M = 105 \quad \{2, 3, 6, 13, 27, 52\} \]
Multiplies each ciphertext block by \[ N^{-1} \pmod{M} \]
\[ N^{-1} = 61 \pmod{105} \]
\[ 174 \cdot 61 = 9 = 3 + 6 = 011000 \]
\[ 280 \cdot 61 = 70 = 2 + 3 + 13 + 52 = 110101 \]
\[ 333 \cdot 61 = 48 = 2 + 6 + 13 + 27 = 101110 \]

Solving super-increasing knapsack

Beautiful but broken

Density of knapsack \[ d = \frac{n}{\max \{\log_2 w_i : 1 \leq i \leq n\}^2} \]

Lattice basis reduction can solve knapsacks of low density. Unfortunately, M-H scheme always produces knapsacks of low density!
**Typical Public Key Setup**

Let $G$ be a group, generator $g \in G$

$$y = g^x \quad 0 \leq x < \text{order}(g)$$

Then $x$ is the discrete log of $y$, base $g$, in $G$

Assume DLP is hard

Note: DLP is easy if $\text{order}(g)$ has only small prime factors

$p = 2r+1$ large "safe" prime, $r$ prime

$g$ generator of $\mathbb{Z}_p^*$

$\text{order}(g) = p-1 = |\mathbb{Z}_p^*|$

$p, g$ public system parameters

Alice picks secret key $x$, $1 \leq x < p-1$

Alice publishes her public key $y = g^x \pmod{p}$

Alice's secret key is protected from disclosure by DLP
**Diffie-Hellman Key Exchange**

- **P, g** public parameters
- **Alice**: secret key $x$, public key $g^x$
- **Bob**: secret key $y$, public key $g^y$

\[ K = (g^y)^x = g^{xy} \quad \text{Eve (passive)} \]
\[ K = (g^x)^y = g^{xy} \]

Require DLP to be hard but not sufficient.

CDH: Computational Diffie Hellman: Given $g^x$ and $g^y$, to compute $g^{xy}$ is hard.

Secure against passive Eve if CDH is hard.

What about active Eve?
EL GAMAL ENCRYPTION

Public key encryption scheme. Assume DLP, CDH are hard.

\[ \mathbb{Z}_p^* \] for large random prime \( p \)

\[ SK = x, \quad 0 \leq x < p - 1 \]

\[ PK = (p, g, g^x) \]

Encryption: Bob does the following

(a) Represent message as integer \( m \in \{0, 1, \ldots, p-1\} \)
(b) Select a random \( k, \quad 1 \leq k < p - 1 \)
(c) \( y = g^k \mod p, \quad s = m \cdot (g^x)^k \mod p \)
(d) Send ciphertext \( c = (y, s) \) to Alice

Decryption: To recover plaintext, Alice does

(a) Compute \( y^{-x} \mod p = y^{p-1-x} \mod p \)
(b) Recover \( m = (y^{-x}) \cdot s \mod p \)
   \[ m \cdot g^{kx} \]
   \[ g^{-kx} \]
Key generation: Entity Alice selects
prime \( p = 2357 \)
generator \( g = 2 \) of \( \mathbb{Z}_{2357}^* \)
Alice chooses private key \( SK = x = 1751 \)
\( g^x \mod p = 2^{1751} \mod 2357 = 1185 \)
Alice’s PK = \((p = 2357, g = 2, g^x = 1185)\)

**Encryption**
Encrypt \( m = 2035 \)
Bob selects a random integer \( k = 1520 \)
and computes
\( y = 2^{1520} \mod 2357 = 1430 \)
and \( S = 2035 \cdot 1185^{1520} \mod 2357 = 697 \)
Bob sends \( y = 1430, S = 697 \) to Alice

**Decryption**
To decrypt Alice computes
\( y^{p-1-x} = 1430^{605} \mod 2357 = 872 \)
and recovers \( m \) by computing
\( m = 872 \cdot 697 \mod 2357 = 2035 \)