Cryptography: History and Simple Encryption Methods and Preliminaries
Cryptography

The word cryptography comes from the Greek words κρυπτός (hidden or secret) and γράφειν (writing).

So historically cryptography has been the “art of secret writing.”

Most of cryptography is currently well grounded in mathematics and it can be debated whether there’s still an “art” aspect to it.
Cryptography can be used at different levels

- **Algorithms**: encryption, signatures, hashing, Random Number Generator (RNG)

- **Protocols** (2 or more parties): key distribution, authentication, identification, login, payment, etc.

- **Systems**: electronic cash, secure filesystems, smartcards, VPNs, e-voting, etc.

- **Attacks**: on all the above
Some Applications of Cryptography

- Network, operating system security
- Protect Internet, phone, space communication
- Electronic payments (e-commerce)
- Database security
- Software/content piracy protection
- Pay TV (e.g., satellite)
- Military communications
- Voting
Open vs. Closed Design Model

- **Open design**: algorithm, protocol, system design (and even possible plaintext) are public information. Only key(s) are kept secret.

- **Closed design**: as much information as possible is kept secret.
Core Issue in Network security : How to Communicate Securely?

Looks simple ... But, the devil is in the details

Note: even storage is a form of communication
The Biggest “Headache” is that…

Good security must be

Effective

Yet

Unobtrusive

Because security is not a service in and of itself, but a burden!
Cryptography is Old ...

• Most sub-fields in CS are fairly new (20-30 years):
  – Graphics, compilers, software, OS, architecture

• And, a few are quite old (more than several decades):
  – Cryptography, database, networking
Some History: Caesar’s Cipher

Homo
Hominem
Lupus!

Krpr
Krplqhp
Oxssxv!
Some History: Rosetta Stone
Some History: Enigma

Alan Turing
(1912-1954)
Historical (Primitive) Ciphers

- **Shift (e.g., Caesar):** \( \text{Enc}_k(x) = x + k \mod 26 \)

- **Affine:** \( \text{Enc}_{k_1,k_2}(x) = k_1 \times x + k_2 \mod 26 \)

- **Substitution:** \( \text{Enc}_{\text{perm}}(x) = \text{perm}(x) \)

- **Vigenere:** \( \text{Enc}_K(x) = (X[0]+K[0], X[1]+K[1], \ldots) \)

- **Vernam:** One-Time Pad (OTP)
Shift (Caesar) Cipher

Example:

\[ K = 11 \]

\[
\begin{array}{cccccccccccccc}
W & E & W & I & L & L & M & E & E & T & A & T & M & I & D & N & I & G & H & T \\
H & P & H & T & W & W & X & P & P & E & L & E & X & T & O & Y & T & R & S & E
\end{array}
\]

- How many keys are there?
- How many trials are needed to find the key?
Substitution Cipher

Example:

\[
\begin{align*}
\text{A} & \rightarrow \text{X} \\
\text{B} & \rightarrow \text{N} \\
\text{C} & \rightarrow \text{Y} \\
\text{D} & \rightarrow \text{A} \\
\text{E} & \rightarrow \text{H} \\
\text{F} & \rightarrow \text{P} \\
\text{G} & \rightarrow \text{O} \\
\text{H} & \rightarrow \text{G} \\
\text{I} & \rightarrow \text{T} \\
\text{J} & \rightarrow \text{K} \\
\text{K} & \rightarrow \text{Z} \\
\text{L} & \rightarrow \text{Q} \\
\text{M} & \rightarrow \text{W} \\
\text{N} & \rightarrow \text{E} \\
\text{O} & \rightarrow \text{L} \\
\text{P} & \rightarrow \text{M} \\
\text{Q} & \rightarrow \text{F} \\
\text{R} & \rightarrow \text{C} \\
\text{S} & \rightarrow \text{V} \\
\text{T} & \rightarrow \text{R} \\
\text{U} & \rightarrow \text{Y} \\
\text{V} & \rightarrow \text{U} \\
\text{W} & \rightarrow \text{X} \\
\text{X} & \rightarrow \text{B} \\
\text{Y} & \rightarrow \text{T} \\
\text{Z} & \rightarrow \text{D}
\end{align*}
\]

\[
\begin{align*}
\text{W} & \rightarrow \text{K} \\
\text{E} & \rightarrow \text{H} \\
\text{W} & \rightarrow \text{Z} \\
\text{I} & \rightarrow \text{T} \\
\text{L} & \rightarrow \text{M} \\
\text{M} & \rightarrow \text{E} \\
\text{E} & \rightarrow \text{T} \\
\text{T} & \rightarrow \text{A} \\
\text{A} & \rightarrow \text{M} \\
\text{T} & \rightarrow \text{I} \\
\text{M} & \rightarrow \text{D} \\
\text{I} & \rightarrow \text{N} \\
\text{G} & \rightarrow \text{H} \\
\text{H} & \rightarrow \text{T}
\end{align*}
\]

- How many keys are there?
- How many trials are needed to find the key?
Substitution Cipher

Cryptanalysis

Probabilities of Occurrence
Substitution Cipher

Cryptanalysis

Frequency of some common digrams
VERNAM One-Time Pad (OTP): World’s Best Cipher

Plaintext = \{p_0, ..., p_{n-1}\}

One-time pad stream = \{otp_0, ..., otp_{n-1}\}

Ciphertext = \{c_0, ..., c_{n-1}\}

where:

\[ c_i = p_i \oplus otp_i \quad \forall 0 < i < n \]

\[ C = A \oplus B \]

\[ C \oplus B = A \]
VERNAM One-Time Pad (OTP): World’s Best Cipher

• Vernam offers perfect information-theoretic security,

but:

• How long does the OTP keystream need to be?
• How do Alice and Bob exchange the keystream?
Encryption Principles

- A cryptosystem has (at least) five ingredients:
  - Plaintext
  - Secret Key
  - Ciphertext
  - Encryption Algorithm
  - Decryption Algorithm

- Security usually depends on the secrecy of the key, not the secrecy of the algorithms
Crypto Basics

Crypto Attacks:
- ciphertext only
- known plaintext
- chosen plaintext
- chosen ciphertext
- brute force

Cryptosystem:
- $P$ -- plaintext
- $C$ -- ciphertext
- $K$ -- keyspace
- $E$ -- encryption rules
- $D$ -- decryption rules

Encryptor/Prover

Decryptor/Verifier
## Average Time Required for Exhaustive Key Search (for Brute Force Attacks)

<table>
<thead>
<tr>
<th>Key Size (bits)</th>
<th>Number of Alternative Keys</th>
<th>Time required at $10^6$ Decr/µs</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>$2^{32} = 4.3 \times 10^9$</td>
<td>2.15 milliseconds</td>
</tr>
<tr>
<td>56</td>
<td>$2^{56} = 7.2 \times 10^{16}$</td>
<td>10 hours</td>
</tr>
<tr>
<td>128</td>
<td>$2^{128} = 3.4 \times 10^{38}$</td>
<td>$5.4 \times 10^{18}$ years</td>
</tr>
<tr>
<td>168</td>
<td>$2^{168} = 3.7 \times 10^{50}$</td>
<td>$5.9 \times 10^{30}$ years</td>
</tr>
</tbody>
</table>
Types of Attainable Security

• **Perfect, unconditional or “information theoretic”:** the security is evident free of any (computational/hardness) assumptions

• **Reducible or “provable”:** security can be shown to be based on some common (often unproven) assumptions, e.g., the conjectured difficulty of factoring large integers

• **Ad hoc:** the security seems good often -> “snake oil”...

Take a look at:

http://www.ciphersbyritter.com/GLOSSARY.HTM
Computational Security

• Encryption scheme is *computationally secure* if
  - cost of breaking it (via brute force) exceeds the value of the encrypted information; or
  - time required to break it exceeds useful lifetime of the encrypted information

• Most modern schemes we will see are *considered* computationally secure
  - Usually rely on very large key-space, impregnable to brute force

• Most advanced schemes rely on lack of knowledge of effective algorithms for certain hard problems, not on a proven inexistence of such algorithms *(reducible security)*!
  - Such as: factorization, discrete logarithms, etc.
Complexity Reminder/Re-cap

- **P**: problems that can be solved in polynomial time, i.e., problems that can be solved/decided “efficiently”

- **NP**: broad set of problems that includes P;
  - answers can be verified “efficiently” (in polynomial time);
  - solutions cannot always be efficiently found (as far as we know).

- **NP-complete**: the believed-to-be-hard decision problems in NP, they appear to have no efficient solution; answers are efficiently verifiable, solution to one is never much harder than a solution to another

- **NP-hard**: hardest; some of them may not be solved by a non-deterministic TM. Many computational version of NP-complete problems are NP-hard.

- **Examples**:
  - Factoring, discrete log are in NP, not know if NP-complete or in P
  - Primality testing was recently (2002) shown to be in P
  - Knapsack is NP-complete

For more info, see: https://www.nist.gov/dads//
P vs NP

P ≠ NP

P = NP = NP-Complete

NP-Complete

NP-Hard

NP

P
Cryptosystems

Classified along three dimensions:

- **Type of operations used for transforming plaintext into ciphertext**
  - Binary arithmetic: shifts, XORs, ANDs, etc.
    - Typical for *conventional* encryption
  - Integer arithmetic
    - Typical for *public key* encryption

- **Number of keys used**
  - Symmetric or conventional (single key used)
  - Asymmetric or public-key (2 keys: 1 to encrypt, 1 to decrypt)

- **How plaintext is processed:**
  - One bit at a time
  - A string of any length
  - A block of bits
Conventional Encryption Principles
Conventional (Symmetric) Cryptography

- Alice and Bob share a key $K_{AB}$ which they somehow agree upon (how?)
  - key distribution / key management problem
  - ciphertext is roughly as long as plaintext
  - examples: Substitution, Vernam OTP, DES, AES
Uses of Conventional Cryptography

• Message transmission (confidentiality):
  • Communication over insecure channels
• Secure storage: crypt on Unix
• Strong authentication: proving knowledge of a secret without revealing it:
  • See next slide
  • Eve can obtain chosen <plaintext, ciphertext> pair
  • Challenge should be chosen from a large pool
• Integrity checking: fixed-length checksum for message via secret key cryptography
  • Send MAC along with the message MAC=H(m,K)
Challenge-Response Authentication Example

\[ K_{AB} \]

challenge

\[ r_a \]

challenge reply

\[ K_{AB}(r_a) \]

challenge

\[ r_b \]

challenge reply

\[ K_{AB}(r_b) \]
Conventional Cryptography

➢ Advantages
  • high data throughput
  • relatively short key size
  • primitives to construct various cryptographic mechanisms

➢ Disadvantages
  • key must remain secret at both ends
  • key must be distributed securely and efficiently
  • relatively short key lifetime
Public Key Cryptography

• Asymmetric cryptography
• Invented in 1974-1978 (Diffie-Hellman and Rivest-Shamir-Adleman)
• Two keys: private (SK), public (PK)
  • Encryption: with public key;
  • Decryption: with private key
• Digital Signatures: Signing by private key; Verification by public key. i.e., “encrypt” message digest/hash -- $h(m)$ -- with private key
  • Authorship (authentication)
  • Integrity: Similar to MAC
  • Non-repudiation: can’t do with secret key cryptography
• Much slower than conventional cryptography
  • Often used together with conventional cryptography, e.g., to encrypt session keys
Public Key Cryptography

Bob’s public key

plaintext message, \( m \) → encryption algorithm → ciphertext \( \text{PK}_B(m) \) → decryption algorithm → plaintext message \( m = \text{SK}_B(\text{PK}_B(m)) \)

Bob’s private key

Alice

Bob's public key

Bob's private key
Uses of Public Key Cryptography

• Data transmission (confidentiality):
  • Alice encrypts $m_a$ using $PK_B$, Bob decrypts it to obtain $m_a$ using $SK_b$.

• Secure Storage: encrypt with own public key, later decrypt with own private key

• Authentication:
  • No need to store secrets, only need public keys.
  • Secret key cryptography: need to share secret key for every person one communicates with

• Digital Signatures (authentication, integrity, non-repudiation)
Public Key Cryptography

- Advantages
  - only the private key must be kept secret
  - relatively long life time of the key
  - more security services
  - relatively efficient digital signatures mechanisms

- Disadvantages
  - low data throughput
  - much larger key sizes
  - distribution/revocation of public keys
  - security based on conjectured hardness of certain computational problems
Comparison Summary

- Public key
  - encryption, signatures (esp., non-repudiation) and key management

- Conventional
  - encryption and some data integrity applications

- Key sizes
  - Keys in public key crypto must be larger (e.g., 2048 bits for RSA) than those in conventional crypto (e.g., 112 bits for 3-DES or 256 bits for AES)
    - most attacks on “good” conventional cryptosystems are exhaustive key search (brute force)
    - public key cryptosystems are subject to “short-cut” attacks (e.g., factoring large numbers in RSA)
Suggested Readings:

Chapters 1 and 2 in KPS book
Optional: Ch 1 in Stinson

Don't forget to check the website!
Did you do it before this lecture?