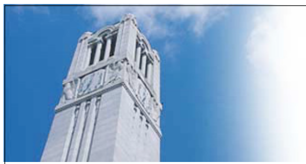


NC STATE UNIVERSITY Computer Science

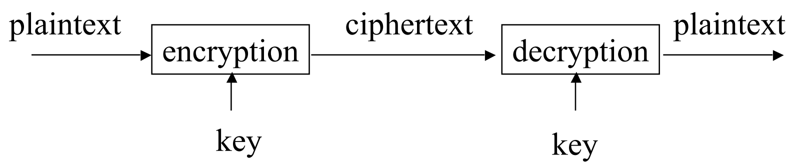
CSC 474/574
Information Systems Security

Topic 2.1 Introduction to Cryptography

Cryptography

- Cryptography
  - Original meaning: The art of secret writing
  - Becoming a science that relies on mathematics (number theory, algebra)
  - Process data into intelligible form, reversible, without data loss
  - Usually one-to-one (not compression)

Encryption/Decryption



- Plaintext: a message in its original form
- Ciphertext: a message in the transformed, unrecognized form
- Encryption: the process that transforms a plaintext into a ciphertext
- Decryption: the process that transforms a ciphertext to the corresponding plaintext
- Key: the value used to control encryption/decryption.

Cryptanalysis

- Ciphertext only:
  - Analyze only with the ciphertext
  - Example: Exhaustive search until “recognizable plaintext”
  - Smarter ways available
- Known plaintext:
  - Secret may be revealed (by spy, time), thus <ciphertext, plaintext> pair is obtained
  - Great for mono-alphabetic ciphers

Cryptanalysis (Cont'd)

- Chosen plaintext:
  - Choose text, get encrypted
  - Useful if limited set of messages
- Chosen ciphertext:
  - Choose ciphertext
  - Get feedback from decryption, etc.

Security of An Encryption Algorithm

- Unconditionally secure
  - It is impossible to decrypt the ciphertext
  - One-time pad (the key is as long as the plaintext)

$$C_i = P_i \oplus k_i$$

- Computationally secure
  - The cost of breaking the cipher exceeds the value of the encrypted information
  - The time required to break the cipher exceeds the useful lifetime of the information

Secret Keys v.s. Secret Algorithms

- Security by obscurity
  - We can achieve better security if we keep the algorithms secret
  - Hard to keep secret if used widely
  - Reverse engineering, social engineering
- Publish the algorithms
  - Security of the algorithms depends on the secrecy of the keys
  - Less unknown vulnerability if all the smart (good) people in the world are examine the algorithms

Secret Keys v.s. Secret Algorithms (cont'd)

- Commercial world
  - Published
  - Wide review, trust
- Military
  - Keep algorithms secret
  - Avoid giving enemy good ideas
  - Military has access to the public domain knowledge anyway.

Some Trivial Codes

- Caesar cipher: substitution cipher:
  - Replace each letter with the one 3 letters later
  - A → D, B → E
- Captain Midnight Secret Decoder rings:
  - shift variable by *n*: IBM → HAL
  - only 26 possibilities

Some Trivial Codes (Cont'd)

- Mono-alphabetic cipher:
  - generalization, arbitrary mapping of one letter to another
  - 26!, approximately 4 × 1026
  - statistical analysis of letter frequencies

Some Trivial Codes (Cont'd)

- Hill Cipher
  - Encryption: **C** = **KP** or
  - Decryption: **P** = $\mathbf{K}^{-1}\mathbf{C}$
  - Problem:
    - Known plaintext attack

Some Trivial Codes (cont'd)

- Poly-alphabetic Ciphers
  - A set of related mono-alphabetic substitution rules is used
  - A key determines which particular rule is chosen for a given transformation

Some Trivial Codes (Cont'd)

- All the previous codes are based on substitution
- Transposition (permutation)

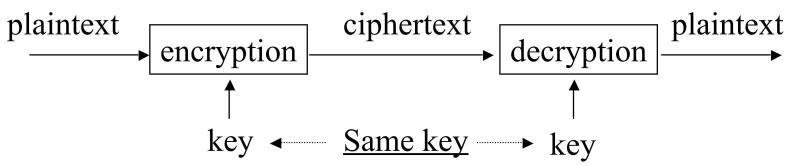
Key:	4	3	1	2	5	6	7
Plaintext:	A	T	T	A	C	K	P
	O	S	T	P	O	N	E
	D	U	N	T	I	L	T
	W	O	A	M	X	Y	Z

- Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ

Types of Cryptography

- Number of keys
  - Hash functions: no key
  - Secret key cryptography: one key
  - Public key cryptography: two keys - public, private
- The way in which the plaintext is processed
  - Block cipher: divides input elements into blocks
  - Stream cipher: process one element (e.g., bit) a time

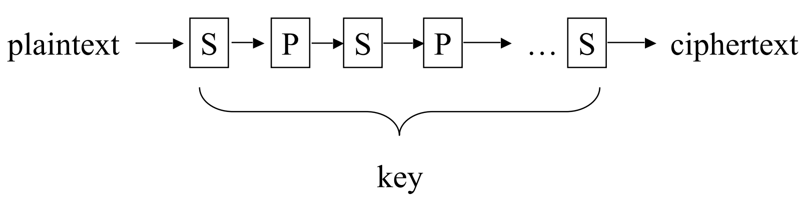
Secret Key Cryptography



- Same key is used for encryption and decryption
- Also known as
  - Symmetric cryptography
  - Conventional cryptography

Secret Key Cryptography (cont'd)

- Basic technique
  - Product cipher:
  - Multiple applications of interleaved substitutions and permutations



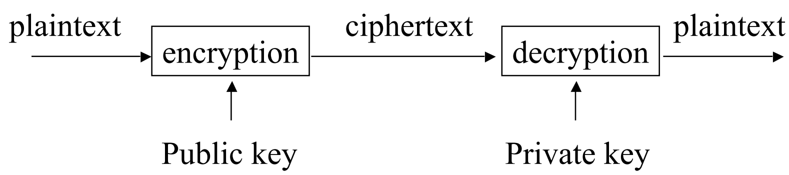
Secret Key Cryptography (cont'd)

- Ciphertext approximately the same length as plaintext
- Examples
  - Stream Cipher: RC4
  - Block Cipher: DES, IDEA, AES

Applications of Secret Key Cryptography

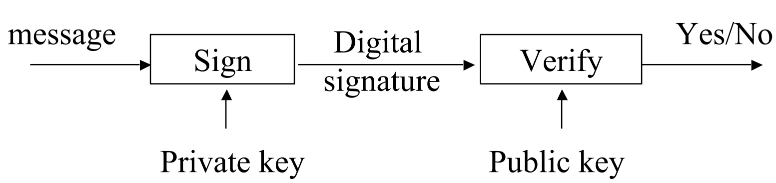
- Transmitting over an insecure channel
  - Challenge: How to share the key?
- Secure Storage on insecure media
- Authentication
  - Challenge-response
  - To prove the other party knows the secret key
  - Must be secure against chosen plaintext attack
- Integrity check
  - Message integrity code (MIC)

Public Key Cryptography



- Invented/published in 1975
- A public/private key pair is used
  - Public key can be publicly known
  - Private key is kept secret by the owner of the key
- Much slower than secret key cryptography
- Also known as
  - Asymmetric cryptography

Public Key Cryptography (Cont'd)



- Another mode: digital signature
  - Only the party with the private key can create a digital signature.
  - The digital signature is verifiable by anyone who knows the public key.
  - The signer cannot deny that he/she has done so.

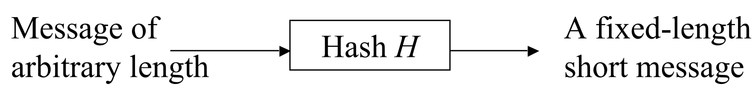
Applications of Public Key Cryptography

- Data transmission:
  - Alice encrypts m_a using Bob's public key e_B, Bob decrypts m_a using his private key d_B.
- Storage:
  - Can create a safety copy: using public key of trusted person.
- Authentication:
  - No need to store secrets, only need public keys.
  - Secret key cryptography: need to share secret key for every person to communicate with.

Applications of Public Key Cryptography (Cont'd)

- Digital signatures
  - Sign hash $H(m)$ with the private key
    - Authorship
    - Integrity
    - Non-repudiation: can't do with secret key cryptography
- Key exchange
  - Establish a common session key between two parties

Hash Algorithms



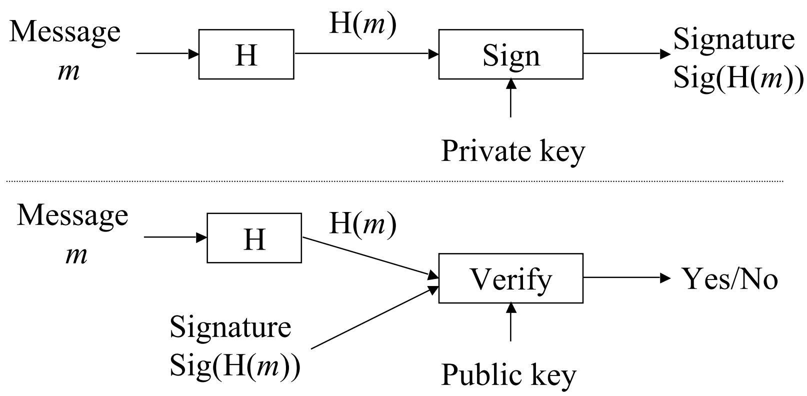
- Also known as
  - Message digests
  - One-way transformations
  - One-way functions
  - Hash functions
- Length of $H(m)$ much shorter then length of m
- Usually fixed lengths: 128 or 160 bits

Hash Algorithms (Cont'd)

- Desirable properties of hash functions
  - Performance: Easy to compute $H(m)$
  - One-way property: Given $H(m)$ but not m, it's difficult to find m
  - Weak collision free: Given $H(m)$, it's difficult to find m' such that $H(m') = H(m)$.
  - Strong collision free: Computationally infeasible to find m_1, m_2 such that $H(m_1) = H(m_2)$

Applications of Hash Functions

- Primary application
  - Generate/verify digital signature



Applications of Hash Functions (Cont'd)

- Password hashing
  - Doesn't need to know password to verify it
  - Store *H*(*password*+*salt*) and salt, and compare it with the user-entered password
  - Salt makes dictionary attack more difficult
- Message integrity
  - Agree on a secrete key *k*
  - Compute *H*(*m*|*k*) and send with *m*
  - Doesn't require encryption algorithm, so the technology is exportable