Cryptography and Cryptanalysis

CMSC 426 - Computer Security

1

Outline

- Cryptology
- Symmetric Encryption
- Cryptanalytic Attacks
- Block and Stream Ciphers

2

Cryptology

The scientific study of codes: creating, using, analyzing, and "breaking."

3

Cryptography and Cryptanalysis

- *Cryptography* - the creation and use of secret codes and related data security mechanisms.
- More generally, the study, theory, and implementation of security mechanisms based on the transformation of data.
- *Cryptanalysis* - the theory and practice of "breaking" cryptographic algorithms and protocols.
- "Breaking" means recovering protected text or otherwise bypassing security without knowledge of secret parameters.

4

Cryptographic Mechanisms (a la X.800)

Data Confidentiality

- *Encipherment*

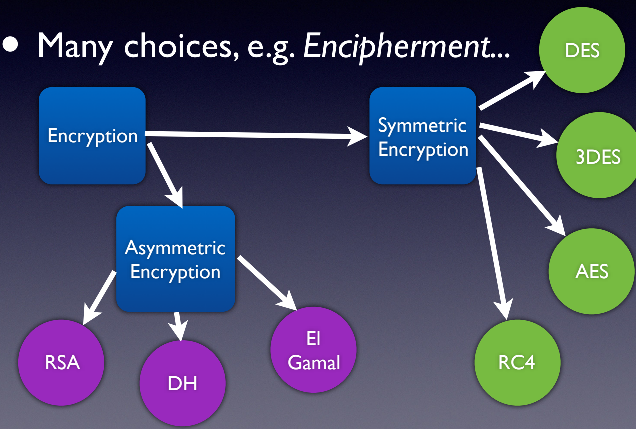
Data and System Integrity

- *Digital Signatures*
- *Data Integrity Mechanisms*
- *Authentication Exchange*

5

Cryptographic Algorithms

- Many choices, e.g. *Encipherment...*



6

In addition to learning about the primary cryptographic security mechanisms, we will study *utility mechanisms*:

- *Key Derivation* - derivation of secret keys for use in encipherment.
- *Random Number Generation* - generation of security-critical random values.

7

Symmetric Encryption

- The main ingredients:
  - *Plaintext* (message before encryption)
  - *Encryption algorithm*
  - *Secret key*
  - *Ciphertext* (message after encryption)
  - *Decryption algorithm*

8

Substitution Ciphers

- Suppose my *plaintext alphabet* consists of `A-Z`, `a-z`, `0-9`, period (`.`), and space.
- Let P be any permutation of the alphabet and P^{-1} the inverse of P.
- For a message $M = m_0\, m_1 \ldots m_{n-1}$, the encipherment is

$$Z = P(m_0)\, P(m_1) \ldots P(m_{n-1}).$$

- The message can be decrypted by applying P^{-1} in the same way to the characters in Z.

9

- The *plaintext* is just a message written in the alphabet.
- The *ciphertext* uses the same alphabet.
- The *secret key* is the permutation P.
- The *encryption algorithm* is the application of P to each character in the plaintext.
- The *decryption algorithm* is the application of the inverse permutation to each character in the ciphertext.

10

Exercise: Substitution

- In the previous example, the alphabet consists of 64 letters.
- How many different permutations of 64 letters are there?

11

Exercise continued...

- Suppose I intercept a message encrypted with an unknown permutation P.
- I've got a special computer that can try all possible permutations to find the one that decrypts the message. It can try one billion (10^9) permutations per second.
- How long will it take to find the answer?

12

That's called BRUTE FORCE.

It's the simplest type of cryptanalytic attack...but it's not always practical.

13

A Better Attack?

- Think back to Caesar Ciphers...wrack your brains...
- What is a better way to attack a substitution cipher?

14

That's called a CIPHERTEXT ONLY attack.

It's a *real* cryptanalytic attack, and in the case of a substitution cipher, is much more efficient than Brute Force.

There are other types of attacks...

15

Cryptanalytic Attacks

- Suppose we intercept ciphertext that we want to decrypt and that we know the details of the encryption algorithm.
- Cryptanalytic attacks are classified according to what *other* information is available to the attacker.

16

Type of Attack	Known to Cryptanalyst (in addition to algorithm and ciphertext)
Ciphertext only	No additional information.
Known plaintext	One or more plaintext/ciphertext pairs encrypted with the same key.
Chosen plaintext	Plaintext messages chosen by analyst along with corresponding ciphertexts encrypted with the same key.
Chosen ciphertext	Ciphertext messages chosen by analyst along with corresponding plaintexts decrypted with the same key.
Chosen text	Chosen plaintexts *and* chosen ciphertexts.

17

A Few Scenarios

- Substitution cipher: **ciphertext only**.
- SIM or smart card in a test harness: **chosen plaintext**.
- Stereotypical messages (e.g. encrypted PDF, Word, or structured messages: **known plaintext**.

Source: www.scmartcarddetective.com

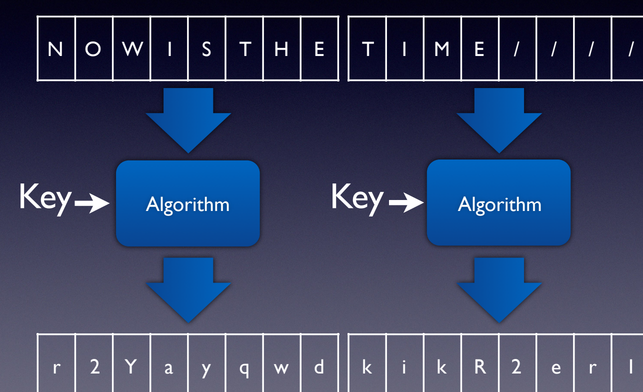
18

Block Ciphers

- Processes plaintext in fixed-size blocks (called the *block size)* and produces ciphertext in blocks of the same size.
- Typical block sizes are 64 and 128 bits.
- Examples: DES, 3DES, and AES

19

Generic Block Cipher



20

Stream Ciphers

- Algorithm (called a *key generator*) produces a pseudo-random string of bits or bytes.
- Pseudo-random stream is XORed with plaintext.
- Typically small and fast.
- Examples: RC4, A5/1

21

Generic Stream Cipher

Key Generator ← Key

Key Stream

Z	N	0	k	I	Q	B	D	m	U	q	K	…

$\oplus$ Plaintext

N	O	W	I	S	T	H	E	T	I	M	E

= Ciphertext

0x14	0x01	0x67	0x22	0x1a	0x05	0x0a	0x01	0x39	0x1c	0x3c	0x0e

22

Alice and Bob

- *Alice* and *Bob* are perpetually searching for ways to communicate securely.
- *Eve* wants to read their communications.
- Using block or stream ciphers, both Alice and Bob must have the same key *K*.
- This is a pain, because they have to meet to agree on the value of *K*…or trust FedEx to

23

Asymmetric Encryption

- Imagine a cryptographic system in which…
  - Alice has two algorithms E_A and D_A.
  - E_A is used to *encrypt* a message to Alice; she can share E_A freely.
  - D_A is used to *decrypt* a message encrypted with E_A; she needs to keep D_A secret.
- Similarly, Bob has E_B and D_B.

24

Alice | **Bob**

Publishes E_A → Receives E_A

Message M

$M = D_A(C)$ ← Ciphertext $C = E_A(M)$

Eve

Shouldn't be able to decrypt C or derive D_A from E_A.

Public Key Algorithms

- Through the magic of number theory…
  - Rivest-Shamir-Adleman-Cocks (RSA)
  - El Gamal
  - Diffie-Hellman (DH)
  - Elliptic Curve variants

Application: SSH

- Asymmetric encryption does *not* replace symmetric algorithms
  - Asymmetric (e.g. RSA) for authentication
  - Asymmetric (e.g. DH) for key agreement
  - Symmetric (e.g. AES) for data encryption

Other Algorithms

- There are a number of other important classes of algorithms:
  - Message Authentication and Hashing
  - Pseudo-Random Number Generation
  - Key Scheduling
- We will cover all of these to some extent.

Next time: block ciphers in detail.