



Practical Aspects of Modern Cryptography

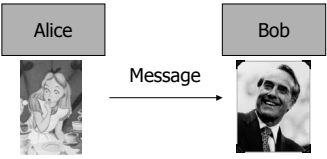
Josh Benaloh & Brian LaMacchia





Lecture 3: Symmetric Ciphers and Hash Functions




Meet Alice and Bob



Alice  → Message →  Bob


January 22, 2002 Practical Aspects of Modern Cryptography 3



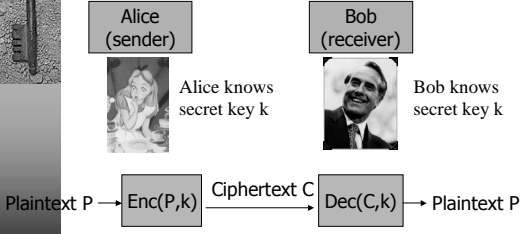
Modern Symmetric Ciphers


- ◆ Setup: Alice wants to send a private message to Bob.
- ◆ Precondition: Alice and Bob **have previously shared some secret known only to them.**
- ◆ The pre-shared secret is the encryption key Alice and Bob will use.


January 22, 2002 Practical Aspects of Modern Cryptography 4



Symmetric Encryption

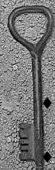


Alice (sender)  Alice knows secret key k

Bob (receiver)  Bob knows secret key k

Plaintext P → $Enc(P,k)$ → Ciphertext C → $Dec(C,k)$ → Plaintext P

January 22, 2002 Practical Aspects of Modern Cryptography 5



What makes a cipher secure?

- ◆ Exhaustive search of keyspace must be infeasible
 - More about this later...
- ◆ It must also be infeasible to find the key given:
 - Sample ciphertext and corresponding plaintext (“known-plaintext attack”)
 - The ability to feed ciphertext in and see what plaintext comes out (“chosen-ciphertext attack”)
 - or the other way around (“chosen-plaintext attack”)
- ◆ If someone can find keys under any of these conditions, the cipher isn’t considered secure

January 22, 2002 Practical Aspects of Modern Cryptography 6

Some bad cipher ideas

- ◆ Repeated XOR mask
 - Pick a key number. XOR with each plaintext
 - XOR with key again to decrypt
 - INSECURE: just one plaintext/ciphertext gives the key
- ◆ Monoalphabetic substitution
 - Key is a table of letters and corresponding ciphertexts
 - a=m, b=x, c=b, d=r, etc.
 - encrypt/decrypt by substitution
 - Exhaustive search is really hard (26! keys to try).
 - INSECURE: statistical analysis of ciphertext frequency

January 22, 2002 Practical Aspects of Modern Cryptography 7

Symmetric Ciphers

Private-key (symmetric) ciphers are usually divided into two classes.

- ◆ Block ciphers
- ◆ Stream ciphers

January 22, 2002 Practical Aspects of Modern Cryptography 8

Symmetric Ciphers

Private-key (symmetric) ciphers are usually divided into two classes.

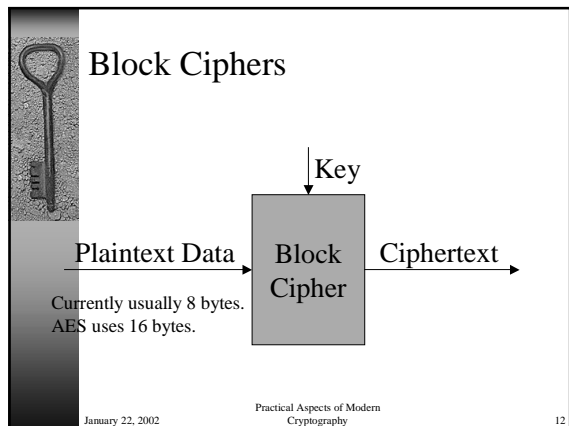
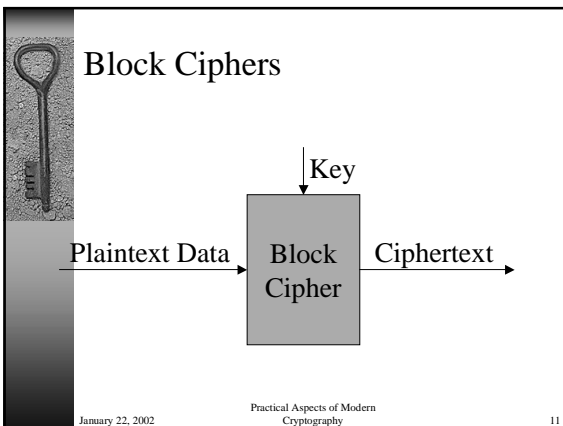
- ◆ Block ciphers
- ◆ Stream ciphers

January 22, 2002 Practical Aspects of Modern Cryptography 9

Block ciphers

- ◆ Encrypt fixed-size blocks
 - $\text{ciphertext} = \text{Encrypt}(\text{key}, \text{cleartext})$
 - $\text{cleartext} = \text{Decrypt}(\text{key}, \text{ciphertext})$
- ◆ Encrypt function converts blocks of cleartext bits to ciphertext bits
 - Decrypt function converts back
- ◆ If the key is wrong, you get the wrong result
- ◆ Shouldn't be possible to derive key given cleartext, ciphertext pairs
- ◆ Examples include DES, 3DES, AES

January 22, 2002 Practical Aspects of Modern Cryptography 10



Fast Facts about Block Ciphers

- ◆ Encrypt/decrypt data a block at a time
- ◆ Encryption/decryption of sequential blocks may be related

Mode of operation

- ◆ We always encrypt/decrypt full blocks
 - No partial blocks allowed by the cipher!
 - Our plaintext may not be an even multiple of blocks, so we may need to pad the last plaintext block

January 22, 2002 Practical Aspects of Modern Cryptography 13

Block Cipher Modes

Electronic Code Book (ECB) Encryption:

January 22, 2002 Practical Aspects of Modern Cryptography 14

Block Cipher Modes

Electronic Code Book (ECB) Decryption:

January 22, 2002 Practical Aspects of Modern Cryptography 15

Problems with ECB

- ◆ Patterns in plaintext preserved in ciphertext

- ◆ No basic integrity protection. Must add or:
 - Cipher block substitution and rearrangement attacks
 - Fabrication of information

January 22, 2002 Practical Aspects of Modern Cryptography 16

Block Cipher Modes

Cipher Block Chaining (CBC) Encryption:

January 22, 2002 Practical Aspects of Modern Cryptography 17

Block Cipher Modes

Cipher Block Chaining (CBC) Decryption:

January 22, 2002 Practical Aspects of Modern Cryptography 18

How to Build a Block Cipher

Plaintext →

Key →

Block Cipher

→ Ciphertext

January 22, 2002 19

Practical Aspects of Modern Cryptography

Feistel Ciphers

Diagram illustrating a single Feistel round. Two horizontal bars represent the left and right halves of the block. Arrows show a cross-swapping of the halves. A box labeled 'Ugly' is connected to the right half, and a circle with a plus sign is connected to the left half.

January 22, 2002 20

Practical Aspects of Modern Cryptography

Feistel Ciphers

Diagram illustrating a single Feistel round. Two horizontal bars represent the left and right halves of the block. Arrows show a cross-swapping of the halves. A box labeled 'Ugly' is connected to the right half, and a circle with a plus sign is connected to the left half.

January 22, 2002 21

Practical Aspects of Modern Cryptography

Feistel Ciphers

Diagram illustrating a single Feistel round. Two horizontal bars represent the left and right halves of the block. Arrows show a cross-swapping of the halves. A box labeled 'Ugly' is connected to the right half, and a circle with a plus sign is connected to the left half.

January 22, 2002 22

Practical Aspects of Modern Cryptography

Feistel Ciphers

Diagram illustrating a single Feistel round. Two horizontal bars represent the left and right halves of the block. Arrows show a cross-swapping of the halves. A box labeled 'Ugly' is connected to the right half, and a circle with a plus sign is connected to the left half.

January 22, 2002 23

Practical Aspects of Modern Cryptography

Feistel Ciphers

Diagram illustrating two Feistel rounds. Two horizontal bars represent the left and right halves of the block. Arrows show a cross-swapping of the halves. A box labeled 'Ugly' is connected to the right half, and a circle with a plus sign is connected to the left half. This structure is repeated for a second round below the first.

January 22, 2002 24

Practical Aspects of Modern Cryptography

Feistel Ciphers

- Typically, most Feistel ciphers are iterated for about 16 rounds.
- Different “sub-keys” are used for each round. Sub-keys are derived from the master key or a derived key schedule
- Even a weak round function can yield a strong Feistel cipher if iterated sufficiently.

January 22, 2002 Practical Aspects of Modern Cryptography 25

Data Encryption Standard (DES)

64-bit Plaintext

56-bit Key

Block Cipher

64-bit Ciphertext

January 22, 2002 Practical Aspects of Modern Cryptography 26

Data Encryption Standard (DES)

64-bit Plaintext

56-bit Key

16 Feistel Rounds

64-bit Ciphertext

January 22, 2002 Practical Aspects of Modern Cryptography 27

Data Encryption Standard (DES)

64-bit Plaintext

56-bit Key

16 Feistel Rounds

64-bit Ciphertext

January 22, 2002 Practical Aspects of Modern Cryptography 28

DES Round

Ugly

January 22, 2002 Practical Aspects of Modern Cryptography 29

Simplified DES Round Function

32 bits

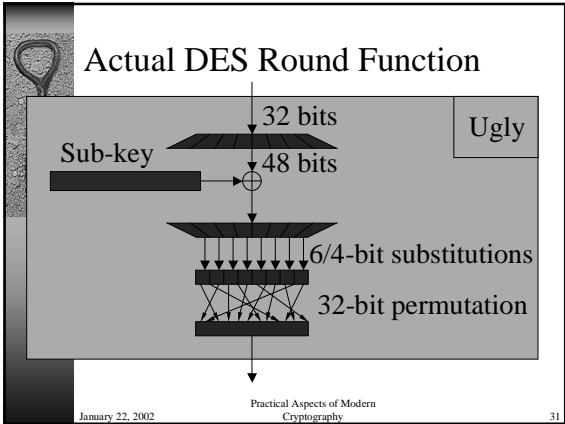
Ugly

Sub-key

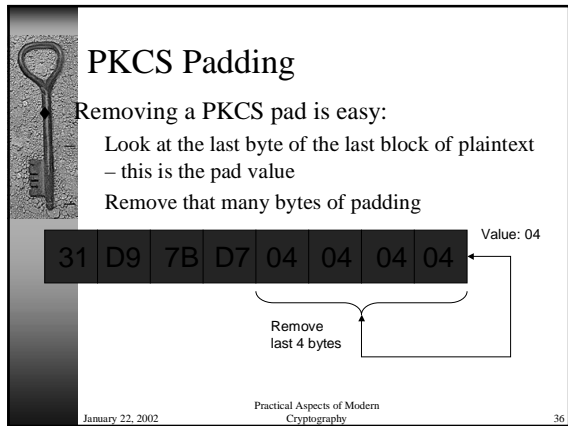
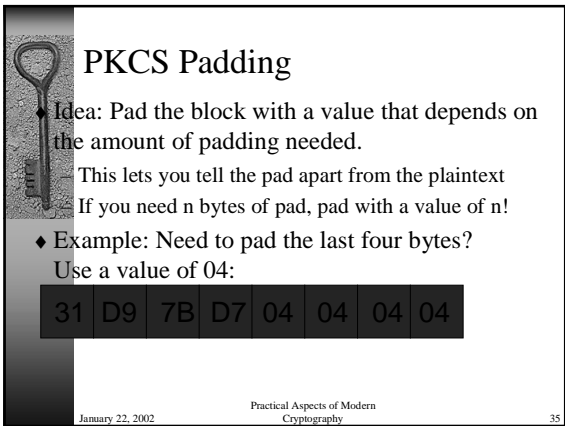
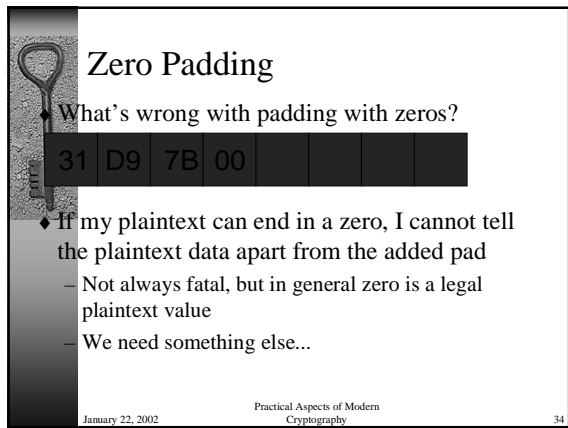
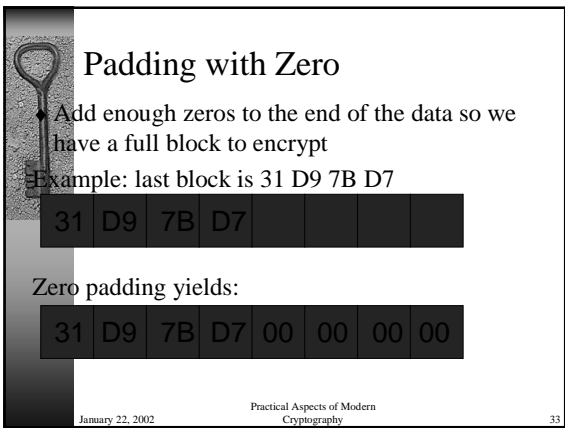
4-bit substitutions

32-bit permutation

January 22, 2002 Practical Aspects of Modern Cryptography 30



- ### Padding Modes
- What do we do if the length of the plaintext is not an even multiple of the cipher's block size?
 - A: Drop the extra data on the floor (You really didn't want it encrypted anyway)
 - B: Throw an exception/return an error "User error"
 - C: Pad the last block of plaintext so it's a full block, then encrypt it normally
- January 22, 2002 Practical Aspects of Modern Cryptography 32



PKCS Padding

- ◆ What did I forget?
- ◆ What happens if the plaintext doesn't need to be padded? (Last block is already full...)

A7 9D 42 46 BE D4 37 B8

Value: B8 (decimal 184)

Remove last 184 bytes!

January 22, 2002 Practical Aspects of Modern Cryptography 37

PKCS Padding

- ◆ If the last block is a full block, add an entire block's worth of padding:

A7 9D 42 46 BE D4 37 B8

08 08 08 08 08 08 08 08

Value: 08

Remove last 8 bytes (entire block)

January 22, 2002 Practical Aspects of Modern Cryptography 38

Block Ciphers in Use Today

- ◆ DES is not secure
 - DES can be brute-forced (2^{56} operations)
 - In January 1998, a combination of the EFF DES Cracker and distributed.net brute-forced a challenge in less than 24 hours
- ◆ Today, the common choices are:
 - Triple-DES (two-key or three-key)
 - AES (Rijndael)

January 22, 2002 Practical Aspects of Modern Cryptography 39

Triple-DES (3DES)

- ◆ Triple-DES is DES run three times
 - Sometimes called 3DES-EDE because it has three stages: encryption-decryption-encryption

```

    graph LR
      Plaintext[Plaintext] --> E1[DES Encryption]
      E1 --> D1[DES Decryption]
      D1 --> E2[DES Encryption]
      E2 --> Ciphertext[Ciphertext]
      K1[Key K1] --> E1
      K2[Key K2] --> D1
      K3[Key K3] --> E2
  
```

January 22, 2002 Practical Aspects of Modern Cryptography 40

Triple-DES (3DES)

```

    graph LR
      Plaintext[Plaintext] --> E1[DES Encryption]
      E1 --> D1[DES Decryption]
      D1 --> E2[DES Encryption]
      E2 --> Ciphertext[Ciphertext]
      K1[Key K1] --> E1
      K2[Key K2] --> D1
      K3[Key K3] --> E2
  
```


- ◆ Triple-DES can be run in either two-key or three-key modes
 - In two-key mode, $K_1 = K_3$
 - In three-key mode, K_1, K_2, K_3 are all distinct
- ◆ If $K_1 = K_2 = K_3$, you have just DES

January 22, 2002 Practical Aspects of Modern Cryptography 41

Be Skeptical of New Ciphers

- ◆ The details are everything
 - And there are a lot of details
 - Cipher design is much harder than you'd think
- ◆ Don't ever design and use your own cipher!
 - It won't be nearly as secure as existing designs like 3DES
- ◆ Don't trust secret algorithms
 - Need lots of review by skeptical experts to gain confidence in a cipher

January 22, 2002 Practical Aspects of Modern Cryptography 42




Symmetric Ciphers

Private-key (symmetric) ciphers are usually divided into two classes.

- ◆ Block ciphers
- ◆ Stream ciphers

January 22, 2002 Practical Aspects of Modern Cryptography 43



Background: The One-Time Pad

- ◆ An unconditionally secure cipher


Key = random bits = 1100010011100100011...

Message = bits = 1110011001100110001...

Ciphertext = XOR of Key, Message = 0010001010000010010...

- ◆ Problems:
 - Number of random bits needed = sum of lengths of all messages to be encrypted (not reusable)
 - Random bits must be known to both sender and recipient.

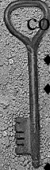
January 22, 2002 Practical Aspects of Modern Cryptography 44



Random Numbers

- ◆ Really good random numbers are hard to acquire
 - Best bits come from physical systems
 - Radioactive decay (<http://www.fourmilab.ch/hotbits/>)
 - Noise diodes
 - Lava Lamps
 - Getting many truly random bits is slow
 - Getting many shared truly random bits is more awkward
- ◆ Getting “good randomness” is important for many crypto algorithms
 - Picking private key components & secret keys
 - Some algorithms (e.g. DSA) require random input!


January 22, 2002 Practical Aspects of Modern Cryptography 45



So how do we get random numbers on a computer?

- ◆ It sounds so easy: “Just pick some random bytes”
- ◆ No good standard source of computer randomness
 - OS state (time-of-day, PID) is very low entropy
 - User keyboard input is very unreliable
- ◆ Best practical options aren’t very good
 - Inter-event timing (keyboard, network), timing loops, fast clocks and interval timers
 - Better would be */dev/random*, or hardware generator
 - Intel 850 chipset (for Pentium motherboards) has on-board hardware RNG


January 22, 2002 Practical Aspects of Modern Cryptography 46



Pseudo-Random Numbers

- ◆ How do we make a lot of “good” random bits from a smaller number of “really good” random bits?
 - We want “pseudo-random bits”
- ◆ Pseudo-random bitstrings are “polynomial time indistinguishable” from truly random bitstrings
- ◆ In practice: use DES, hash functions to generate bits from a random seed (FIPS 186)


January 22, 2002 Practical Aspects of Modern Cryptography 47



Stream Ciphers


- ◆ Use the secret key as a seed to a pseudo-random number-generator.
- ◆ Take the stream of output bits from the PRNG and XOR it with the plaintext to form the ciphertext.


January 22, 2002 Practical Aspects of Modern Cryptography 48

 **Stream ciphers**


- ◆ Generate mask bits
 - $\text{ciphertext}[i] = \text{cleartext}[i] + \text{stream}(\text{key}, \text{state})$
 - $\text{cleartext}[i] = \text{ciphertext}[i] - \text{stream}(\text{key}, \text{state})$
- ◆ Cipher produces a sequence of bits that is added to the cleartext to produce ciphertext
 - Receiver can generate the same sequence and subtract from ciphertext to recover cleartext
- ◆ Must never re-use same part of stream
- ◆ Each bit is encrypted independently

January 22, 2002 Practical Aspects of Modern Cryptography 49


 **Stream Cipher Encryption**

Plaintext: 


$\oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus$


PRNG(seed): 

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓


Ciphertext: 

January 22, 2002 Practical Aspects of Modern Cryptography 50


 **Stream Cipher Decryption**

Plaintext: 


↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

PRNG(seed): 

$\oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus \oplus$

Ciphertext: 

January 22, 2002 Practical Aspects of Modern Cryptography 51

 **A PRNG: Alleged RC4**

Initialization

$S[0..255] = 0, 1, \dots, 255$


$K[0..255] = \text{Key}, \text{Key}, \text{Key}, \dots$

for $i = 0$ to 255

$j = (j + S[i] + K[i]) \bmod 256$

swap $S[i]$ and $S[j]$

January 22, 2002 Practical Aspects of Modern Cryptography 52

 **A PRNG: Alleged RC4**

Iteration

$i = (i + 1) \bmod 256$


$j = (j + S[i]) \bmod 256$

swap $S[i]$ and $S[j]$

$t = (S[i] + S[j]) \bmod 256$

Output $S[t]$


January 22, 2002 Practical Aspects of Modern Cryptography 53

 **Stream Cipher Integrity**

- ◆ It is easy for an adversary (even one who can't decrypt the ciphertext) to alter the plaintext in a known way.

Bob to Bob's Bank: Please transfer \$0,000,002.00 to the account of my good friend Alice.

January 22, 2002 Practical Aspects of Modern Cryptography 54




Stream Cipher Integrity

- It is easy for an adversary (even one who can't decrypt the ciphertext) to alter the plaintext in a known way.

Bob to Bob's Bank: Please transfer \$1,000,002.00 to the account of my good friend Alice.

January 22, 2002 Practical Aspects of Modern Cryptography 55




Stream Cipher Integrity

- It is easy for an adversary (even one who can't decrypt the ciphertext) to alter the plaintext in a known way.

Bob to Bob's Bank: Please transfer \$1,000,002.00 to the account of my good friend Alice.

- This can be protected against by the careful addition of appropriate redundancy.


January 22, 2002 Practical Aspects of Modern Cryptography 56



One-Way Hash Functions

- The idea of a *check sum* is great, but it is designed to prevent accidental changes in a message.
- For cryptographic integrity, we need an integrity check that is resilient against a smart and determined adversary.


January 22, 2002 Practical Aspects of Modern Cryptography 57



One-Way Hash Functions

Generally, a *one-way hash function* is a function $H : \{0,1\}^* \rightarrow \{0,1\}^k$ (typically k is 128 or 160) such that given an input value x , one cannot find a value $x' \neq x$ such $H(x) = H(x')$.

January 22, 2002 Practical Aspects of Modern Cryptography 58

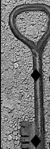


One-Way Hash Functions

There are many measures for one-way hashes.

- Non-invertability:** given y , it's difficult to find any x such that $H(x) = y$.
- Collision-intractability:** one cannot find a pair of values $x' \neq x$ such that $H(x) = H(x')$.

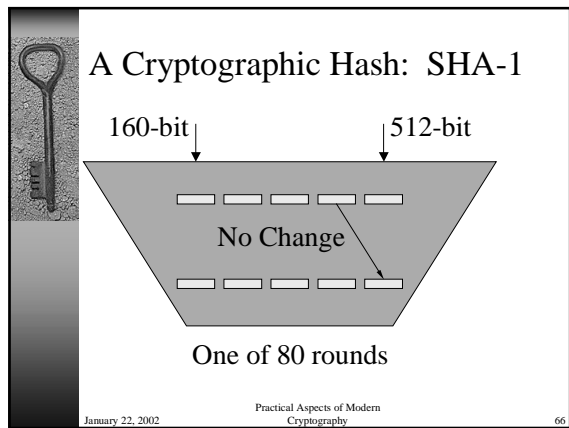
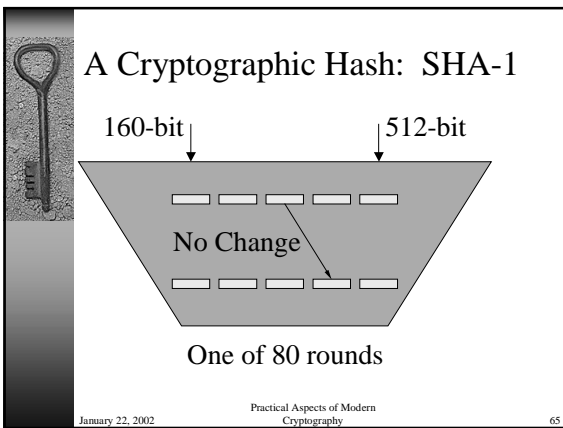
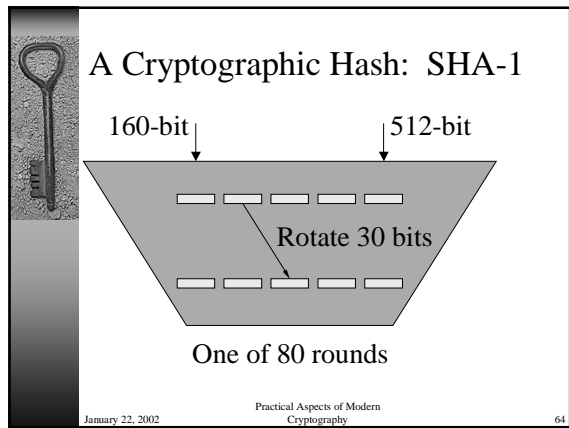
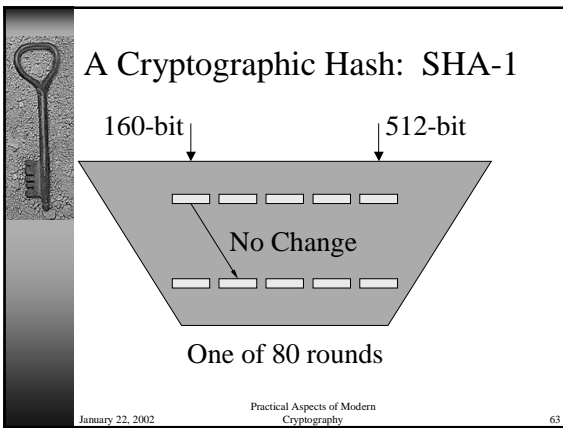
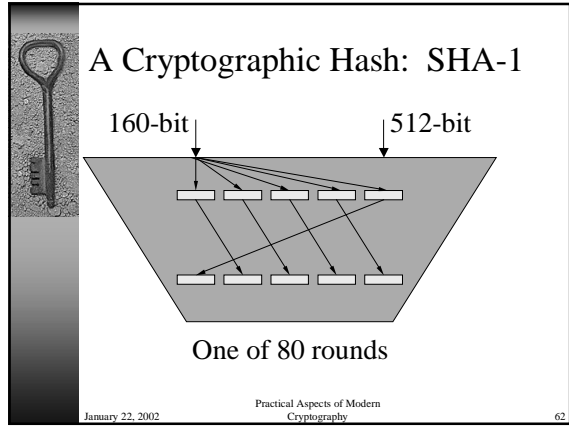
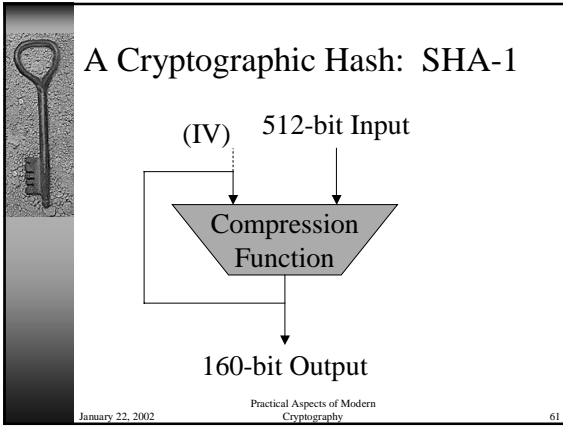
January 22, 2002 Practical Aspects of Modern Cryptography 59



An Example Hash: SHA-1

- SHA-1 was designed by the US Government as part of the Digital Signature Standard
- SHA-1 is the most-commonly used hash function today
 - It's the hash function in which we have the most faith right now
- SHA-1 takes any size input and produces a 160-bit output (the digest value)

January 22, 2002 Practical Aspects of Modern Cryptography 60



A Cryptographic Hash: SHA-1

160-bit 512-bit

?

One of 80 rounds

January 22, 2002 Practical Aspects of Modern Cryptography 67

A Cryptographic Hash: SHA-1

What's in the final 32-bit transform?

- ◆ Take the rightmost word.
- ◆ Add in the leftmost word rotated 5 bits.
- ◆ Add in a round-dependent function f of the middle three words.

January 22, 2002 Practical Aspects of Modern Cryptography 68

A Cryptographic Hash: SHA-1

160-bit 512-bit

f

One of 80 rounds

January 22, 2002 Practical Aspects of Modern Cryptography 69

A Cryptographic Hash: SHA-1

Depending on the round, the “non-linear” function f is one of the following.

$$f(X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$$

$$f(X, Y, Z) = (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z)$$

$$f(X, Y, Z) = X \oplus Y \oplus Z$$

January 22, 2002 Practical Aspects of Modern Cryptography 70

A Cryptographic Hash: SHA-1

What's in the final 32-bit transform?

- ◆ Take the rightmost word.
- ◆ Add in the leftmost word rotated 5 bits.
- ◆ Add in a round-dependent function f of the middle three words.

January 22, 2002 Practical Aspects of Modern Cryptography 71

A Cryptographic Hash: SHA-1

What's in the final 32-bit transform?

- ◆ Take the rightmost word.
- ◆ Add in the leftmost word rotated 5 bits.
- ◆ Add in a round-dependent function f of the middle three words.
- ◆ Add in a round-dependent constant.

January 22, 2002 Practical Aspects of Modern Cryptography 72

A Cryptographic Hash: SHA-1

What's in the final 32-bit transform?

- ◆ Take the rightmost word.
- ◆ Add in the leftmost word rotated 5 bits.
- ◆ Add in a round-dependent function f of the middle three words.
- ◆ Add in a round-dependent constant.
- ◆ Add in a portion of the 512-bit message.

January 22, 2002 Practical Aspects of Modern Cryptography 73

A Cryptographic Hash: SHA-1

160-bit 512-bit

One of 80 rounds

January 22, 2002 Practical Aspects of Modern Cryptography 74

One-Way Hash Functions

- ◆ When using a stream cipher, a hash of the message can be appended to ensure integrity. [Message Authentication Code]
- ◆ When forming a digital signature, the signature need only be applied to a hash of the message. [Message Digest]

January 22, 2002 Practical Aspects of Modern Cryptography 75