

Principles of Information Security  
CS 5323 Lecture 11

Prof. William Winsborough  
October 23, 2008

## Business

- Return exams
- Next week we will start chapter 9 (reading)
- Preliminary project proposals due Tuesday, November 4 (1-2 pages)
  - Projects are of your own design
  - They can be programming or paper based
    - You can write a critique of one to three papers from the literature
    - You can present a paper in class
- Questions from previous lectures?

23 October 2008

Winsborough CS 5323 Lecture 11

2

## Exam Scores

A's	B's	C's
100	87	71
99	87	68
99	85	64
99	85	64
99	80	55
99	78	
98	75	
98		
97		
96		
95		
95		
94		
94		
93		
93		
92		
90		

23 October 2008

Winsborough CS 5323 Lecture 11

3

## DES Modes

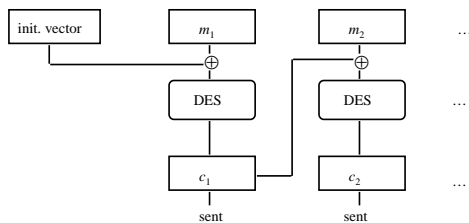
- Electronic Code Book Mode (ECB)
  - Encipher each block independently (rarely used)
- Cipher Block Chaining Mode (CBC)
  - Xor each block with previous ciphertext block
  - Requires an initialization vector for the first one
- Encrypt-Decrypt-Encrypt Mode (EDE - 2 keys:  $k, k'$ )
  - $c = \text{DES}_k(\text{DES}_{k'}^{-1}(\text{DES}_k(m)))$
- Encrypt-Encrypt-Encrypt Mode (triple DES - 3 keys:  $k, k', k''$ )
  - $c = \text{DES}_k(\text{DES}_{k'}(\text{DES}_{k''}(m)))$

23 October 2008

Winsborough CS 5323 Lecture 11

4

## CBC Mode Encryption

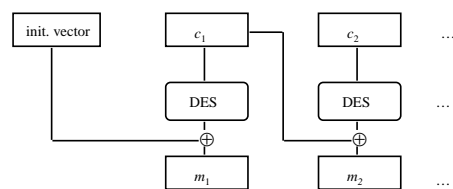


23 October 2008

Winsborough CS 5323 Lecture 11

5

## CBC Mode Decryption



23 October 2008

Winsborough CS 5323 Lecture 11

6

## Self-Healing Property

- Initial message
  - 3231343336353837 3231343336353837
  - 3231343336353837 3231343336353837
- Received as (underlined 4c should be 4b)
  - ef7c4cb2b4ce6f3b f6266e3a97af0e2c
  - 746ab9a6308f4256 33e60b451b09603d
- Which decrypts to
  - efca61e19f4836f1 3231333336353837
  - 3231343336353837 3231343336353837
  - Incorrect bytes underlined
  - Plaintext “heals” after 2 blocks

23 October 2008

Winsborough CS 5323 Lecture 11

7

## Public Key Cryptography

- Two keys
  - *Private key* known only to individual
  - *Public key* available to anyone
    - Public key, private key inverses
- Idea
  - Confidentiality: encipher using public key, decipher using private key
  - Integrity/authentication: encipher using private key, decipher using public one

23 October 2008

Winsborough CS 5323 Lecture 11

8

## Requirements

1. It must be computationally easy to encipher or decipher a message given the appropriate key
2. It must be computationally infeasible to derive the private key from the public key
3. It must be computationally infeasible to determine the private key from a chosen plaintext attack

23 October 2008

Winsborough CS 5323 Lecture 11

9

## RSA

- Exponentiation cipher
- Relies on the difficulty of determining the number of numbers relatively prime to a large integer  $n$

23 October 2008

Winsborough CS 5323 Lecture 11

10

## Background

- Totient function  $\phi(n)$ 
  - Number of positive integers less than  $n$  and relatively prime to  $n$ 
    - *Relatively prime* means with no factors in common with  $n$
- Example:  $\phi(10) = 4$ 
  - 1, 3, 7, 9 are relatively prime to 10
- Example:  $\phi(21) = 12$ 
  - 1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20 are relatively prime to 21

23 October 2008

Winsborough CS 5323 Lecture 11

11

## Algorithm

- Choose two large prime numbers  $p, q$ 
  - Let  $n = pq$ ; then  $\phi(n) = (p-1)(q-1)$
  - Choose  $e < n$  such that  $e$  is relatively prime to  $\phi(n)$ .
  - Compute  $d$  such that  $ed \bmod \phi(n) = 1$
- Public key:  $(e, n)$ ; private key:  $d$
- Encipher:  $c = m^e \bmod n$
- Decipher:  $m = c^d \bmod n$

23 October 2008

Winsborough CS 5323 Lecture 11

12

## Example: Confidentiality

- Take  $p = 7$ ,  $q = 11$ , so  $n = 77$  and  $\phi(n) = 60$
- Alice chooses  $e = 17$ , making  $d = 53$
- Bob wants to send Alice secret message HELLO (07 04 11 11 14)
  - $07^{17} \bmod 77 = 28$
  - $04^{17} \bmod 77 = 16$
  - $11^{17} \bmod 77 = 44$
  - $11^{17} \bmod 77 = 44$
  - $14^{17} \bmod 77 = 42$
- Bob sends 28 16 44 44 42

23 October 2008

Winsborough CS 5323 Lecture 11

13

## Example

- Alice receives 28 16 44 44 42
- Alice uses private key,  $d = 53$ , to decrypt message:
  - $28^{53} \bmod 77 = 07$
  - $16^{53} \bmod 77 = 04$
  - $44^{53} \bmod 77 = 11$
  - $44^{53} \bmod 77 = 11$
  - $42^{53} \bmod 77 = 14$
- Alice translates message to letters to read HELLO
  - No one else could read it, as only Alice knows her private key and that is needed for decryption

23 October 2008

Winsborough CS 5323 Lecture 11

14

## Example: Integrity/Authentication

- Take  $p = 7$ ,  $q = 11$ , so  $n = 77$  and  $\phi(n) = 60$
- Alice chooses  $e = 17$ , making  $d = 53$
- Alice wants to send Bob message HELLO (07 04 11 11 14) so Bob knows it is what Alice sent (no changes in transit, and authenticated)
  - $07^{53} \bmod 77 = 35$
  - $04^{53} \bmod 77 = 09$
  - $11^{53} \bmod 77 = 44$
  - $11^{53} \bmod 77 = 44$
  - $14^{53} \bmod 77 = 49$
- Alice sends 35 09 44 44 49

23 October 2008

Winsborough CS 5323 Lecture 11

15

## Example

- Bob receives 35 09 44 44 49
- Bob uses Alice's public key,  $e = 17$ ,  $n = 77$ , to decrypt message:
  - $35^{17} \bmod 77 = 07$
  - $09^{17} \bmod 77 = 04$
  - $44^{17} \bmod 77 = 11$
  - $44^{17} \bmod 77 = 11$
  - $49^{17} \bmod 77 = 14$
- Bob translates message to letters to read HELLO
  - Alice sent it as only she knows her private key, so no one else could have enciphered it
  - If (enciphered) message's blocks (letters) altered in transit, would not decrypt properly

23 October 2008

Winsborough CS 5323 Lecture 11

16

## Example: Both

- Alice wants to send Bob message HELLO both enciphered and authenticated (integrity-checked)
  - Alice's keys: public (17, 77); private: 53
  - Bob's keys: public: (37, 77); private: 13
- Alice enciphers HELLO (07 04 11 11 14):
  - $(07^{53} \bmod 77)^{37} \bmod 77 = 07$
  - $(04^{53} \bmod 77)^{37} \bmod 77 = 37$
  - $(11^{53} \bmod 77)^{37} \bmod 77 = 44$
  - $(11^{53} \bmod 77)^{37} \bmod 77 = 44$
  - $(14^{53} \bmod 77)^{37} \bmod 77 = 14$
- Alice sends 07 37 44 44 14

23 October 2008

Winsborough CS 5323 Lecture 11

17

## Security Services

- Confidentiality
  - Only the owner of the private key knows it, so text enciphered with public key cannot be read by anyone except the owner of the private key
- Authentication
  - Only the owner of the private key knows it, so text enciphered with private key must have been generated by the owner

23 October 2008

Winsborough CS 5323 Lecture 11

18

## More Security Services

- Integrity
  - Enciphered letters cannot be changed undetectably without knowing private key
- Non-Repudiation
  - Message enciphered with private key came from someone who knew it

23 October 2008

Winsborough CS 5323 Lecture 11

19

## Warnings

- Encipher message in blocks considerably larger than the examples here
  - If 1 character per block, RSA can be broken using statistical attacks (just like classical cryptosystems)
  - Attacker cannot alter letters, but can rearrange them and alter message meaning
    - Example: reverse enciphered message of text ON to get NO

23 October 2008

Winsborough CS 5323 Lecture 11

20

## Cryptographic Checksums

- Mathematical function to generate a set of  $k$  bits from a set of  $n$  bits (where  $k \leq n$ ).
  - $k$  is smaller than  $n$  except in unusual circumstances
- Example: ASCII parity bit
  - ASCII has 7 bits; 8th bit is “parity”
  - Even parity: even number of 1 bits
  - Odd parity: odd number of 1 bits

23 October 2008

Winsborough CS 5323 Lecture 11

21

## Example Use

- Bob receives “10111101” as bits.
  - Sender is using even parity; 6 1 bits, so character was received correctly
    - Note: could be garbled, but 2 bits would need to have been changed to preserve parity
  - Sender is using odd parity; even number of 1 bits, so character was not received correctly

23 October 2008

Winsborough CS 5323 Lecture 11

22

## Definition

- Cryptographic checksum  $h: A \rightarrow B$ :
  1. For any  $x \in A$ ,  $h(x)$  is easy to compute
  2. For any  $y \in B$ , it is computationally infeasible to find  $x \in A$  such that  $h(x) = y$
  3. It is computationally infeasible to find two inputs  $x, x' \in A$  such that  $x \neq x'$  and  $h(x) = h(x')$ 
    - Alternate form (stronger): Given any  $x \in A$ , it is computationally infeasible to find a different  $x' \in A$  such that  $h(x) = h(x')$ .

23 October 2008

Winsborough CS 5323 Lecture 11

23

## Collisions

- If  $x \neq x'$  and  $h(x) = h(x')$ ,  $x$  and  $x'$  are a *collision*
  - Pigeonhole principle: if there are  $n$  containers for  $n+1$  objects, then at least one container will have 2 objects in it.
  - Application: if there are 32 files and 8 possible cryptographic checksum values, at least one value corresponds to at least 4 files

23 October 2008

Winsborough CS 5323 Lecture 11

24

## Keys

- Keyed cryptographic checksum: requires cryptographic key
  - DES in chaining mode: encipher message, use last  $n$  bits. Requires a key to encipher, so it is a keyed cryptographic checksum.
- Keyless cryptographic checksum: requires no cryptographic key
  - MD5 and SHA-1 are best known; others include MD4, HAVAL, and Snefru

23 October 2008

Winsborough CS 5323 Lecture 11

25

## HMAC

- Make keyed cryptographic checksums from keyless cryptographic checksums
- $h$  keyless cryptographic checksum function that takes data in blocks of  $b$  bytes and outputs blocks of  $l$  bytes.  $k'$  is cryptographic key of length  $b$  bytes
  - If short, pad with 0 bytes; if long, hash to length  $b$
- $ipad$  is 00110110 repeated  $b$  times
- $opad$  is 01011100 repeated  $b$  times
- $HMAC-h(k, m) = h(k' \oplus opad || h(k' \oplus ipad || m))$ 
  - $\oplus$  exclusive or,  $||$  concatenation

23 October 2008

Winsborough CS 5323 Lecture 11

26

## Key Points

- Two main types of cryptosystems: classical and public key
- Classical cryptosystems encipher and decipher using the same key
  - Or one key is easily derived from the other
- Public key cryptosystems encipher and decipher using different keys
  - Computationally infeasible to derive one from the other
- Cryptographic checksums provide a check on integrity

23 October 2008

Winsborough CS 5323 Lecture 11

27