Cryptography

Dr. Ferdin Joe John Joseph

Cryptography

Cryptography

The field of study related to encoded information (comes from Greek word for "secret writing")

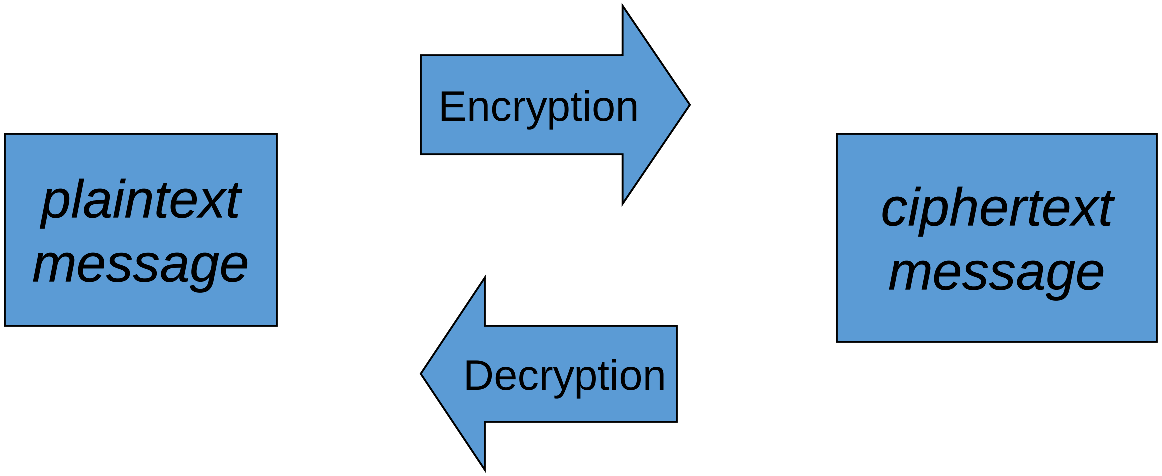
Encryption

The process of converting plaintext into ciphertext

Decryption

The process of converting ciphertext into plaintext

Cryptography



Encrypted(Information) cannot be read

Decrypted(Encrypted(Information)) can be

Cryptography

Cipher

An algorithm used to encrypt and decrypt text

Key

The set of parameters that guide a cipher

Neither is any good without the other

Substitution Ciphers

- A cipher that substitutes one character with another.
- These can be as simple as swapping a list, or can be based on more complex rules.
- These are NOT secure anymore, but they used to be quite common. What has changed?

Caesar ciphers

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

Substitute the letters in the second row for the letters in the top row to encrypt a message

Encrypt(COMPUTER) gives FRPSXWHU

Substitute the letters in the first row for the letters in the second row to decrypt a message

Decrypt(Encrypt(COMPUTER))

    = Decrypt(FRPSXWHU) = COMPUTER

Transposition Cipher

```
T O D A Y
+ I S + M
O N D A Y
```

Write the letters in a row of five, using '+' as a blank. Encrypt by starting spiraling inward from the top left moving counter clockwise

Encrypt(TODAY IS MONDAY) gives T+ONDAYMYADOIS+

Decrypt by recreating the grid and reading the letters across the row

The key are the dimension of the grid and the route used to encrypt the data

Cryptanalysis

Cryptanalysis

The process of decrypting a message without knowing the cipher or the key used to encrypt it

Substitution and transposition ciphers are easy for modern computers to break

To protect information more sophisticated schemes are needed

Encryption on computers

- Roughly speaking, there are two different broad types of encryption that are used on computers today
    - Symmetric encryption relies on keeping keys totally secret
    - Asymmetric encryption actually publicizes one key, but keeps some information private also
- Neither is really “better” - they just use different principles.
- In reality, both are vulnerable to attacks.

Symmetric, or private key cryptography

- Most common type is called a block cipher
  - Processes the plaintext in fixed sizes blocks
- Examples include DES, 3DES, and AES
- All require a secret key which is known by both parties in the communication
- Main issue here: need to securely swap the key. How can we do this?

DES: Data Encryption Standard

- Adopted in 1977 by National Bureau of Standards (now NIST)
- Divides message into blocks of 64 bits, and uses a key of 56 bits
- Key idea for this: XOR the data with the key
    - (Remember XOR? How did it work?)

DES

- In July 1998, DES was officially cracked by a machine built by the EFF
  - Total cost: under $250,000
  - Total time: 6-8 months
- They then published the details of their approach, which essentially was a brute force attack
- Note: 56 bits means 2^{56} keys to try
- Also, not as easy as just trying. What do you always do to files before sending them somewhere?

3DES

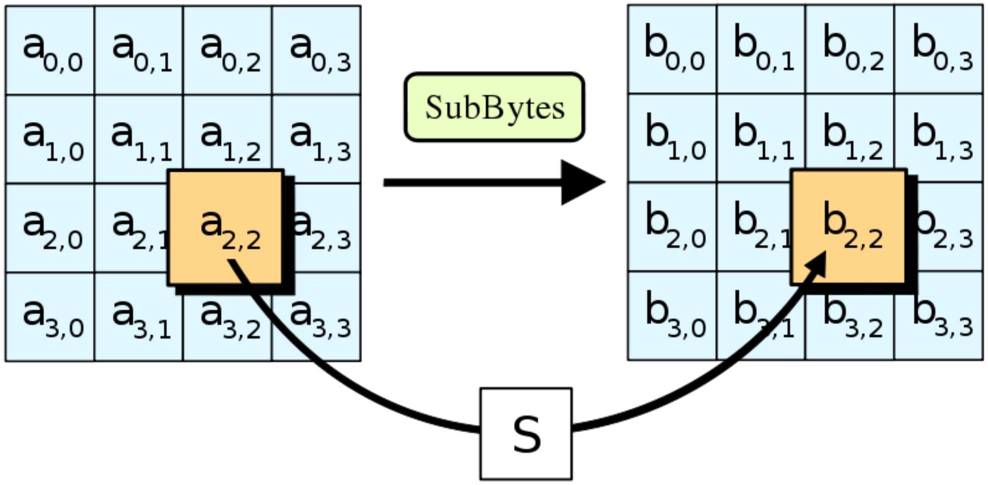
- Effort to salvage DES
- Main algorithm: repeat DES 3 times with different keys (so key size is now 168 bits)
- Still very secure - brute force attacks would take too long, and that is the only way to attack this algorithm
- Main problem: SLOW

Advanced Encryption Standard (AES)

- Designed in response to a call by NIST in 1998, and officially adopted in 2001
- Block length is 128 bits, and keys can be 128, 192, or 256 bits.
- Essentially, proceeds in 4 rounds (which are repeated):
  - Substitute bytes
  - Permute
  - Mix columns
  - Add round key

Stage 1: substitute bytes

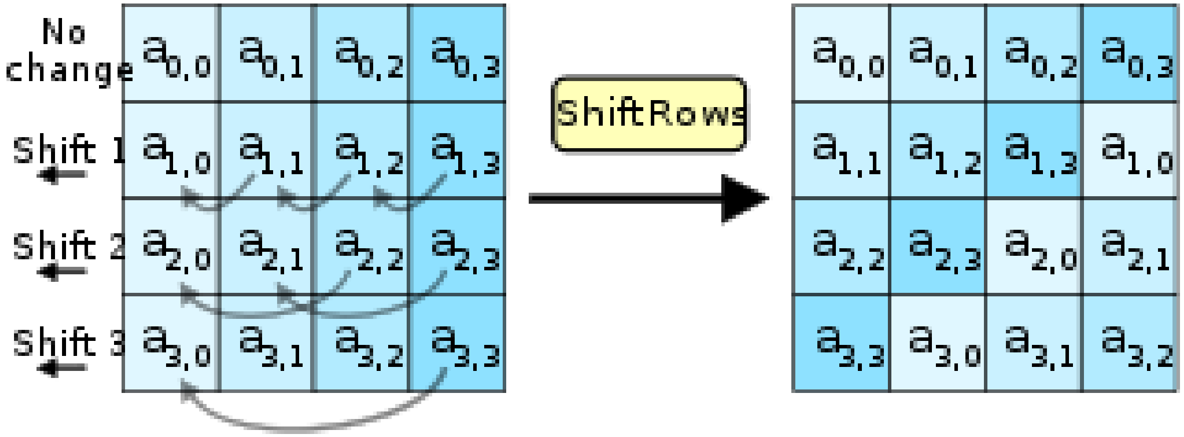
- AES computes a matrix which maps every 8-bit value to a different 8-bit value
- Computed using properties of finite fields (go take some math classes to learn more about this)



$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
S

Stage 2: permute

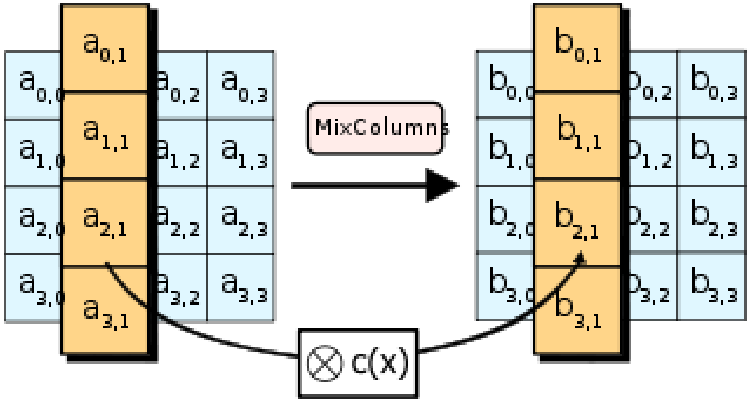
- AES then shifts each row, where each row is shifted a different amount



$$\begin{array}{c|cccc} \text{No change} & a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ \text{Shift 1} & a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ \text{Shift 2} & a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ \text{Shift 3} & a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{array} \xrightarrow{\text{ShiftRows}} \begin{array}{cccc} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,1} & a_{1,2} & a_{1,3} & a_{1,0} \\ a_{2,2} & a_{2,3} & a_{2,0} & a_{2,1} \\ a_{3,3} & a_{3,0} & a_{3,1} & a_{3,2} \end{array}$$

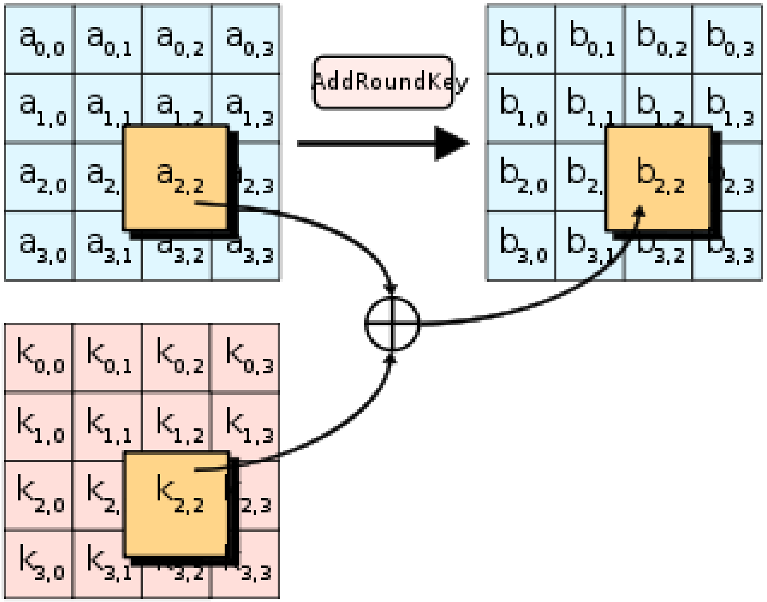
Stage 3: Mix columns

- Here, the 4 bytes in each column are combined using a linear transformation
- Essentially, the output of any byte depends on all the input bytes, so this "mixes" them together



Stage 4: Add round key

- Use XOR to combine the key with the message


$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
AddRoundKey
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
$\oplus$
$k_{0,0}$ $k_{0,1}$ $k_{0,2}$ $k_{0,3}$
$k_{1,0}$ $k_{1,1}$ $k_{1,2}$ $k_{1,3}$
$k_{2,0}$ $k_{2,1}$ $k_{2,2}$ $k_{2,3}$
$k_{3,0}$ $k_{3,1}$ $k_{3,2}$ $k_{3,3}$

Public Key Cryptography

- First revolution in cryptography in hundreds of years
- Originally introduced in a paper in 1976: “New directions in cryptography”, by Diffie and Hellman
- Initially, based on the goal of computing a common secret key (so combines well with AES or other symmetric algorithms)

Public/Private Keys

What is it?

An approach in which each user has two related keys, one public and one private

One's public key is distributed freely

A person encrypts an outgoing message, using the receiver's public key.

Only the receiver's private key can decrypt the message

Basic operations

- Logarithms: defined as the the exponent to which a fixed number, the base, must be raised to in order to produce that number
- Examples:
  - $\text{Log}_3 9 = 2$, since $3^2 = 9$
  - $\text{Log}_{10} 1000 = 3$, since $10^3 = 1000$
  - $\text{Log}_2 64 = 6$, since $2^6 = 64$

Basic operations (cont)

- Modulo operation: just taking remainders
- a mod b = remainder when a is divided by b
- Examples:
    - 1 mod 3 = 1
    - 15 mod 10 = 5
    - 256 mod 2 = 0

Public and private keys

- First, choose X, a secret key
- Then choose Q = a prime number, and A = some other number
- Set $Y = A^X \bmod Q$
- Note that $X = \log_A Y \bmod Q$

Public and private keys

- Now publish Y, A, and Q, but keep X secret
- Anyone knows that X = $\log_A$ Y mod Q, but this is difficult to compute!
- This is called the discrete logarithm problem - very similar to factoring in terms of difficulty, so no polynomial time algorithm is known.
- Essentially, computing Y given X is easy, but computing X given Y is much harder.
- (Go take number theory.)

How to encrypt

- So I know X, Y, A, and Q (but you don't know X).
- You get your own X', and the tell me $Y' = A^{X'}$ mod Q
- We can now compute our own secret key (and use it for AES or some other algorithm)
  - I will compute $(Y')^{X}$ mod Q = $(A^{X'})^{X}$ mod Q
  - You compute $(Y)^{X'}$ mod Q = $(A^{X})^{X'}$ mod Q
- These are equal! But an eavesdropper can't compute them, since they don't have X or X'

Attacks

- One downside: this is less secure than pure symmetric encryption
- There are ways to attack this that do better than brute force
- Number theory and group theory allow theoretical attacks that are provably better than exponential, but worse than polynomial time
- So it is NOT known if this problem is really hard! Someone could develop a polynomial time attack. It just hasn’t been done yet.

RSA

- In 1977, Rivest, Shamir, and Adleman came up with another way to use public key cryptography
- Rather than secure key exchanges, this one actually lets you encrypt whole messages
- Today, this is the most commonly used public key cryptosystem on the market

How RSA works

- Choose 2 prime numbers, p and q
- Set n=pq
- Compute $\phi(n)$ = the number of numbers less than n which are relatively prime to n
  - (That means numbers which have no common divisors.)
- Here, $\phi(n) = \phi(p)\ \phi(q)$
  - What is $\phi(p)$? $\phi(q)$?

RSA (cont.)

- So $\phi(n) = (p-1)(q-1)$, which we can compute.
- Note that this is hard to find if you don't know p and q, but it's easy if you do.
- Now pick a value e, where e is relatively prime to $\phi(n)$. This is your public key.
- Compute another value d, where we must have de = 1 mod $\phi(n)$. This is your private key.
  - Example: Suppose e = 2, n = 11. Then d = 6, since we know (6)(2) mod 11= 12 mod 11 = 1

Encrypting with RSA

- Now I have a message m, as well as e and n.
- I compute c = m^e mod n, and send it to you.
- You have d, so you can compute the value c^d mod n = $(m^e)^d$ mod n = m^1 mod n.
- But without d, this is not easy! Equivalent to factoring in difficulty.

Public/Private Keys:
Other uses

Digital signature

Data that is appended to a message, made from the message itself and the sender's private key, to ensure the authenticity of the message

Digital certificate

A representation of a sender's authenticated public key used to minimize malicious forgeries