



Computers and Society

Encryption

Chris Brooks

Department of Computer Science

University of San Francisco



Terminology

- Code
 - Replacement based on words or semantic structures
 - Each word has a corresponding code word.

Terminology

- Code
 - Replacement based on words or semantic structures
 - Each word has a corresponding code word.
- Cipher
 - Replacement based on symbols
 - Each letter is mapped to another letter.

Terminology

- Cryptography
 - The science of encrypting or hiding *secrets*
- Cryptanalysis
 - The science of decrypting messages or breaking codes and ciphers
- Cryptology
 - The combination of the two.

Applications

- Why might someone want to use encryption?

Applications

- Why might someone want to use encryption?
 - Military uses
 - Protect business secrets
 - Protect financial information (credit card numbers, etc)
 - Protect communication from unauthorized access
 - Protecting stored data

Applications

- Why might someone want to use encryption?
 - Authenticating payment or permission
 - More flexible payment schemes (digital cash)
 - Protecting intellectual property
 - Espionage/sabotage
 - Others?

More Terminology

- Plaintext - an unencrypted message
- Ciphertext - an encrypted message
- An encryption scheme will depend on being easy to generate ciphertext from plaintext, but hard to generate plaintext from ciphertext.

More Terminology

- Symmetric-key encryption
 - Also called *secret key encryption*
 - One key is used for both encryption and decryption
- Asymmetric key encryption
 - Also called *public key encryption*
 - Complementary keys are used to encrypt and decrypt

Simple examples

- Caesar cipher (shift cipher)
 - Replace each letter with $+3 \pmod{26}$
 - “Attack at dawn” becomes “Dttdfn dw gdzq”
- Two components:
 - Algorithm: shift each letter by a fixed amount
 - Key: The amount to shift each letter.
- Knowing the algorithm (but not the key) makes this cipher easy to crack.
- How many possible plaintexts does “Dwwdfn dw gdzq” have?

Weaknesses of the Caesar Cipher

- Word structure is preserved.
- An attacker could notice that 'dw' is a two-letter word, so either d or w must be a vowel.
- Our encryption scheme does not hide all of the information in the plaintext.
 - Solution: Break message into equal-sized blocks.
 - “dww dfn dwg dzq”

Weaknesses of the Caesar Cipher

- Letter frequency is a big clue
 - e,t,a,o are the most common English letters.
 - Using a single key preserves frequency.
- Solution: use multiple keys
 - E.g. shift first char by 3, second by 5, third by 7.
 - “Attack at dawn” becomes “dva dhr dvk dbu”
 - Better, but there is still frequency information present.
 - An attacker that knows the block size can determine which characters were encoded with each key.
 - Frequency information is a powerful tool for subverting an encryption scheme.

Caesar cipher

- The Caesar cipher is still useful as a way to prevent people from unintentionally reading something.
 - ROT-13
 - By taking action to decrypt, the user agrees that they want to view the content.
- Fundamental problem with Caesar cipher: message is longer than the key.

Vernam Cipher

- 1920's: introduction of the one-time pad.
 - Message represented as a bitstring
 - Randomly generated key
 - Same length as message
 - XORed with message
 - Theoretically unbreakable
 - Attacker can do no better than guessing
 - Ciphertext gives no information about plaintext.

Vernam Cipher

- Example: winning lottery number is 117
 - 1110101 (7 bits)
 - Randomly generated key: 0110101
 - XOR: 1000000
- No two bits are encoded with the same mapping
- An attacker has no frequency information to help guess the key.
- Problem: keys are very large.
- New key is needed for each message.
- How to distribute these keys?
- Shared source of randomness?

Keyspace

- In thinking about how difficult an encryption algorithm is to break, we'll often talk about the size of the *keyspace*.
- Set of all possible keys.
 - Caesar cipher: keyspace = $\{0, 1, 2, \dots, 25\}$
 - Vernam cipher: $|\text{keyspace}| = 2^n - 1$

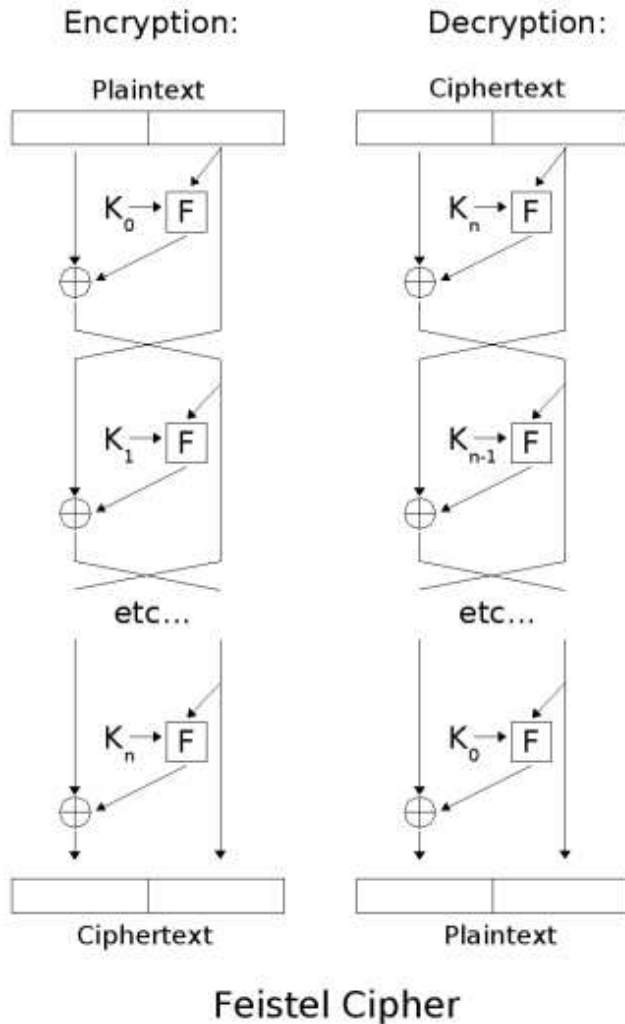
Symmetric Key Encryption

- Caesar cipher and the one-time pad are examples of symmetric key (secret key) encryption.
- Same key used to encrypt and decrypt.
- All users share key.
- Advantage: Very fast
- Disadvantage: How to securely distribute keys?

DES

- Data Encryption Standard. (DEA is actually the algorithm)
- DES uses a 56-bit key.
- The message is broken into a set of 64-bit blocks, each of which is encrypted separately.
- Each block is passed through a *Feistel structure*

Feistel structure



- 16 distinct 48-bit keys are generated from the 56-bit key.
- At each stage of the structure, the key is used to permute the lower half of the message.
- This is XORed with the upper half.
- The same structure is used for decryption.

DES

- DES was the first commercial-grade encryption algorithm whose implementation details were open.
- Keyspace: $2^{56} = 10^{17}$
- No longer considered secure, due to key size.
 - DES has been cracked in under 24 hours by distributed.net and EFF's DeepCrack.

Extensions to DES

- 3DES
- Message is run through DES 3 times
- $C = k_3(k_2(k_1(M)))$
- Backwards-compatible with DES if all three keys are the same.
- Keyspace is 10^{168}
- Drawback: bit-oriented operations are slow to implement in software

Other symmetric-key algorithms

- AES - eventual replacement for DES
- IDEA - used within PGP
- Blowfish - fast and compact, popular in commercial applications.
- RC5 - Fast, highly parameterized, low memory requirements.

Issues with symmetric-key encryption

- Advantages:

- Fast and secure

- Disadvantages:

- How to distribute the secret key?
- Particularly in one-shot communications.

Public-Key Encryption

- With public-key (or *asymmetric-key*) encryption, we use one key to encrypt the message and one key to decrypt.
- Users create two keys: a public key and a private key.
- The public key is made available to everyone.
- The private key is kept to yourself.

Public-key encryption

- To send a message, Alice finds Bob's public key and encrypts the message.
- The ciphertext is then sent to Bob.
- Only someone with Bob's private key can decrypt the message.
- To reply, Bob creates a message and encrypts it with Alice's public key.
- Only someone with Alice's private key can decrypt the message.

One-way functions

- Most functions are easily *invertible*. (if $y = f(x)$, then $x = f^{-1}(y)$)
 - e.g. addition/subtraction, square/square root.
- A function which is easy to compute in one direction but hard to compute in the other is known as a *one-way* function.
 - Hashing, modular arithmetic
- A one-way function which becomes easy to compute in the “reverse” direction with an additional piece of knowledge is known as a *trapdoor one-way function*.
- Public-key encryption depends on the use of trapdoor one-way functions.

Overview of RSA

- RSA is the most common and well-known public key cryptosystem
- Basic notation: a key pair (e,d) contains two keys:
 - e is the public key (used to encrypt documents)
 - d is the private key (used to decrypt documents)
 - M is the plaintext message.
 - Let R be the encryption function.
 - $R(e, M) = C$. $R(d, C) = M$. - encryption
 - $R(d, M) = C'$ $R(e, C') = M$ - signing
 - $R(e, R(d, M)) = M = R(d, R(e, M))$
- Same function is used for both operations.

Modular Arithmetic

- RSA's security is based on modular arithmetic.
- $a = b(\text{mod } n) \implies$ there is a q such that $a - b = qn$
- b is the remainder after dividing a by n
- $23 = 3(\text{mod } 5)$
- Two numbers p and q are said to be relatively prime if their greatest common divisor is 1.
 - 5 and 17, 8 and 9, 10 and 21

Multiplicative Inverses

- An inverse is a number that maps a given
- number to the identity (1 in our case)
- We are particularly interested in multiplicative inverses for modular arithmetic.
- $(ab) = 1(\text{mod } n)$

Multiplicative Inverses

- 3 and 2 are multiplicative inverses mod 5.
- 7 and 6 are multiplicative inverses mod 41.
- 5 and 2 are multiplicative inverses mod 9.
- For $n > 1$, if a and n are relatively prime, there is a unique multiplicative inverse: $ax = 1 \pmod{n}$

The RSA Algorithm

- Pick two large (100 digit) primes p and q .
- Let $n = pq$
- Select a relatively small integer d that is prime to $(p - 1)(q - 1)$
- Find e , the multiplicative inverse of $d \bmod (p - 1)(q - 1)$
- (d, n) is the public key. To encrypt M , compute
 - $En(M) = M^e \pmod n$
- (e, n) is the private key. To decrypt C , compute
 - $De(C) = C^d \pmod n$

Example

- Let $p = 11, q = 13$
- $n = pq = 143$
- $(p - 1)(q - 1) = 120$
- Possible d : 7, 11, 13, 17, ... (let's use 7)
- Find e : $e * 7 = 1(\text{mod } 120) \rightarrow 103$
- Public key: (7, 143)
- Private key: (103, 143)
- $En(42) = 427 \pmod{143} = 81$
- $De(81) = 81 \pmod{143} = 42$

Security of RSA

- The security of RSA is dependent on the assumption that it's difficult to generate the private key d from the public key e and the modulus n .
- Equivalent to integer factorization problem.
- This is how we got e and d in the first place.
- Factoring is thought to be computationally hard.
 - No proof, though!

Breaking RSA in practice

- RSA is the most well-known public-key algorithm.
- RSA systems has sponsored a series of challenges to crack RSA.
- RSA-512 broken in 1999.
- Estimate: capability grows by 13-14 bits (about 4 digits) a year.
- At this rate, 1024-bit RSA will be secure until about 2037.
- Question: How long does your data need to be secure?

Issues with public-key encryption

- Advantages:
 - key distribution simplified
- Disadvantages
 - Encryption/decryption much slower
 - Much larger keys needed
 - Spoofing of public keys

Digital signing

- Alice wants to send Bob a message. She also wants Bob to be able to verify that it really came from her.
- She can encrypt the message with her private key and send it to Bob.
- Anyone can use her public key to read the message.
- Only Alice (or someone with her private key) can encrypt a message that can be read with her public key.
- This is called a *digital signature*; if Alice's private key is secure, then she must have sent the message.
- We can combine encryption and digital signing.

Digital Signatures

- Desirable properties of a digital signature:
 - A receiver must be able to validate the signature
 - The signature must not be forgeable
 - The signer must not be able to repudiate the signature.
- Encrypt with private key, validate with public key.
- For security and authenticity, encrypt the signed message with the receiver's public key.

Validating with Hash Functions

- To sign a document, I compute its hash, encrypt that with my private key, and send the encrypted hash along with the original document as plaintext.
- The receiver hashes the plaintext and then uses my public key to verify that I was the one who sent the document.
- Can also detect tampering.

Combining Public and Secret Keys

- Since public-key encryption/decryption is slow, it's not very practical for large data transfers, such as SSL or HTTPS.
- Secret-key systems work better, but a method is needed to exchange keys.
- Public-key encryption is often used to synchronize secret session keys.

Combining Public and Secret Keys

- A generates a secret key and sends it to B, encrypted with B's public key.
- For handshaking, include a random number.
- B decrypts the message and has the secret key.
- For handshaking, B encrypts the random number with A's public key and returns it.

Summary

- Symmetric-key: fast, relatively small keys. Same function used to encrypt and decrypt.
- Challenge: how to securely exchange keys?
- Asymmetric-key: slower, larger keys. Inverse functions used.
- Allows for communication between strangers.

Coming Attractions

- Public key infrastructure
- Applications of encryption
- Regulation of encryption