Cryptography and Complexity

- Symmetric key Encryption
- Key Exchange
- Asymmetric key encryption
- RSA
- NP-complete problems & cryptography
  - graph coloring
  - knapsack.

**Symmetric Key Encryption**

\[ c = E_k (m) \]

plaintext

Ciphertext

secret key

encryption function

\[ m = d_k (c) \]

decryption function

\[ E, d \] permute & reverse-permute
reversible operations \[ \oplus \land +/- \text{, shift left/right} \]

Symmetric algs: AES, RC5, DES
Key Management Question

How does secret key $k$ get exchanged/shared?

Pirates won’t touch locked box, but will take away keys, messages in unlocked box(es)

How does Alice send a message 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 $m$!
Diffe-Hellman Key Exchange

\[ g = F_p^* \]

finite field (mod p, \( p \) prime)

\(*\) means invertible elements only

\[ \{1, 2, \ldots, p-1\} \]

\underline{Alice} \hspace{1cm} g \text{ public} \hspace{1cm} \underline{Bob}

Select a

Compute \( g^a \)

\( g^b \) \hspace{1cm} \rightarrow \hspace{1cm} \text{Select } b \hspace{1cm} \text{Compute } g^b

Alice can compute \((g^b)^a \mod p = K\)

Bob can compute \((g^a)^b \mod p = K\)

\textbf{Attack? Man-in-the-middle}

Alice doesn't know she is communicating with Bob.

Alice agrees to a key with Eve (thinks she is Bob)

Bob agrees to a key with Eve (thinks she is Alice)

Eve can see all communications
Public Key Encryption

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.

RSA

Alice picks two large secret primes p & q.
Alice computes \( N = p \cdot q \)
Alice chooses an encryption exponent \( e \) which satisfies
\[
\gcd(e, (p-1)(q-1)) = 1 \quad e = 3, 17, 65537
\]
Alice public key = \( (N, e) \)

Decryption exponent obtained using Extended Euclidean Algorithm by Alice
\[
e \cdot d \equiv 1 \pmod{(p-1)(q-1)}
\]
Alice private key = \( (d, p, q) \)
Encryption & Decryption with RSA

\[ c = m^e \pmod{N} \]
\[ m = c^d \pmod{N} \]

**Why it works**

Group \((\mathbb{Z}/N\mathbb{Z})^*\) is ring with all invertible elements 
\(\gcd(x, N) = 1\)

By Lagrange's theorem
\[ x^{(p-1)(q-1)} \equiv 1 \pmod{N} \quad \text{for all} \quad x \in (\mathbb{Z}/N\mathbb{Z})^* \]

Since \(\phi(N) = (p-1)(q-1)\)
\(\phi\) is the Euler function
\(\phi\) is the number of relatively prime integers to \(N\) in \(\mathbb{Z}/N\mathbb{Z}\)

For some integer \(s\) we have:
\[ ed - s(p-1)(q-1) = 1 \]

\[ c^d = (m^e)^d \]
\[ = m^{ed} \]
\[ = m^{s(p-1)(q-1)} \]
\[ = m^{1+s(p-1)(q-1)} \]
\[ = m^{s(p-1)(q-1)+1} \]
\[ = m \]

5
HARDNESS OF RSA

1) Given N, hard to factor into p, q

2) Given e such that
   \( \gcd(e, (p-1)(q-1)) = 1 \)
   and c, find m such that
   \( m^e = c \pmod{N} \)

NP-Completeness

Is N composite? E NP with a factor within a range

Is a graph k-colorable? NP-complete
   Assign k colors to each vertex
   such that no two vertices connected
   by an edge share the same color
   not 3-colorable

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 \( S \)?

\[ S = b_1 w_1 + b_2 w_2 + \ldots + b_n w_n \]
NP-completeness & 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, because average graph, beyond a certain size, is not 3-colorable.

Consider standard backtracking search to determine 3-colorability.

Order vertices v_1, \ldots, 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 1st vertex \implies NOT
Successfully color last vertex \implies YES.

Random graph of t vertices, average number of vertices traveled < 147, regardless of t!
**KNAPSACK CRYPTOGRAPHY**

General knapsack problem: NP-complete

Super-increasing knapsacks: linear time solvable

\[ w_j \geq \frac{1}{2} \sum_{i=1}^{j-1} w_i \quad \{2, 3, 6, 13, 27, 52\} \]

Merkle-Hellman Cryptosystem:

Private key → super-increasing knapsack problem

\[ \text{PRIVATE TRANSFORM} \]

Public key ← "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 \( M \).

\( N = 31, M = 105 \)

Private key = \( \{2, 3, 6, 13, 27, 52\} \)

Public key = \( \{62, 93, 81, 88, 102, 37\} \).
Merkle–Hellman Example

Message: 011000 110101 101110

Ciphertext:
- \(0 \text{1} \text{1} \text{0} \text{0} \text{0} \text{0}\)
- \(1 \text{1} \text{0} \text{1} \text{0} \text{1}\)
- \(1 \text{0} \text{1} \text{1} \text{1} \text{0}\)

\[93 + 81 = 174\]
\[62 + 93 + 88 + 37 = 280\]
\[62 + 81 + 88 + 102 = 333\]

= 174, 280, 333

Recipient knows \(N = 31, M = 105\): \(\{2, 3, 6, 13, 27, 52\}\)

Multiply each ciphertext block by \(N^{-1} \pmod{M}\)

\(N^{-1} = 61 \pmod{105}\)

\[174 \cdot 61 = 9 = 3 + 6 = 011000\]
\[280 \cdot 61 = 70 = 2 + 3 + 13 + 52 = 110101\]
\[333 \cdot 61 = 48 = 2 + 6 + 13 + 27 = 101110\]

**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!

\(\mathcal{I}\) on average, easy to solve!