Lecture 22: Cryptography: Encryption

- Symmetric key encryption
- Key exchange
- Asymmetric key encryption
- RSA
- NP-complete problems and cryptography
  - graph coloring
  - knapsack

Symmetric key encryption

$$c = e_k(m)$$
$$m = d_k(c)$$

Here c is the *ciphertext*, m is the *plaintext*, e is the encryption function, d is the decryption function and k is the secret key. e, d permute and reverse-permute the space of all messages.

Reversible operations: $\oplus$, $+/-$, shift left/right.

Symmetric algorithms: AES, RC5, DES

Key Management Question

How does secret key k get exchanged/shared?

Alice wants to send a message to Bob. There are pirates between Alice and Bob, that will take any keys or messages in unlocked box(es), but won't touch locked boxes. How can Alice send a message or a key to Bob (without pirates knowing what was sent)?

Solution:

- Alice puts m in box, locks it with k_A
- Box sent to Bob

- Bob locks box with k_B
- Box sent to Alice
- Alice unlocks k_A
- Box sent to Bob
- Bob unlocks k_B, reads m

Notice that this method relied on the commutativity of the locks. This means that the order of the lock and unlock operations doesn't matter.

Diffie-Hellman Key Exchange

$$G = F_p^*$$

Here F_p^* is a finite field (mod p, a prime). $*$ means invertible elements only ($\{1, 2, ..., p-1\}$)

Alice	g public p public	Bob	$2 \leq g \leq p-2$
Select a Compute g^a	$\longrightarrow$	g^a Select b Compute g^b	$1 \leq a, b \leq p-2$
g^b	$\longleftarrow$		

Alice can compute $\left(g^b\right)^a \mod p = k$.
Bob can compute $(g^a)^b \mod p = k$.
Assumes the Discrete Log Problem is hard (given g^a, compute a) and Diffie Hellman Problem is hard (given g^a, g^b compute g^{ab}).
Can we attack this? Man-in-the-middle:

- Alice doesn't know she is communicating with Bob.
- Alice agrees to a key exchange with Eve (thinking she is Bob).
- Bob agrees to a key exchagne with Eve (thinking she is Alice).
- Eve can see all communications.

Public Key Encryption

$$\text{message} + \text{public key} = \text{ciphertext}$$
$$\text{ciphertext} + \text{private key} = \text{message}$$

The two keys need to be linked in a mathematical way. Knowing the public key should tell you nothing about the private key.

RSA

- Alice picks two large secret primes p and q.
- Alice computes $N = p \cdot q$.
- Chooses an encryption exponent e which satisfies $gcd(e, (p-1)(q-1)) = 1$, $e = 3, 17, 65537$.
- Alice's public key= (N, e).
- Decryption exponent obtained using Extended Euclidean Algorithm by Alice such that $e \cdot d \equiv 1 \mod (p-1)(q-1)$.
- Alice private key=(d, p, q) (storing p and q is not absolutely necessary, but we do it for efficiency).

Encryption and Decryption with RSA

$$c = m^e \mod N \qquad \text{encryption}$$
$$m = c^d \mod N \qquad \text{decryption}$$

Why it works

$$\phi = (p-1)(q-1)$$

Since $ed \equiv 1 \mod \phi$ there exists an integer k such that $ed = 1 + k\phi$.
Two cases:
Case 1 $gcd(m, p) = 1$. By Fermat's theorem,

$$m^{p-1} \equiv 1 \mod p$$
$$\left(m^{p-1}\right)^{k(q-1)} \cdot m \equiv m \mod p$$
$$m^{1+k(p-1)(q-1)} = m^{ed} \equiv m \mod p$$

Case 2 $gcd(m, p) = p$. This means that $m \mod p = 0$ and so $m^{ed} \equiv m$

Thus, in both cases, $m^{ed} \equiv m \mod p$. Similarly, $m^{ed} \equiv m \mod q$. Since p, q are distinct primes, $m^{ed} \equiv m \mod N$. So $c^d = (m^e)^d \equiv m \mod N$

Hardness of RSA

- Factoring: given N, hard to factor into p, q.
- RSA Problem: given e such that $gcd(e, (p-1)(q-1)) = 1$ and c, find m such that $m^e \equiv c \mod N$.

NP-completeness

Is N composite with a factor within a range? unknown if NP-complete

Is a graph k-colorable? In other words: can you assign one of k colors to each vertex such that no two vertices connected by an edge share the same color? NP-complete

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 total weight S? NP-complete

NP-completeness and Cryptography

- NP-completeness: about worst-case complexity
- Cryptography: want a problem instance, with suitably chosen parameters that is hard on average

Most knapsack cryptosystems have failed.

Determining if a graph is 3-colorable is NP-complete, but very easy on average. This is because an average graph, beyond a certain size, is not 3-colorable!

Consider a standard backtracking search to determine 3-colorability.

- Order vertices $v_1, ..., v_t$. Colors $= \{1, 2, 3\}$
- Traverse graph in order of vertices.
- On visiting a vertex, choose smallest possible color that "works".
- If you get stuck, backtrack to previous choice, and try next choice.

- Run out of colors for 1^{st} vertex → output 'NO'
- Successfully color last vertex → output 'YES'

On a random graph of t vertices, average number of vertices traveled < 197, regardless of t!

Knapsack Cryptography

General knapsack problem: NP-complete

Super-increasing knapsack: linear time solvable. In this problem, the weights are constrained as follows:

$$w_j \geq \sum_{i=1}^{j-1} w_i$$

Merkle Hellman Cryptosystem

Private key → super-increasing knapsack problem $\xrightarrow{\text{Private transform}}$ "hard" general knapsack problem → public key.

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

Multiply all values in the sequence by N and then take mod M.

Example: $N = 31$, $M = 105$, private key=$\{2, 3, 6, 14, 27, 52\}$, public key=$\{62, 93, 81, 88, 102, 37\}$

Merkle Hellman Example

$$\begin{aligned}
\text{Message} &= 011000 \quad 110101 \quad 101110 \\
\text{Ciphertext:} &011000 \qquad 93 + 81 = 174 \\
&110101 \qquad 62 + 93 + 88 + 37 = 280 \\
&101110 \qquad 62 + 81 + 88 + 102 = 333 \\
&= 174, 280, 333
\end{aligned}$$

Recipient knows $N = 31, M = 105$, $\{2, 3, 6, 14, 27, 52\}$. Multiplies each ciphertext block by $N^{-1} \mod M$. In this case, $N^{-1} = 61 \mod 105$.

$$\begin{aligned}
174 \cdot 61 &= 9 = 3 + 6 = 011000 \\
280 \cdot 61 &= 70 = 2 + 3 + 13 + 52 = 110101 \\
333 \cdot 61 &= 48 = 2 + 6 + 13 + 27 = 101110
\end{aligned}$$

Beautiful but broken

Lattice based techniques break this scheme.

Density of knapsack $d = \frac{n}{\max\{\log_2 w_i : 1 \leq i \leq n\}}$

Lattice basis reduction can solve knapsacks of low density. Unfortunately, M-H scheme always produces knapsacks of low density.