CISC 3325 - INFORMATION SECURITY

Toolbox: Cryptography

Topics for today

- Cryptography:
  - Problems encryption is designed to solve
  - Encryption tools categories, strengths, weaknesses
    - applications of each
  - Certificates and certificate authorities

CRYPTOGRAPHY

https://www.tripwire.com/state-of-security/security-data-protection/cryptography/ordinary-people-need-cryptography/

Encryption

- Encoding a message so that its meaning is not obvious

Problems Addressed by Encryption

- Suppose a sender wants to send a message to a recipient. An attacker may attempt to:
  - Block the message
  - Intercept the message
  - Modify the message
  - Fabricate an authentic-looking alternate message
- Encryption can address all of these problems

Encryption Terminology

- Sender
- Recipient
- Transmission medium
- Interceptor/intruder
- Encrypt, encode, or encipher
  - Process that hides the meaning of the message
- Decrypt, decode, or decipher
  - Reveal the original message

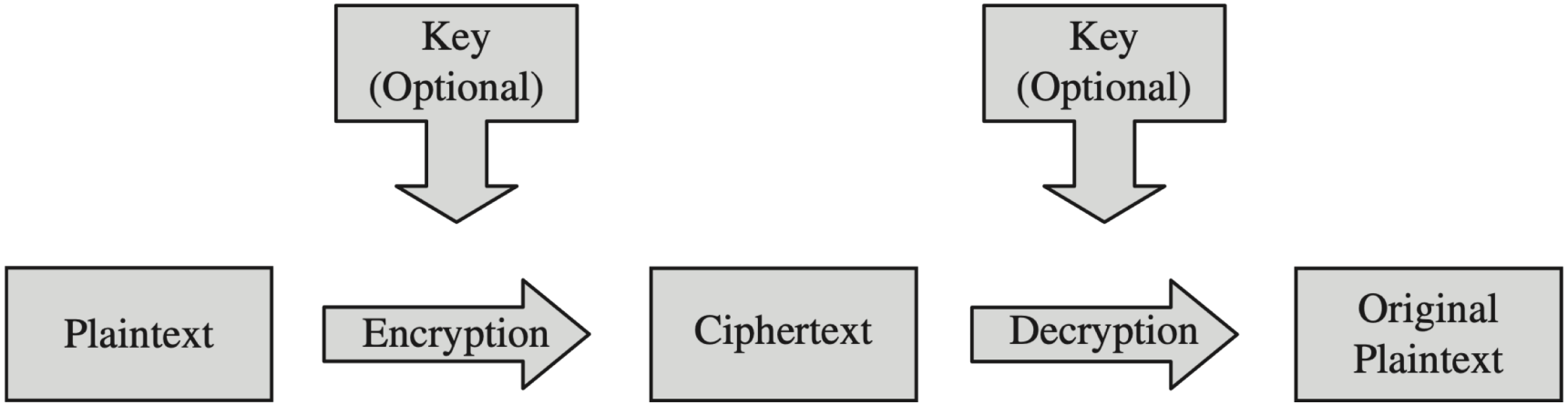
Encryption Terminology

- Cryptosystem
  - A system for encryption and decryption
- Plaintext
  - Original message
- Ciphertext
  - Encrypted message

Encryption Keys

- A cryptosystem involves a set of rules for how to encrypt a plaintext and decrypt the ciphertext
  - Rules == algorithms
- Typically uses a device called a **key (*K*)**
- The resulting ciphertext depends on:
  - The original message
  - The algorithm
  - The key value
    - => the original message is secret, the algorithm typically known
- An encryption key that does not require key is called *keyless cipher*

Encryption/Decryption Process



Basics

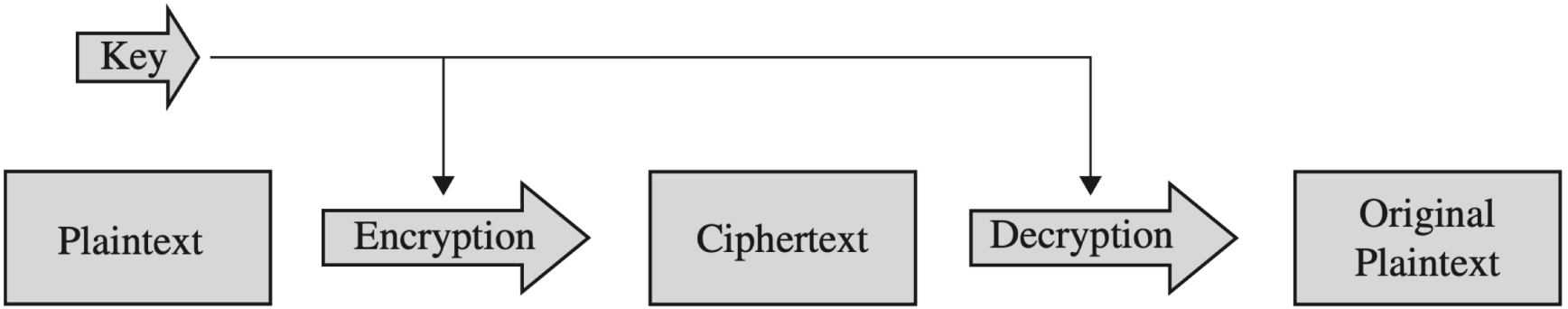
- Notation
  - Secret key K
  - Encryption function EK(P)
  - Decryption function DK(C)
  - Plaintext length typically same as ciphertext length
  - Encryption and decryption are permutation functions (bijections) on the set of all n-bit arrays

http://www.atksolutions.com/articles/basics_on_how_encryption_algorithm_works.html

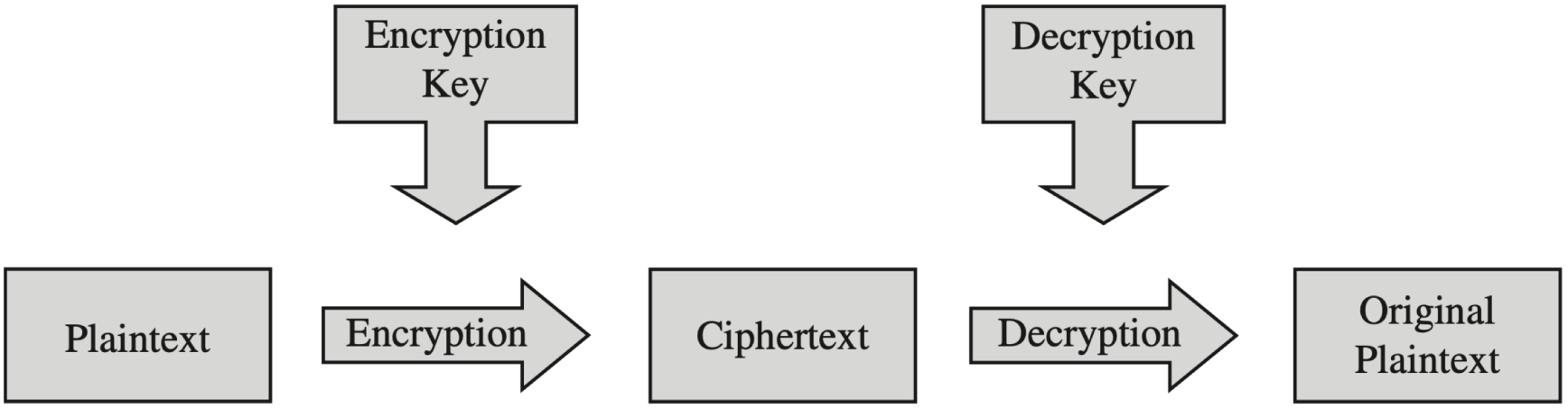
Symmetric and Asymmetric Encryption

- Symmetric encryption uses the same key, K, both to encrypt a message and later to decrypt it
  - Also called single-key or secret key encryption
  - D and E are mirror-image processes
- Asymmetric encryption uses a pair of keys
  - A decryption key, KD, inverts the encryption of key KE, so that $P = D(K_D, E(K_E,P))$

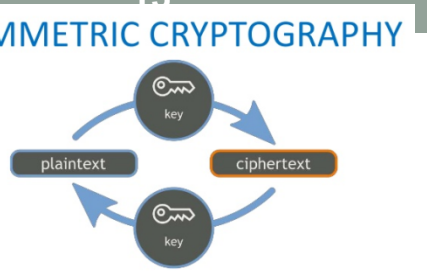
Symmetric vs. Asymmetric



(a) Symmetric Cryptosystem



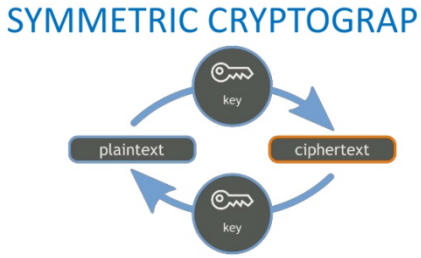
(b) Asymmetric Cryptosystem



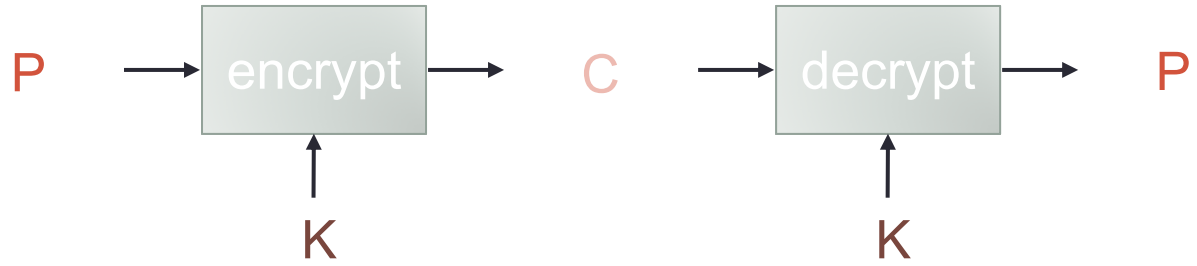
Symmetric Cryptosystem

- Scenario
  - Alice wants to send a message (plaintext P) to Bob.
  - The communication channel is insecure and can be eavesdropped
  - If Alice and Bob have previously agreed on a symmetric encryption scheme and a secret key K, the message can be sent encrypted (ciphertext C)

Symmetric Cryptosystem



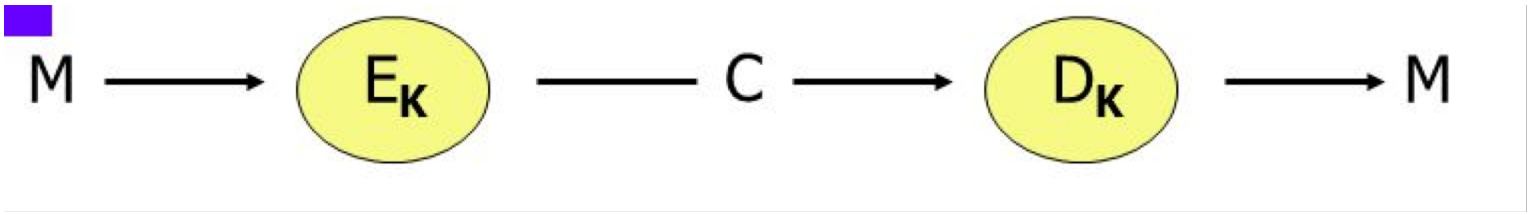
- Issues
  - What is a good symmetric encryption scheme?
  - What is the complexity of encrypting/decrypting?
  - What is the size of the ciphertext, relative to the plaintext?



https://www.slideshare.net/TusharSwami1/cryptography-7296903

Basics

- Efficiency
  - functions EK and DK should have efficient algorithms
- Consistency
  - Decrypting the ciphertext yields the plaintext
  - DK(EK(P)) = P



http://slideplayer.com/slide/10975387/

Cryptanalysis

- A cryptanalyst's chore is to break an encryption
  - Attempts to deduce the original meaning of a ciphertext message
  - Attempts to determine which decrypting algorithm, and key matches the encrypting algorithm t
    - to be able to break other messages encoded in the same way

Cryptanalysis

- An encryption algorithm is called ***breakable*** if it is possible to decrypt the original message
  - given enough time and data,
- However, an algorithm that is theoretically breakable may be impractical to break
  - May take too long
    - Multiple of our lifetimes
- The difficulty of breaking an encryption is called its ***work factor***

Key Exchange

- ***Symmetric*** algorithms use one key, which works for both encryption and decryption
  - Usually, the decryption algorithm is closely related to the encryption one
    - running the encryption in reverse
- Both parties share a secret key
  - they can both encrypt sent information as well as decrypt information from the other

Key Exchange

- As long as the key remains secret, the system also provides authenticity
  - Authenticity is ensured because only the legitimate sender can produce a message that will decrypt properly
    - with the shared key

Key Exchange

- How do two users A and B obtain their shared secret key?
  - And only A and B can use that key for their encrypted communications
  - If A wants to share encrypted communication with another user C, A and C need a different shared secret key.
- Managing keys is the major difficulty in using symmetric encryption

Key Exchange

- If we have *n* users, how many keys do we need?
  - *n* users who want to communicate in pairs need n * (n – 1)/2 keys
    - $O(N^2)$
- The number of keys needed increases at a rate proportional to the square of the number of users
- Symmetric encryption systems require a means of key distribution
- How do we solve this?

Key Exchange

- ***Asymmetric or public key*** systems typically have precisely matched pairs of keys.
- The keys are produced together
  - One may be derived mathematically from the other
  - Process computes both keys as a set
- Asymmetric systems good for key management
  - public key may be emailed or post it in a public directory
- Only the corresponding private key can decrypt what has been encrypted with the public key

CIPHERS

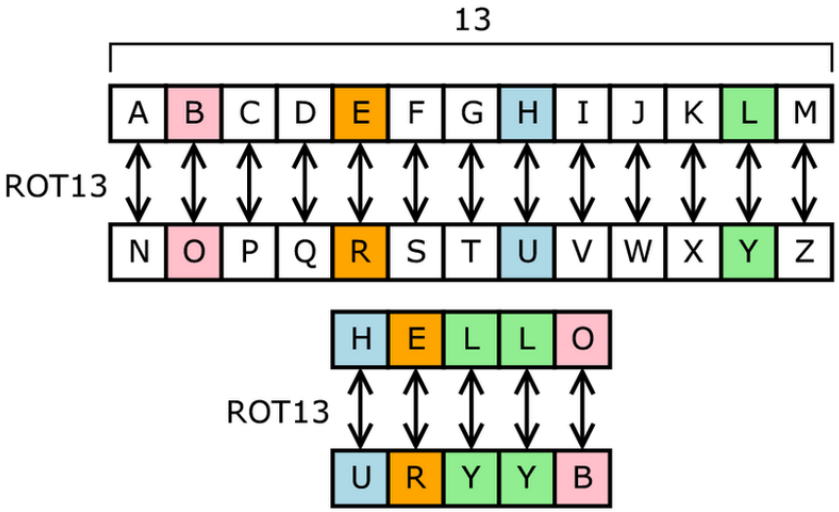
Substitution Ciphers

- Each letter is uniquely replaced by another.
- There are 26! possible substitution ciphers.
- There are more than 4.03 x 1026 such ciphers.

Public domain image from http://en.wikipedia.org/wiki/File:ROT13.png

Substitution Ciphers

- One popular substitution “cipher” for some Internet posts is ROT13.



Public domain image from http://en.wikipedia.org/wiki/File:ROT13.png

Frequency Analysis

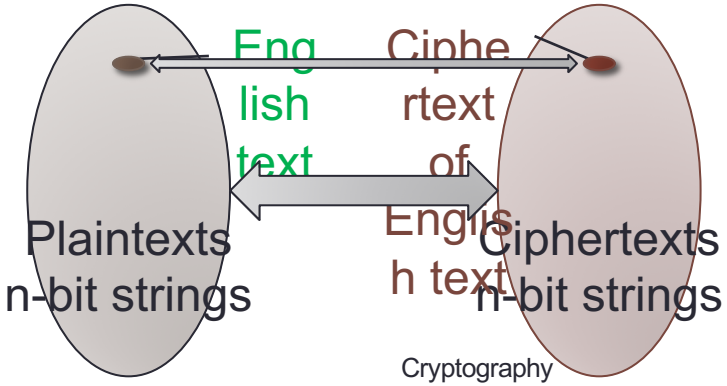
- Letters in a natural language, like English, are not uniformly distributed.
- Knowledge of letter frequencies, including pairs and triples can be used in cryptologic attacks against substitution ciphers.

a:	8.05%	b:	1.67%	c:	2.23%	d:	5.10%
e:	12.22%	f:	2.14%	g:	2.30%	h:	6.62%
i:	6.28%	j:	0.19%	k:	0.95%	l:	4.08%
m:	2.33%	n:	6.95%	o:	7.63%	p:	1.66%
q:	0.06%	r:	5.29%	s:	6.02%	t:	9.67%
u:	2.92%	v:	0.82%	w:	2.60%	x:	0.11%
y:	2.04%	z:	0.06%				

Letter frequencies in the book *The Adventures of Tom Sawyer*, by Twain.

Encrypting English Text

- English text typically represented with 8-bit ASCII encoding
- A message with t characters corresponds to an n-bit array, with n = 8t
- Redundancy due to repeated words and patterns
  - E.g., “th”, “ing”
- English plaintexts are a very small subset of all n-bit arrays



Entropy of Natural Language

- How much information can an alphabet with 8 characters carry?
  - 3 bits of information
  - $2^3 = 8$
- For the English language, each character can convey log (26) = 4.7 bits of information

Entropy of Natural Language

- However, meaningful English text is only ~1.25 bits per char
- Therefore, plaintext redundancy = 4.7 – 1.25 = 3.45
- For example, if a word has 8 characters, what is the effective dictionary size?
  - How many words on average will we have?
  - $2^{8*1.25} = 2^{10} = 1024$

Entropy of English Language

- How do you statistically calculate the entropy of the next letter when the previous N - 1 letters are known?
  - In a word
- As N increases, the entropy approaches H, or the entropy of English

	F_0	F_1	F_2	F_3
26 letter	4.70	4.14	3.56	3.3

http://people.seas.harvard.edu/~jones/cscie129/papers/stanford_info_paper/entropy_of_english_9.htm

Entropy of English Language

- Do we include the space?
  - The 27-letter sequences include the space as a letter
  - Onecanalmostalwaysfillinthespacesfromasequenceofwo rdswithnospaces
  - Therefore, spaces are basically redundant and will cause lower calculated entropies when taken into account

Entropy of English Language

- Do we include the space?
  - Entropy of each letter in a string in English:

	F_0	F_1	F_2	F_3
26 letter	4.70	4.14	3.56	3.3
27 letter	4.76	4.03	3.32	3.1

    - Spaces are basically redundant and will cause lower calculated entropies when taken into account
    - Only in the case where no statistics are taken into account, F0, is the entropy higher when the space is added
      - This simply adds another possible symbol, which means more uncertainty

http://people.seas.harvard.edu/~jones/cscie129/papers/stanford_info_paper/entropy_of_english_9.htm

Substitution Ciphers

- Each letter is uniquely replaced by another.
- There are 26! possible substitution ciphers.
- There are more than 4.03 x 1026 such ciphers.
  - => Frequency attacks may challenge security of substitution ciphers

Public domain image from http://en.wikipedia.org/wiki/File:ROT13.png

Substitution Boxes

- Substitution can also be done on binary numbers.
- Such substitutions are usually described by substitution boxes, or S-boxes.

	00	01	10	11
00	0011	0100	1111	0001
01	1010	0110	0101	1011
10	1110	1101	0100	0010
11	0111	0000	1001	1100

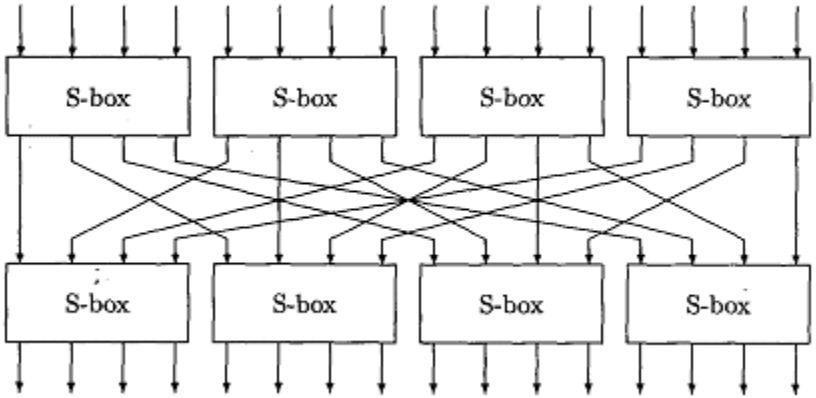
(a)

	0	1	2	3
0	3	8	15	1
1	10	6	5	11
2	14	13	4	2
3	7	0	9	12

(b)

Figure 8.3: A 4-bit S-box (a) An S-box in binary. (b) The same S-box in decimal. This particular S-box is used in the Serpent cryptosystem, which

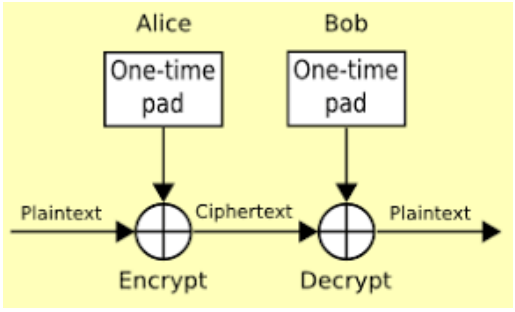
Substitution Boxes



https://www2.dmst.aueb.gr/dds/secimp/crypto/sp.htm

One-Time Pads

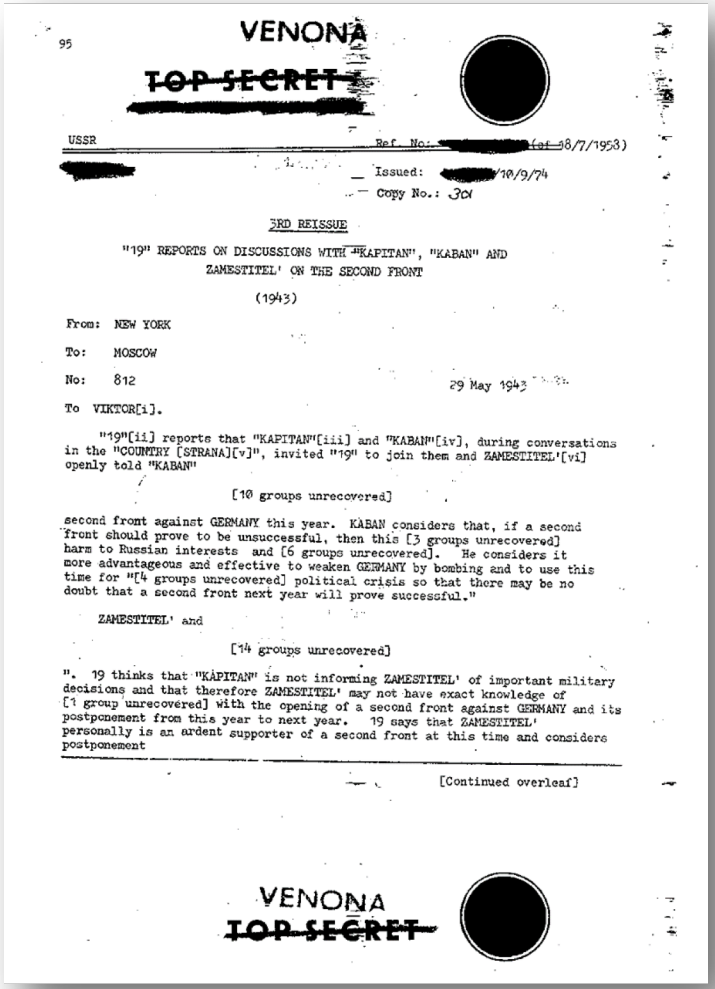
- There is one type of substitution cipher that is absolutely unbreakable.
  - The one-time pad was invented in 1917 by Joseph Mauborgne and Gilbert Vernam
  - We use a block of shift keys, (k_1, k_2, . . . , k_n), to encrypt a plaintext, M, of length n
    - with each shift key being chosen uniformly at random
- Since each shift is random, every ciphertext is equally likely for any plaintext.



https://programmingcode4life.blogspot.com/2015/10/one-time-pad-cipher.html

Weaknesses of the One-Time Pad

- In spite of their perfect security, one-time pads have some weaknesses
- The key has to be as long as the plaintext
- Keys can never be reused
  - Repeated use of one-time pads allowed the U.S. to break some of the communications of Soviet spies during the Cold War.



95

VENONA

~~TOP SECRET~~

USSR

Ref. No: [redacted] (of 8/7/1953)

[redacted] Issued: [redacted]/10/9/74

Copy No.: 304

3RD REISSUE

"19" REPORTS ON DISCUSSIONS WITH "KAPITAN", "KABAN" AND ZAMESTITEL' ON THE SECOND FRONT

(1943)

From: NEW YORK

To: MOSCOW

No: 812

29 May 1943

To VIKTOR[i].

"19"[ii] reports that "KAPITAN"[iii] and "KABAN"[iv], during conversations in the "COUNTRY [STRANA][v]", invited "19" to join them and ZAMESTITEL'[vi] openly told "KABAN"

[10 groups unrecovered]

second front against GERMANY this year. KABAN considers that, if a second front should prove to be unsuccessful, then this [3 groups unrecovered] harm to Russian interests and [6 groups unrecovered]. He considers it more advantageous and effective to weaken GERMANY by bombing and to use this time for "[4 groups unrecovered] political crisis so that there may be no doubt that a second front next year will prove successful."

ZAMESTITEL' and

[14 groups unrecovered]

". 19 thinks that "KAPITAN" is not informing ZAMESTITEL' of important military decisions and that therefore ZAMESTITEL' may not have exact knowledge of [1 group unrecovered] with the opening of a second front against GERMANY and its postponement from this year to next year. 19 says that ZAMESTITEL' personally is an ardent supporter of a second front at this time and considers postponement

[Continued overleaf]

VENONA

~~TOP SECRET~~

Stream and Block Ciphers

- How is data processed to be concealed?
- The data stream may come in bursts
  - Example: streaming a video from a satellite
  - What is the best way to encrypt it?

Stream Encryption

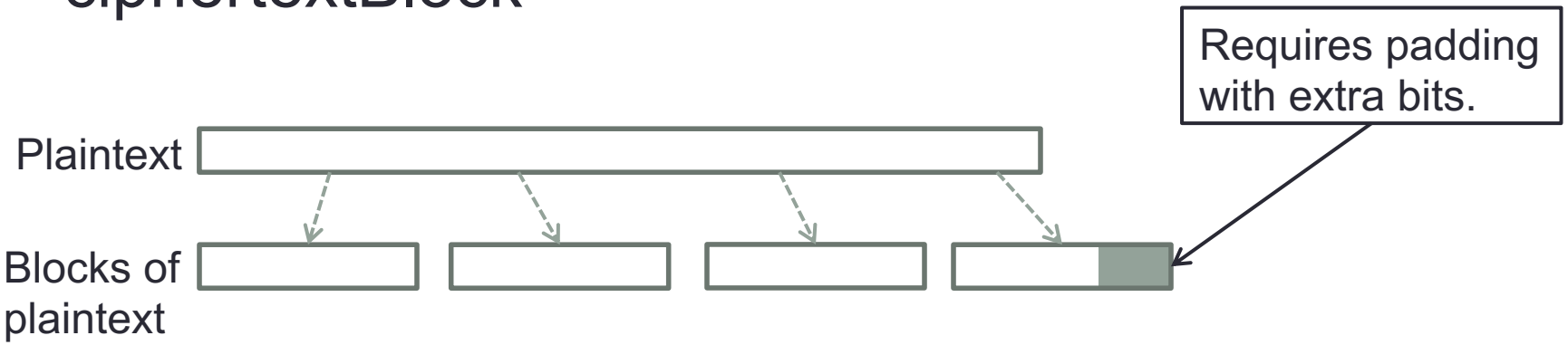
- Each bit or byte of the data stream is encrypted separately
  - May be applicable for data stream processing
- Advantage: it can be applied immediately to whatever data items are ready to transmit
- Disadvantage: most encryption algorithms involve complex transformations
  - To do these transformations on one or a few bits at a time is expensive

Block Ciphers

- Encrypts a group of plaintext symbols as a single block.
  - Unlike a stream cipher
- A block cipher algorithm performs its work on a quantity of plaintext data all at once
- Perhaps the most-used technique for encryption

Block Ciphers

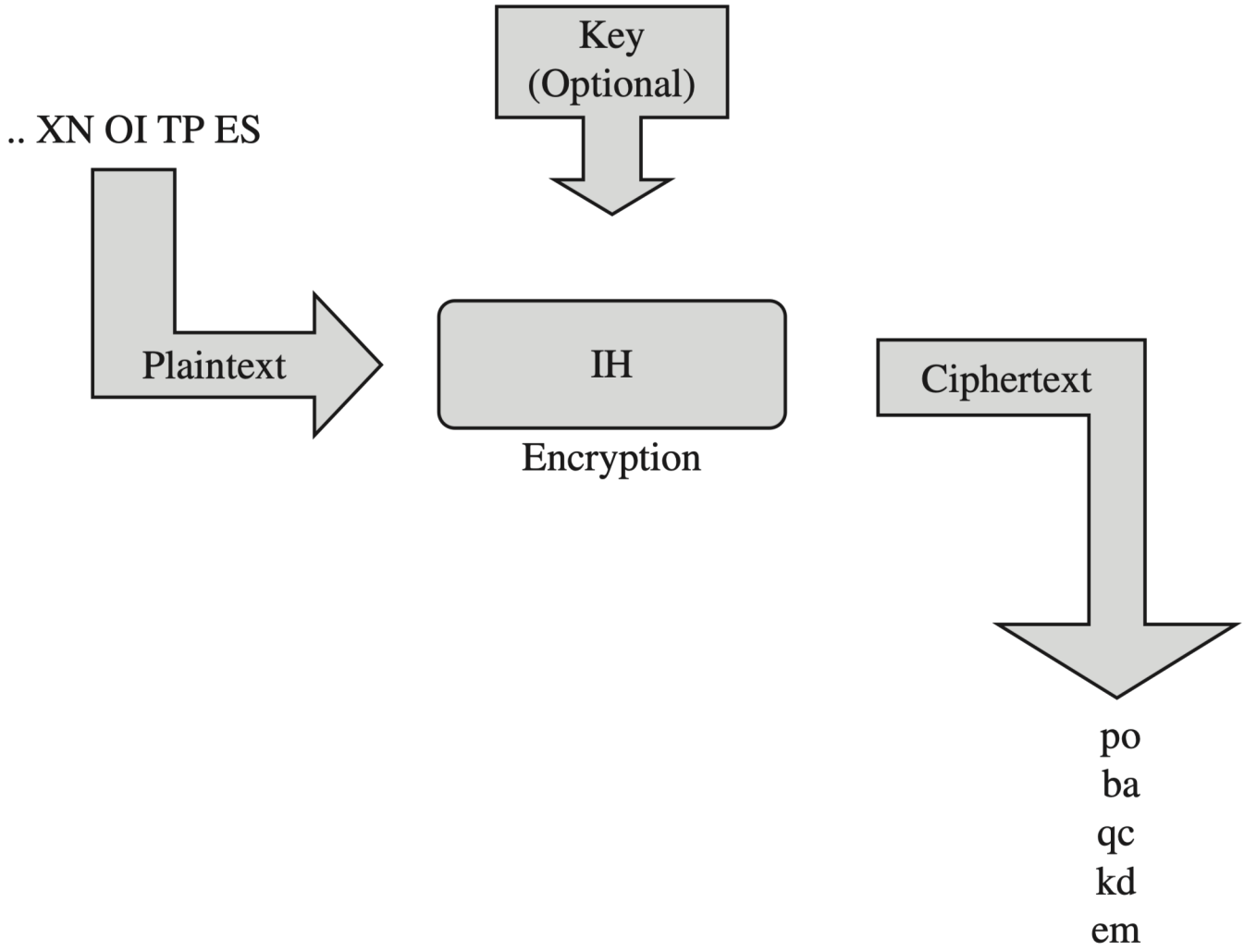
- Plaintext is partitioned into a sequence of blocks,
  - P = P[0], …, P[m–1]
- Each block is of fixed length b (e.g., 128 bits)
- Each block is encrypted (more or less) separately, BlockCipher(key, plaintextBlock) → ciphertextBlock



Padding

- Plaintext length must be a multiple of the block size
  - Otherwise we must pad last partial block to a full block
  - Must be able to tell when data ends, padding begins
- Example for block-size = 128 (16 bytes)
  - Plaintext: “Roberto” (7 bytes)
  - Padded plaintext: “Roberto999999999” (16 bytes)
    - Problem: cannot tell if plaintext was “Roberto” or “Roberto9”
  - Better: padded plaintext “Roberto099999999”

Block Ciphers

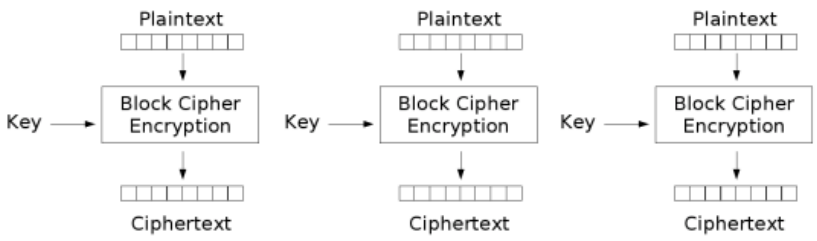


Stream vs. Block

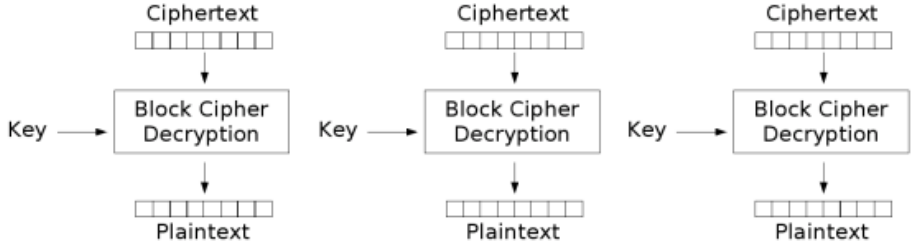
	Stream	Block
Advantages	• Speed of transformation • Low error propagation	• High diffusion • Immunity to insertion of symbol
Disadvantages	• Low diffusion • Susceptibility to malicious insertions and modifications	• Slowness of encryption • Padding • Error propagation

Block Cipher Modes

- A block cipher mode describes the way a block cipher encrypts and decrypts a sequence of message blocks.
- Electronic Code Book (ECB) Mode (is the simplest):



Electronic Codebook (ECB) mode encryption

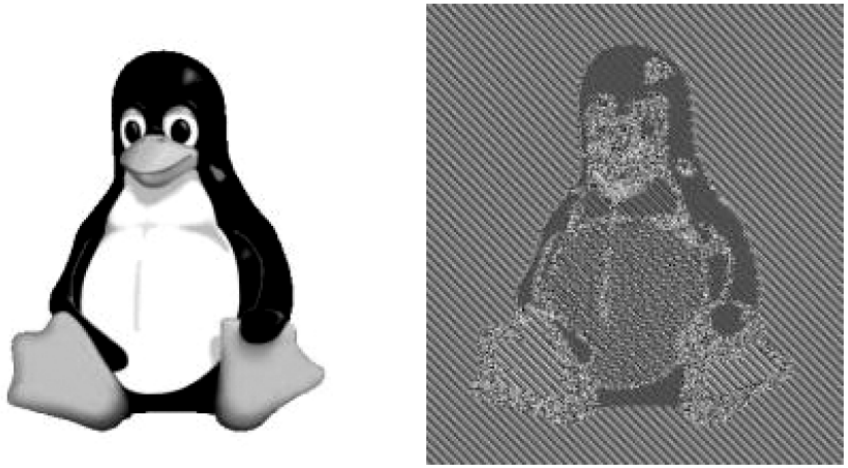


Electronic Codebook (ECB) mode decryption

Strengths and Weaknesses of ECB

- Strengths:
  - Is very simple
  - Allows for parallel encryptions of the blocks of a plaintext
  - Can tolerate the loss or damage of a block
- Weakness:
  - Documents and images are not suitable for ECB encryption since patters in the plaintext are repeated in the ciphertext:

Strengths and Weaknesses of ECB



(a) (b)

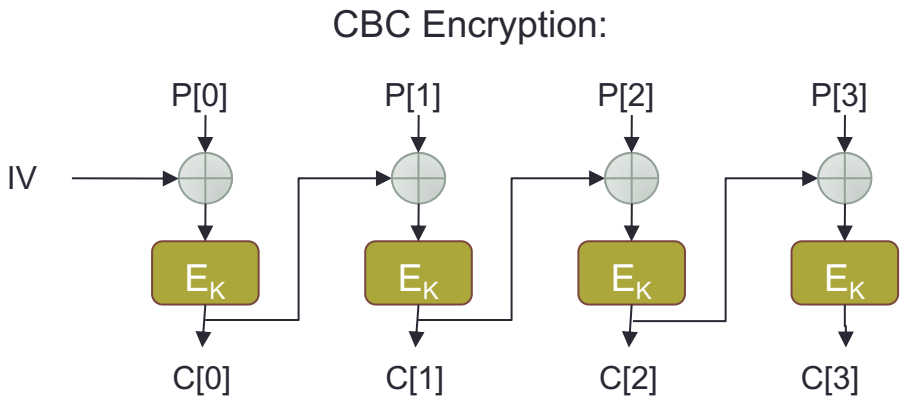
Figure 8.6: How ECB mode can leave identifiable patterns in a sequence of blocks: (a) An image of Tux the penguin, the Linux mascot. (b) An encryption of the Tux image using ECB mode. (The image in (a) is by Larry Ewing, lewing@isc.tamu.edu, using The Gimp; the image in (b) is by Dr. Juzam. Both are used with permission via attribution.)

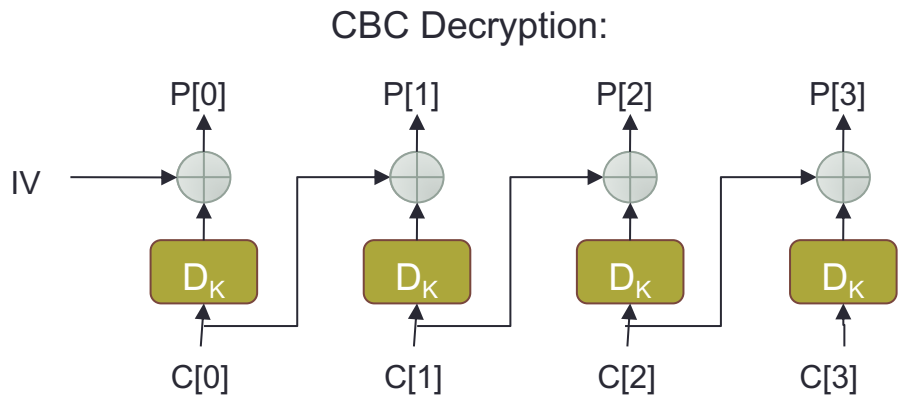
Cipher Block Chaining (CBC) Mode

- How do we overcome this weakness?
  - Use randomized encryption: encrypting the same thing many times yield a different ciphertext every time

Cipher Block Chaining (CBC) Mode

- In Cipher Block Chaining (CBC) Mode
  - The previous ciphertext block is combined with the current plaintext block $C[i] = EK (C[i –1] \oplus P[i])$
  - C[–1] = IV, a random block separately transmitted encrypted (known as the initialization vector)
  - Decryption: $P[i] = C[i –1] \oplus DK (C[i])$





Cipher Block Chaining (CBC) Mode

- How do we overcome this weakness?
  - Use randomized encryption: encrypting the same thing many times yield a different ciphertext every time
  - CBC is a randomized encryption mode
    - if we choose a different iv – initial value

Strengths and Weaknesses of CBC

- Strengths:
  - Doesn’t show patterns in the plaintext
  - Is the most common mode
  - Is fast and relatively simple
- Weaknesses:
  - CBC requires the reliable transmission of all the blocks sequentially
  - CBC is not suitable for applications that allow packet losses (e.g., music and video streaming)

BLOCK CIPHERS IN PRACTICE

Data Encryption Standard (DES)

- Symmetric encryption algorithm
- Developed by IBM and adopted by NIST in 1977
- Encrypts 64-bit blocks using 56-bit keys
- Small key space makes exhaustive search attack feasible since late 90s
  - Not secure – should not be used!

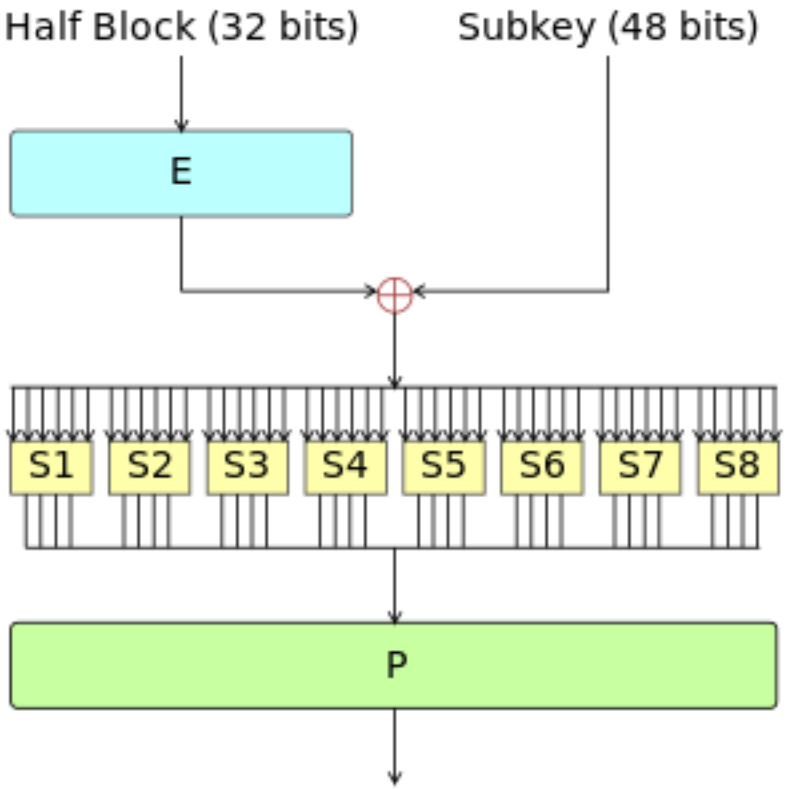
Data Encryption Standard (DES)

- DES is a combination of two fundamental building blocks of encryption:
  - substitution and transposition
- These techniques are repeated one on top of the other, for a total of 16 cycles

Data Encryption Standard (DES)

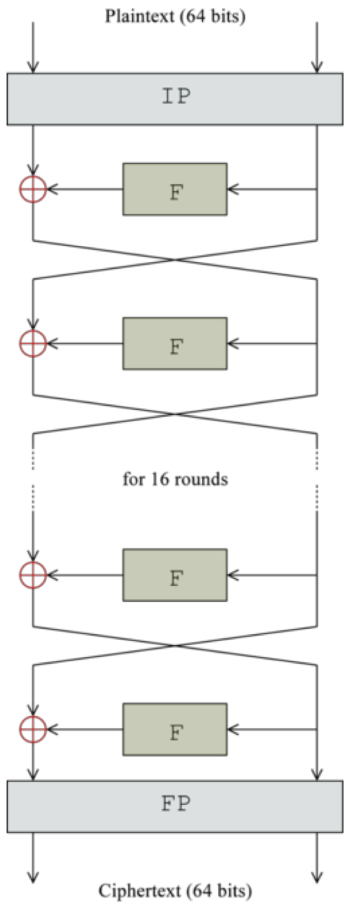
- Each round includes:
  - Substitution:
    - replacing blocks of bits
  - Permutation:
    - Shuffling the bits
  - Transformation:
    - Mingling bits from the key

DES Single Round (Feistel Function)



https://en.wikipedia.org/wiki/Data_Encryption_Standard

DES Algorithm



https://en.wikipedia.org/wiki/Data_Encryption_Standard

DES Algorithm

- DES 56-bit key length is not long enough
  - Not secure
    - Computing power increased rapidly in last few decades
  - How can we make it secure?
    - Explore longer-key version of DES
    - **Problem**: the DES algorithm design is **fixed to a 56-bit key**

Double DES

- Perform two encryptions using two distinct keys:
  - $E(k_2 , E(k_1 ,m))$
- Shown not to be secure [Merkle & Hellman 81]

Triple DES (3DES)

- Nested application of DES with three different keys K_A, K_B, and K_C
- Effective key length is 168 bits
  - making exhaustive search attacks unfeasible
- Ciphertext $C = EK_C(DK_B(EK_A(P)))$;
- Plaintext $P = DK_A(EK_B(DK_C(C)))$
- Equivalent to DES when $K_A=K_B=K_C$ (backward compatible)

Encryption Standards

Form	Operation	Properties	Strength
DES	Encrypt with one key	56-bit key	Inadequate for high-security applications by today's computing capabilities
Double DES	Encrypt with first key; then encrypt result with second key	Two 56-bit keys	Only doubles strength of 56-bit key version
Two-key triple DES	Encrypt with first key, then encrypt (or decrypt) result with second key, then encrypt result with first key (E-D-E)	Two 56-bit keys	Gives strength equivalent to about 80-bit key (about 16 million times as strong as 56-bit version)
Three-key triple DES	Encrypt with first key, then encrypt or decrypt result with second key, then encrypt result with third key (E-E-E)	Three 56-bit keys	Gives strength equivalent to about 112-bit key about 72 quintillion ($72*10^{15}$) times as strong as 56-bit version

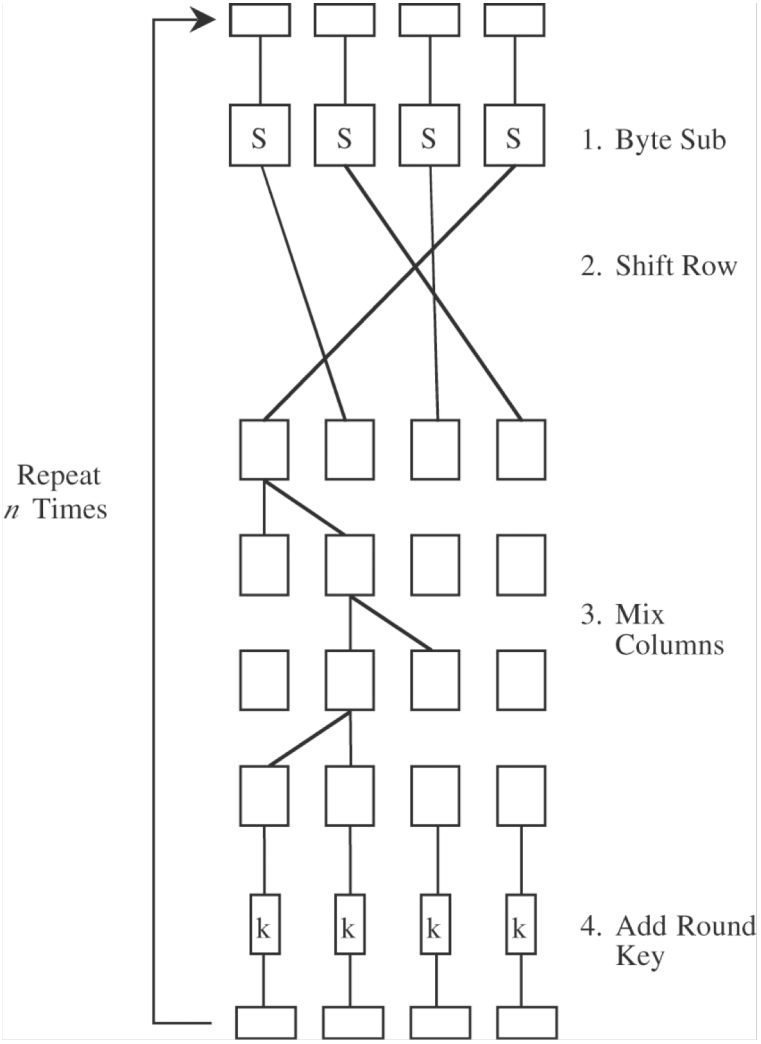
Advanced Encryption Standard (AES)

- Symmetric block cipher
- Developed in 1999 by independent Dutch cryptographers
- Selected by NIST in 2001 through open international competition and public discussion

Advanced Encryption Standard (AES)

- 128-bit blocks and several possible key lengths: 128, 192 and 256 bits
- Exhaustive search attack not currently possible
- AES-256 is the symmetric encryption algorithm of choice
  - Still in common use

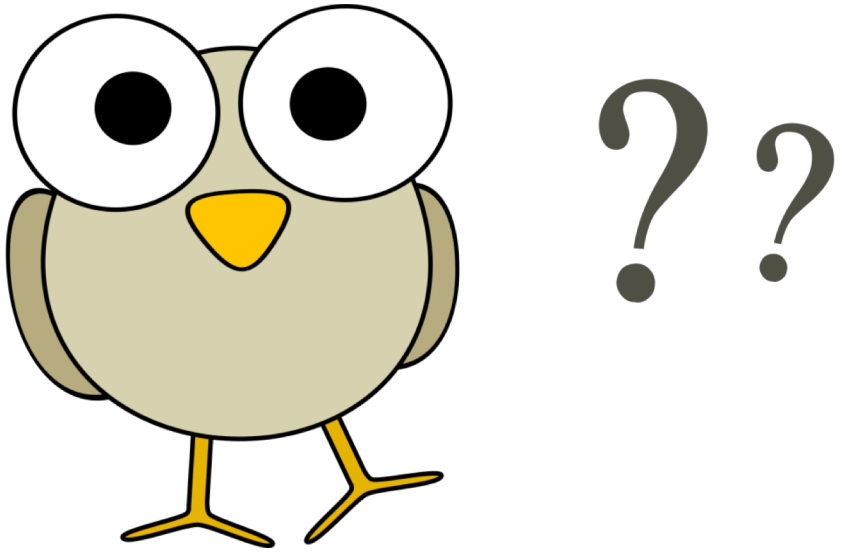
AES: Advanced Encryption System



DES vs. AES

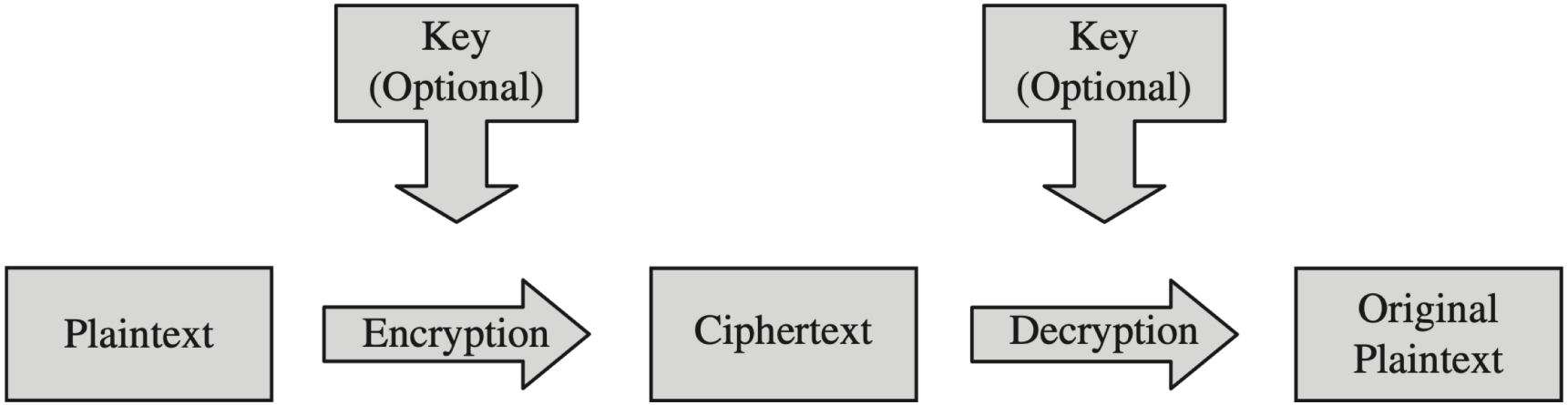
	DES	AES
Date designed	1976	1999
Block size	64 bits	128 bits
Key length	56 bits (effective length); up to 112 bits with multiple keys	128, 192, 256 (and possibly more) bits
Operations	16 rounds	10, 12, 14 (depending on key length); can be increased
Encryption primitives	Substitution, permutation	Substitution, shift, bit mixing
Cryptographic primitives	Confusion, diffusion	Confusion, diffusion
Design	Open	Open
Design rationale	Closed	Open
Selection process	Secret	Secret, but open public comments and criticisms invited
Source	IBM, enhanced by NSA	Independent Dutch cryptographers

- Questions?

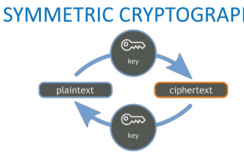


PUBLIC KEY (ASYMMETRIC) ENCRYPTION

Encryption/Decryption Process



Reminder - Symmetric Cryptosystem



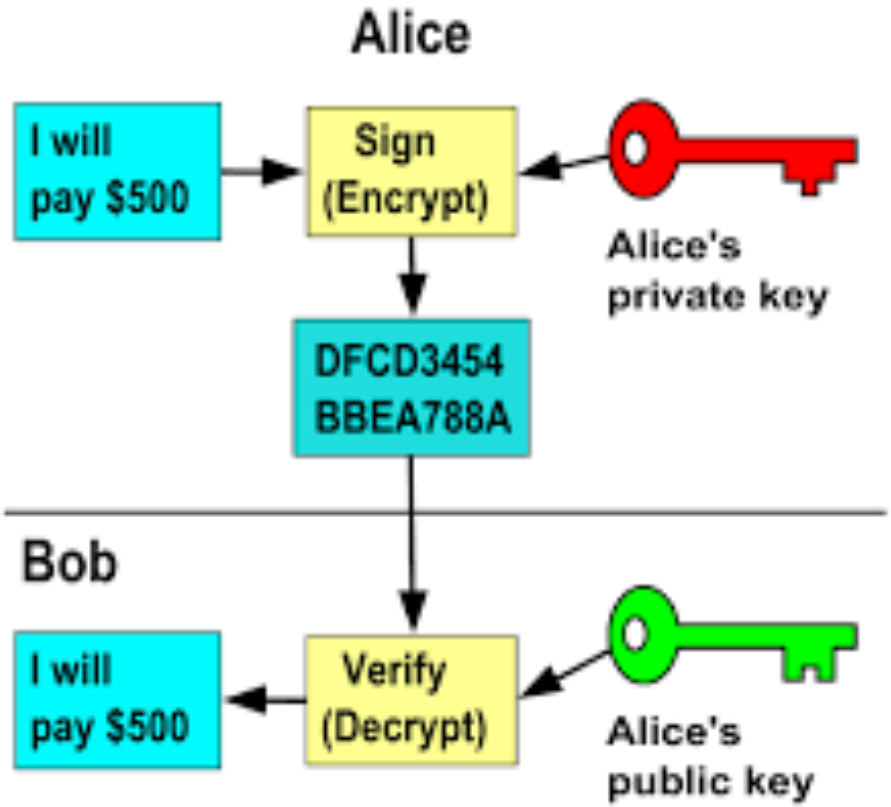
- Scenario
  - Alice wants to send a message (plaintext P) to Bob.
  - The communication channel is insecure and can be eavesdropped
  - If Alice and Bob have previously agreed on a symmetric encryption scheme and a secret key K, the message can be sent encrypted (ciphertext C)

https://www.slideshare.net/TusharSwami1/cryptography-7296903

Public Key (Asymmetric) Cryptography

- Instead of two users sharing one secret key, each user has two keys: one public and one private
- Messages encrypted using the user's public key can only be decrypted using the user's private key, and vice versa
  - P = D(Kpriv,E(Kpub,P)) or
  - P = D(Kpub,E(Kpriv,P))

Public Key Encryption



https://commons.wikimedia.org/wiki/File:Public_key_signing.svg

Public Key Encryption

- A cryptographic system that uses pairs of keys
  - public keys which may be disseminated widely
    - Any person can encrypt the message
  - private keys which are known only to the owner
    - Message can only be decrypted with this key
- Analogous to a self-closing door
  - Need key to open it
  - Closing it shuts it automatically

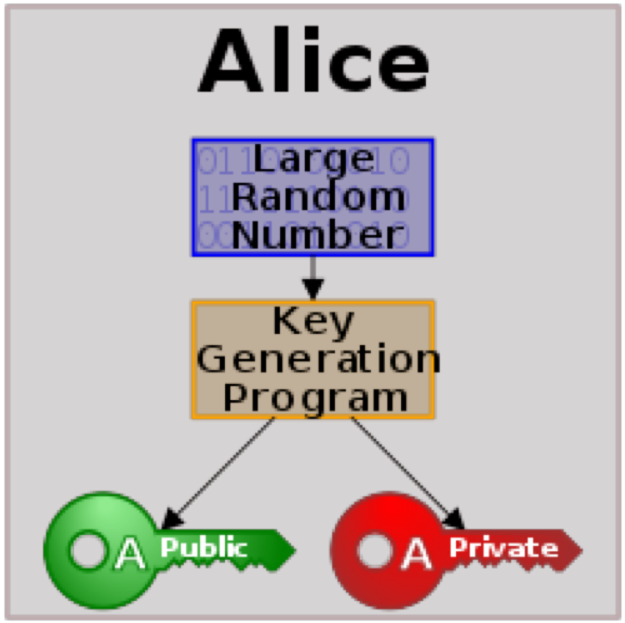
Public Key Encryption

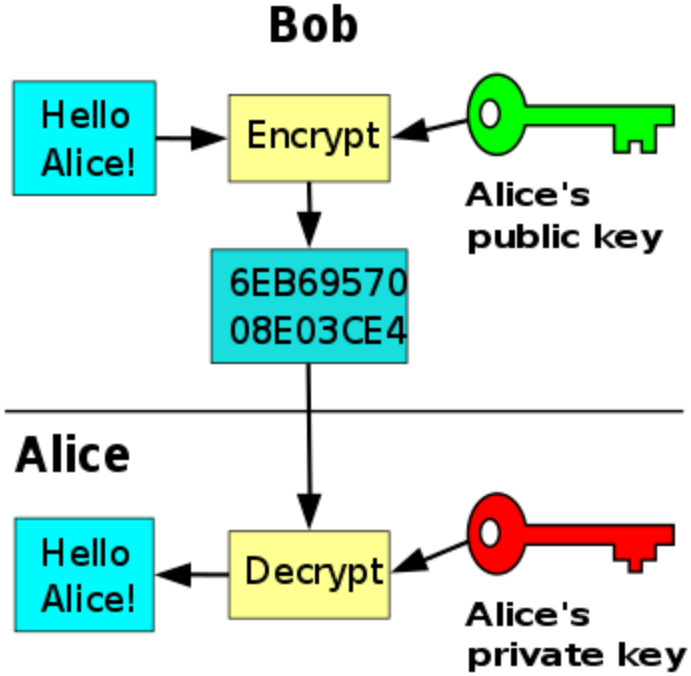
- Accomplishes two functions: authentication and encryption
- Authentication: the public key verifies that a holder of the paired private key sent message
  - Decryption will fail otherwise
- Encryption: only the paired private key holder can decrypt the encrypted message

Public Key Encryption

- Why does it work?
- It is not feasible to compute the private key
  - From knowledge of its paired public key
- Therefore, only the private key is kept private
  - The public key can be openly distributed without compromising security
- Public key cryptography relies on math problems that have no efficient solution

Public Key Encryption





https://en.wikipedia.org/wiki/Public-key_cryptography

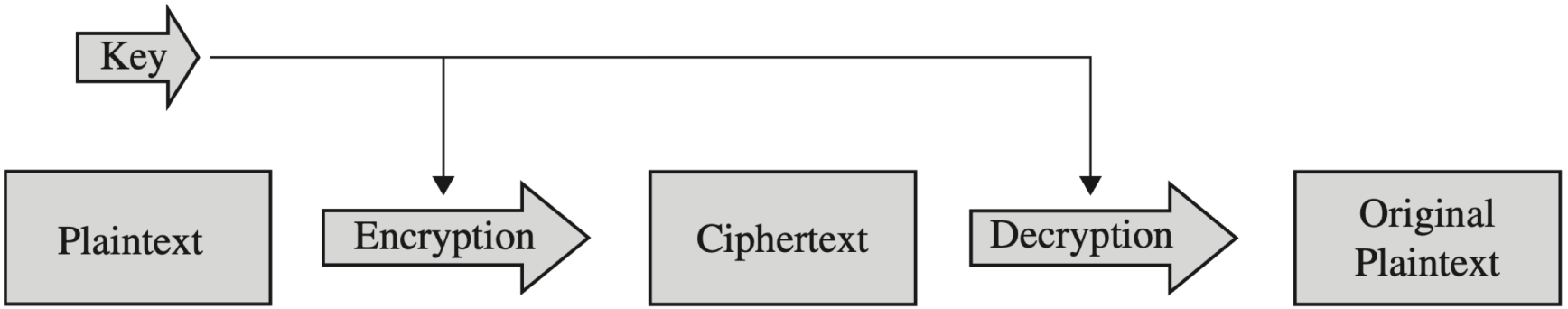
Public Key Encryption

- Often used to secure communication
  - Over the internet, open networks
  - Typically used for key exchange

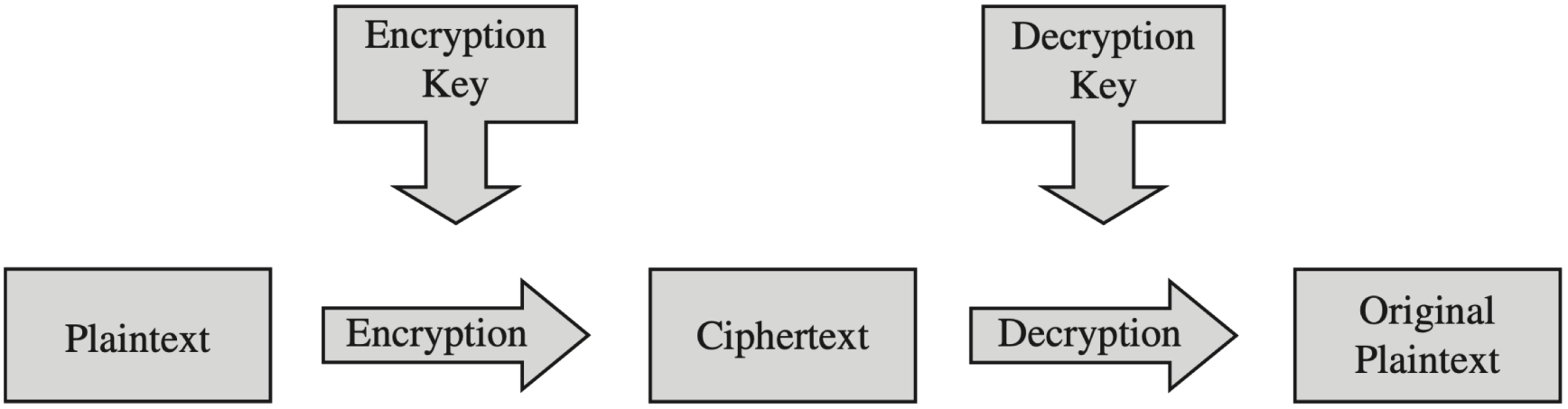
Secret Key vs. Public Key Encryption

	Secret Key (Symmetric)	Public Key (Asymmetric)
Number of keys	1	2
Key size (bits)	56–112 (DES), 128–256 (AES)	Unlimited; typically no less than 256; 1000 to 2000 currently considered desirable for most uses
Protection of key	Must be kept secret	One key must be kept secret; the other can be freely exposed
Best uses	Cryptographic workhorse. Secrecy and integrity of data, from single characters to blocks of data, messages and files	Key exchange, authentication, signing
Key distribution	Must be out-of-band	Public key can be used to distribute other keys
Speed	Fast	Slow, typically by a factor of up to 10,000 times slower than symmetric algorithms

Symmetric vs. Asymmetric



(a) Symmetric Cryptosystem



(b) Asymmetric Cryptosystem

Symmetric vs. Asymmetric

- The critical difference between symmetric and asymmetric is that symmetric uses a single key for both encryption and decryption
  - whereas asymmetric uses complementary keys

Rivest-Shamir-Adelman Public Key Encryption (RSA)

- A public key system
- Based on an underlying hard problem
  - Makes is hard to break the system
    - Get the private key
- RSA uses two keys, *d* and e, for decryption and encryption

Rivest-Shamir-Adelman Public Key Encryption (RSA)

- The two keys can be used interchangeably:
- C = RSA(P,*e*) - *e* can be either Kpriv or Kpub

- Then
- P = RSA(RSA(P,*e*),*d*) where *d* is the other key

Rivest-Shamir-Adelman Public Key Encryption (RSA)

- The keys can be 256 to 1000s of bits
- The relationship between the keys relies on the fact that a large number is the product of two large prime numbers.

RSA Runtime

- Keys are long: 256 is the minimum
- Encryption is done by exponentiation
  - In contrast to substitution and transposition used in symmetric algorithm
    - Significantly slower to run on a computer
- RSA runtime significantly longer than DES or AES

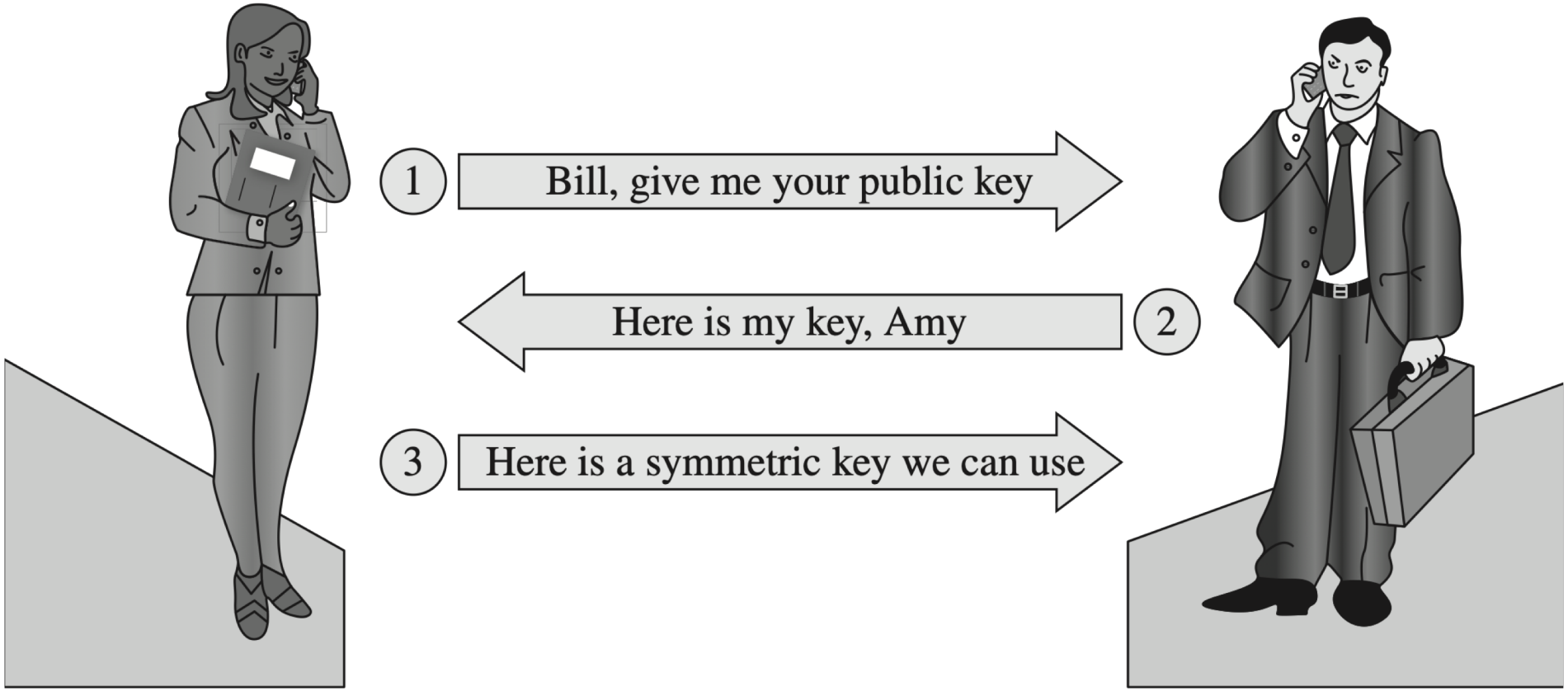
Secret Key vs. Public Key Encryption

- Symmetric and asymmetric algorithms have complementary strengths and weaknesses
- Used for different purposes
  - in concert with each other

Secret Key vs. Public Key Encryption

	Secret Key (Symmetric)	Public Key (Asymmetric)
Number of keys	1	2
Key size (bits)	56–112 (DES), 128–256 (AES)	Unlimited; typically no less than 256; 1000 to 2000 currently considered desirable for most uses
Protection of key	Must be kept secret	One key must be kept secret; the other can be freely exposed
Best uses	Cryptographic workhorse. Secrecy and integrity of data, from single characters to blocks of data, messages and files	Key exchange, authentication, signing
Key distribution	Must be out-of-band	Public key can be used to distribute other keys
Speed	Fast	Slow, typically by a factor of up to 10,000 times slower than symmetric algorithms

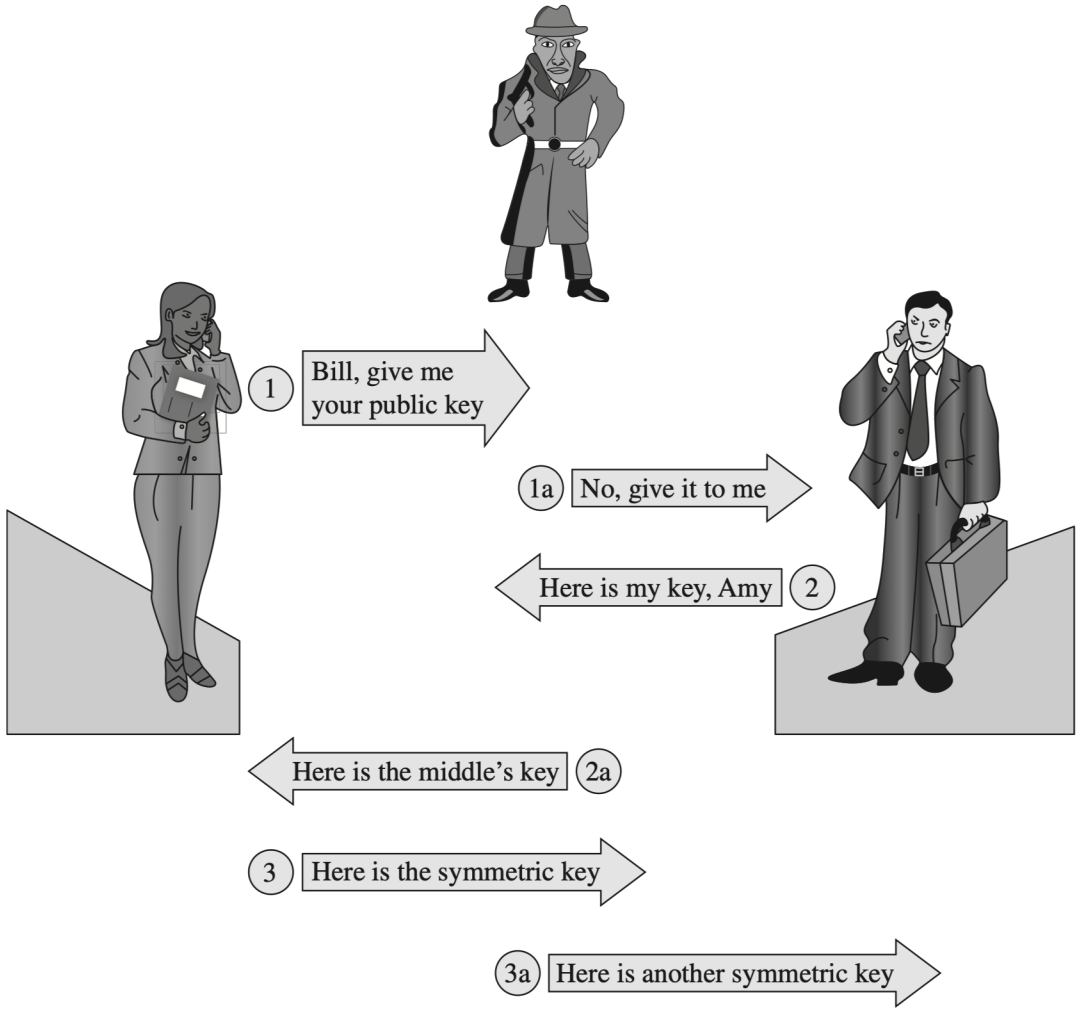
Public Key to Exchange Secret Keys



Key Exchange Man in the Middle

- What if we have a man in the middle attack?
  - Revised protocol is needed

Key Exchange Man in the Middle



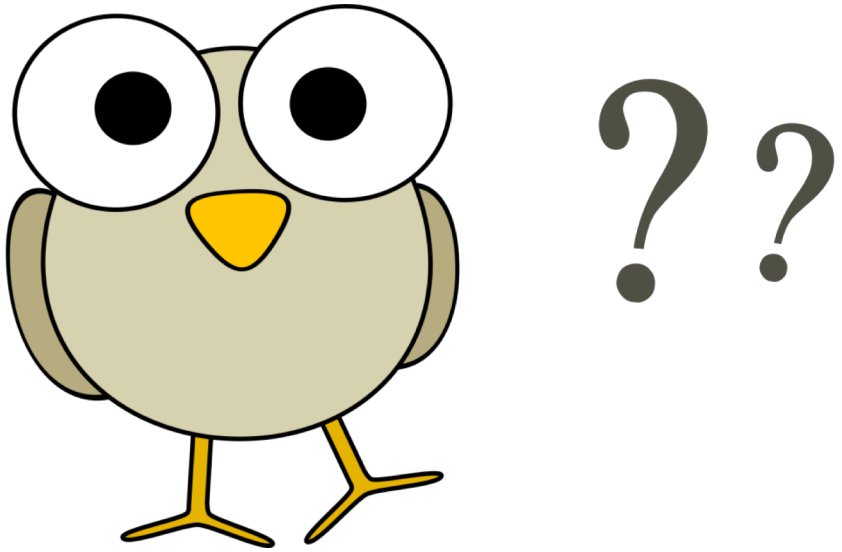
Revised Key Exchange Protocol

- Secure key exchange is complicated
- Many alternative secure protocols exist
  - Protocols need to be proven secure
  - New attacks introduced continuously

How to Break Cryptography?

- How to break cryptography

- Questions?



Error Detecting Codes

- Communications are prone to transmission errors
- Need to have a way to verify intact transmission
  - For sensitive data
- Different error detection mechanisms exist
- Aims to indicate that a message has changed
  - Different techniques work for different errors

Error Detecting vs. Correcting Codes

- ***Error detecting codes*** detect when an error has occurred
- ***Error correcting codes*** can actually correct errors without requiring a copy of the original data
  - without requiring a copy of the original data

Error Detecting vs. Correcting Codes

- The error code is computed and stored safely on the presumed intact, original data;
- Error code can be recomputed later
  - check whether the received result matches the expected value.
  - If the values do not match, a change has occurred;

Error Detecting Codes

- Demonstrates that a block of data has been modified
- Simple error detecting codes:
  - Parity checks
    - Parity bit added to a string of binary code to ensure that the total number of 1-bits in the string is even or odd
  - Cyclic redundancy checks – used on hardware devices

Error Detecting Codes

- Cryptographic error detecting codes:
  - One-way hash functions
  - Cryptographic checksums
  - Digital signatures

Parity Check

Original Data	Parity Bit	Modified Data	Modification Detected?
0 0 0 0 0 0 0 0	1	0 0 0 0 0 0 0 1	
0 0 0 0 0 0 0 0	1	1 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0	1	1 0 0 0 0 0 0 1	
0 0 0 0 0 0 0 0	1	0 0 0 0 0 0 1 1	
0 0 0 0 0 0 0 0	1	0 0 0 0 0 1 1 1	
0 0 0 0 0 0 0 0	1	0 0 0 0 1 1 1 1	
0 0 0 0 0 0 0 0	1	0 1 0 1 0 1 0 1	

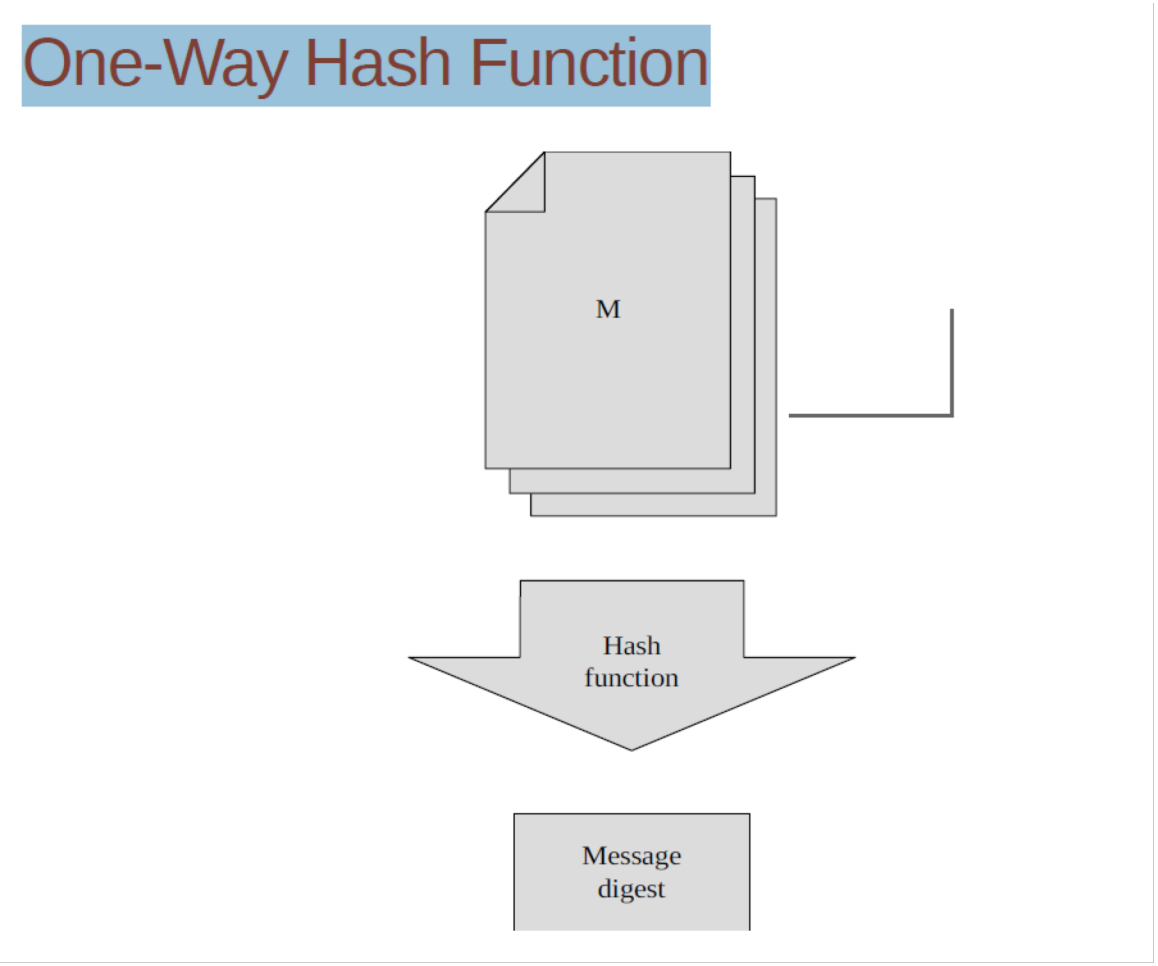
Parity Check

Original Data	Parity Bit	Modified Data	Modification Detected?
0 0 0 0 0 0 0 0	1	0 0 0 0 0 0 0 1	Yes
0 0 0 0 0 0 0 0	1	1 0 0 0 0 0 0 0	Yes
0 0 0 0 0 0 0 0	1	1 0 0 0 0 0 0 1	No
0 0 0 0 0 0 0 0	1	0 0 0 0 0 0 1 1	No
0 0 0 0 0 0 0 0	1	0 0 0 0 0 1 1 1	Yes
0 0 0 0 0 0 0 0	1	0 0 0 0 1 1 1 1	No
0 0 0 0 0 0 0 0	1	0 1 0 1 0 1 0 1	No

Hash Codes

- We want to know that a file has not been tampered with
  - Similar to a seal on an envelope
- How can we use cryptography for that?
  - Use cryptography to create a ***hash*** of the data
    - Other options include checksums or message digests

One-Way Hash Function



Hash Functions

- A hash function h maps a plaintext x to a fixed-length value x = h(P) called hash value or digest of P
  - A collision is a pair of plaintexts P and Q that map to the same hash value, h(P) = h(Q)
  - Collisions are unavoidable
  - For efficiency, the computation of the hash function should take time proportional to the length of the input plaintext

Birthday Attack

- The brute-force birthday attack aims at finding a collision for a hash function h
  - Randomly generate a sequence of plaintexts X1, X2, X3,…
  - For each Xi compute yi = h(Xi) and test whether yi = yj for some j < i
  - Stop as soon as a collision has been found
- If there are m possible hash values, the probability that the i-th plaintext does not collide with any of the previous i -1 plaintexts is 1 - (i - 1)/m

Birthday Attack

- Probability F_k that the attack fails (no collisions) after k plaintexts is

$$F_k = (1 - 1/m)\,(1 - 2/m)\,(1 - 3/m)\,\ldots\,(1 - (k - 1)/m)$$

- Using the standard approximation $1 - x \approx e^{-x}$

$$F_k \approx e^{-(1/m + 2/m + 3/m + \ldots + (k-1)/m)} = e^{-k(k-1)/2m}$$

- The attack succeeds/fails with probability ½ when $F_k = ½$, that is,

$$e^{-k(k-1)/2m} = ½$$

$$k \approx 1.17\, m^{½}$$

  - We conclude that a hash function with b-bit values provides about b/2 bits of security

Birthday Attack

- Probability that 2 people were not born on same day of the year = 30/31
- The probability that the third person was not both on either of those days = 29/31
- For 11 people, we get:
  - P(no collusion) = $\frac{30*29*28*\cdots*21}{31^{10}}$ = 0.13 = 13%
  - P(collusion) = 87%

Birthday Attack

- Notice: Once we know that the first ten do not collide, the probability that the eleventh one will collide is 10/31
  - 10 times larger than the first, which was 1/31
- So each person we add has significantly larger probability of collusion

Cryptographic Checksum

- Cryptographic function that produces checksum.
- A digest function using a cryptographic key
  - Key is presumably known only to the originator and the proper recipient of the data
- The attacker does not have a key with which to recompute the checksum
  - => Checksum important for data modification detection

Cryptographic Checksum

- Major uses of cryptographic checksums:
  - Code-tamper protection
    - Detect changes in system files
  - Message integrity protection in transit
    - Calculate checksum on received data and compare to sent values
- AES can be used as a cryptographic checksum algorithm
  - However, simpler algorithms can be used for less sensitive data
  - SHA (Secure Hash Algorithm) defined by U.S. gov.

Secure Hash Algorithm (SHA)

- Developed by NSA and approved as a federal standard by NIST
- SHA-0 and SHA-1 (1993)
  - 160-bits
  - Considered insecure
  - Still found in legacy applications

Secure Hash Algorithm (SHA)

- SHA-2 family (2002)
  - 256 bits (SHA-256) or 512 bits (SHA-512)
  - Still considered secure despite published attack techniques
- SHA-3 (2015)
  - More complex and secure hash algorithm
  - Faster than SHA-1 and SHA=2

Secure Hash Algorithm (SHA)

- Older hash functions include MD4, MD5
  - Significant vulnerabilities found
  - Should never be used
- Only SHA-2 AND SHA-3 should be used

Digital Signatures

- Tool to demonstrate authenticity
  - similar to a paper signature
- A way by which a person or organization can affix a bit pattern to a file
  - implies confirmation,
  - pertains to that file only
  - cannot be forged

Digital Signatures

- A digital signature often uses asymmetric or public key cryptography
  - public key cryptographic protocols involve several sequences of messages and replies
    - Time consuming if both parties are not immediately available
  - In this situation, a technique that could authenticate a party even if it is inactive would be useful
    - Similar to a paper signature

Paper Signatures

- Traditional properties of a check signature:
  - A check is a *tangible object*
    - authorizing a financial transaction.
  - The check signature confirms authenticity
    - (presumably) only legitimate signer can produce that signature.
  - In the case of an alleged forgery, a third party can be called in to judge authenticity.
  - Once a check is cashed, it is canceled
    - it cannot be reused.
  - The paper check is not alterable.
    - most forms of alteration are easily detected.

Digital Signatures

- Properties required of digital signatures:
  - It must be unforgeable
    - If person S signs message M with signature Sig(S,M), no one else can produce the pair [M,Sig(S,M)].
  - It must be authentic
    - If a person R receives the pair [M, Sig(S,M)] purportedly from S, R can check that the signature is really from S.
    - Only S could have created this signature
      - the signature is firmly attached to M.

Digital Signatures

- Desired Properties:
  - It is not alterable
    - After being transmitted, M cannot be changed
      - By either S, R, or an interceptor.
  - It is not reusable
    - A previous message presented again will be instantly detected by R

Digital Signatures

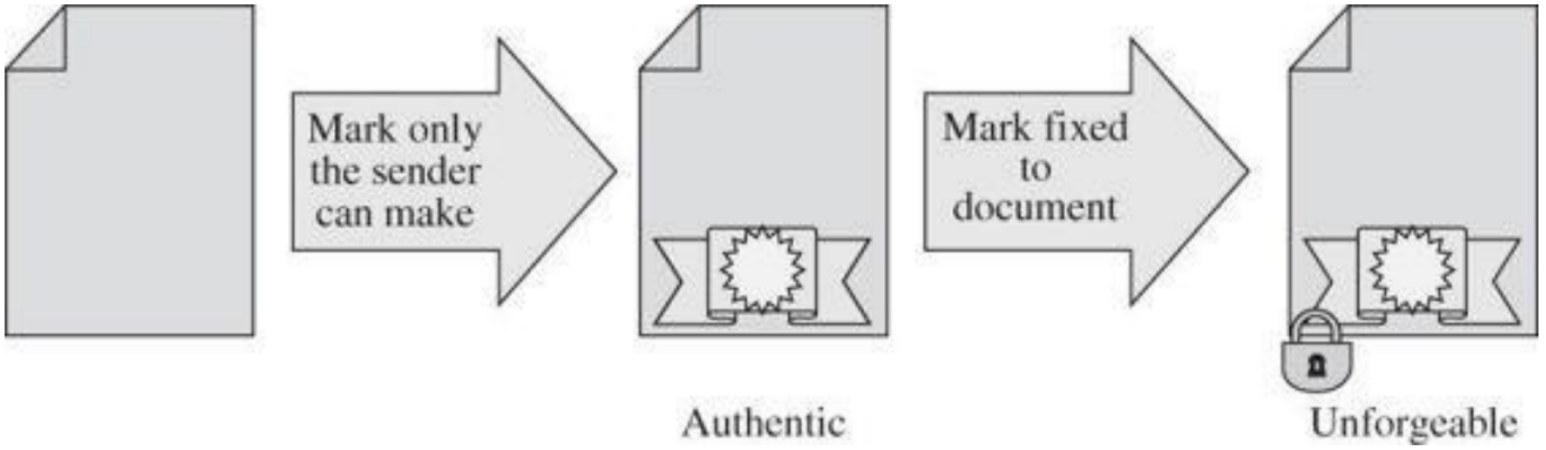


FIGURE 2-26 Digital Signature Requirements

Public Key for Signatures

- Let's assume that:
  - Public key encryption for user U is E(M,KU)
  - Private key transformation for U is written as D(M,KU)
- How do we prove authenticity?

Public Key for Signatures

- Authenticity:
  - Sender S sends D(M, KS) to R
    - Using his private key
  - R decodes the message with the public key transformation
    - Computing E(D(M, KS), KS) = M
  - Since only S can create that message the message must genuinely have come from S
    - Only S has the private key

Public Key for Signatures

- Unforgeability (integrity):
  - R will save D(M, KS)
  - R can show at a later date M and D(M, KS)
    - If S tries to change message M, claim forgery
  - Anyone can verify M since D(M, KS) is transformed to M with the public key transformation of S
    - E(D(M, KS), KS) = M
  - => public key satisfies unforgeability

Trust

- How do we know that a webpage indeed belongs to the listed company?
  - Web pages can be replaced and faked
    - No warning to the user
- We can establish trust based on a common and trusted entity
  - The ***certificate authority***

Certificates: Trustable Identities and Public Keys

- A certificate is a public key and an identity bound together and signed by a ***certificate authority***.
- A ***certificate authority*** is an authority that users trust
  - accurately verifies identities before generating certificates that bind those identities to keys.
- Certificates have a lifetime and must be maintained.

Digital Certificate Authorities, April, 2016 - Wikipedia

Rank	Issuer	Usage	Market Share
1	Comodo	8.1%	40.6%
2	Symantec	5.2%	26.0%
3	GoDaddy	2.4%	11.8%
4	GlobalSign	1.9%	9.7%
5	IdenTrust	0.7%	3.5%
6	DigiCert	0.6%	3.0%
7	StartCom	0.4%	2.1%
8	Entrust	0.1%	0.7%
9	Trustwave	0.1%	0.5%
10	Verizon	0.1%	0.5%

Certificate Signing and Hierarchy

- It is possible to create a certificate that includes another certificate
  - Creating a chain of trust

Certificate Signing and Hierarchy

To create Diana's certificate:

Diana creates and delivers to Edward:

Name: Diana Position: Division Manager Public key: 17EF83CA ...

Edward adds:

Name: Diana Position: Division Manager Public key: 17EF83CA ...	hash value 128C4

Edward signs with his private key:

Name: Diana Position: Division Manager Public key: 17EF83CA ...	hash value 128C4

Which is Diana's certificate.

To create Delwyn's certificate:

Delwyn creates and delivers to Diana:

Name: Delwyn Position: Dept Manager Public key: 3AB3882C ...

Diana adds:

Name: Delwyn Position: Dept Manager Public key: 3AB3882C ...	hash value 48CFA

Diana signs with her private key:

Name: Delwyn Position: Dept Manager Public key: 3AB3882C ...	hash value 48CFA

And appends her certificate:

Name: Delwyn Position: Dept Manager Public key: 3AB3882C ...	hash value 48CFA
Name: Diana Position: Division Manager Public key: 17EF83CA ...	hash value 128C4

Which is Delwyn's certificate.

Certificate Signing and Hierarchy - Example

- Diana's certificate is made using Edward's signature.
- Delwyn's certificate includes Diana's certificate so that it can effectively be tied back to Edward, creating a chain of trust.

Certificate Signing and Hierarchy

To create Diana's certificate:

Diana creates and delivers to Edward:

Name: Diana Position: Division Manager Public key: 17EF83CA ...

Edward adds:

Name: Diana Position: Division Manager Public key: 17EF83CA ...	hash value 128C4

Edward signs with his private key:

Name: Diana Position: Division Manager Public key: 17EF83CA ...	hash value 128C4

Which is Diana's certificate.

To create Delwyn's certificate:

Delwyn creates and delivers to Diana:

Name: Delwyn Position: Dept Manager Public key: 3AB3882C ...

Diana adds:

Name: Delwyn Position: Dept Manager Public key: 3AB3882C ...	hash value 48CFA

Diana signs with her private key:

Name: Delwyn Position: Dept Manager Public key: 3AB3882C ...	hash value 48CFA

And appends her certificate:

Name: Delwyn Position: Dept Manager Public key: 3AB3882C ...	hash value 48CFA
Name: Diana Position: Division Manager Public key: 17EF83CA ...	hash value 128C4

Which is Delwyn's certificate.

Digital Signatures

- A digital signature consists of
  - a file
  - demonstration that the file has not been altered
  - indication of who applied the signature
  - validation that the signature is authentic, that is, that it belongs to the signer
  - connection of the signature to the file

Complete Digital Signature

- Usage of a Digital Signature:
  - Includes an encrypted file
  - Ensuring Authenticity: a secure hash code of the file is used to compute a message digest
    - Included in the signature
    - Digest shows that the file has not changed
      - The recipient can recompute the hash function and compare with the message digest.
      - If the hash codes match, the recipient can conclude that the received file is the one that was signed.

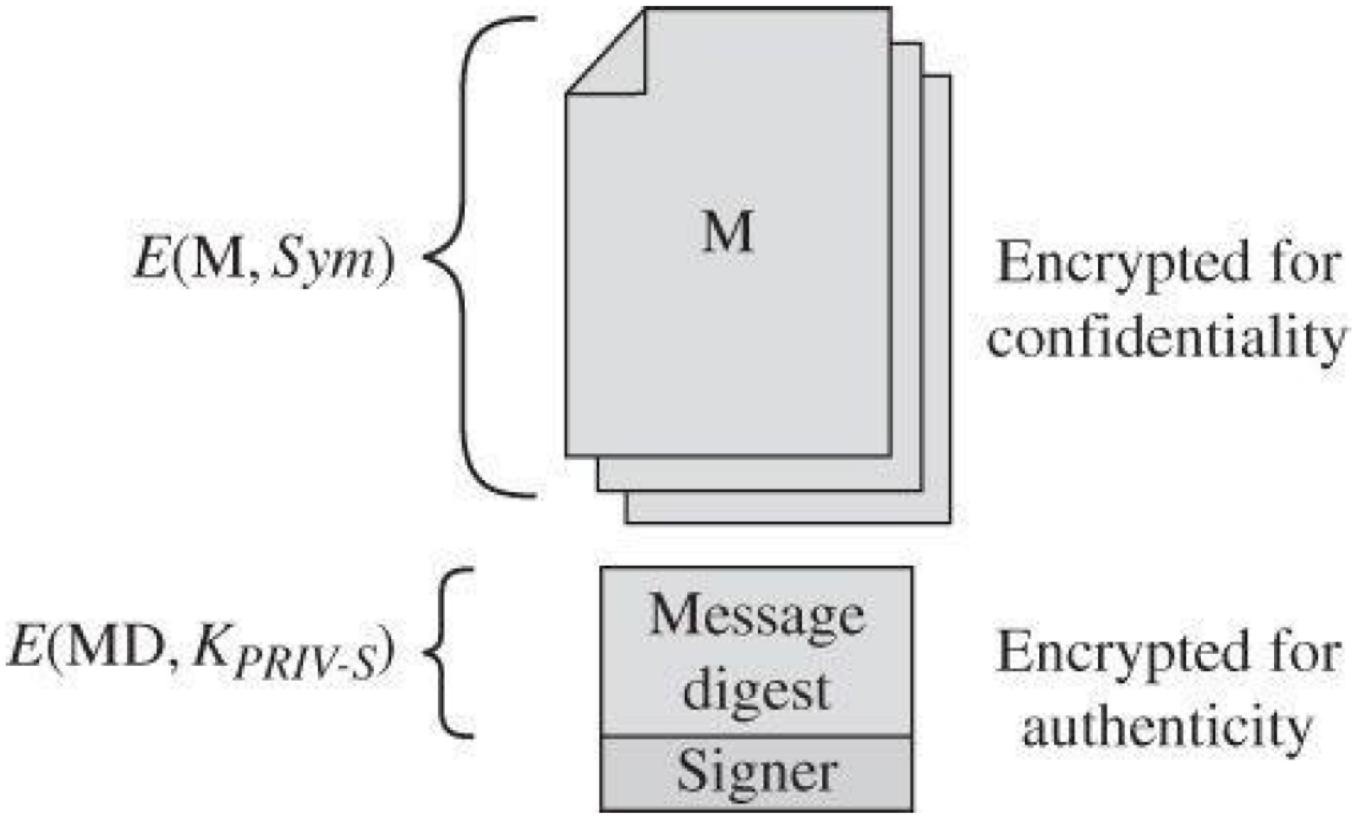
Complete Digital Signature

- Signer private key is used to encrypt the message digest
  - Only the signer has the private key
    - Authenticates the signer
- Signer identity is added unencrypted to the message
  - The recipient has to know who the signer was so that the correct public key is used to decrypt the message.

Complete Digital Signature

- Confidentiality:
  - If confidentiality is required, signer can encrypt the file
    - Using symmetric key encryption and the user key

Complete Digital Signature



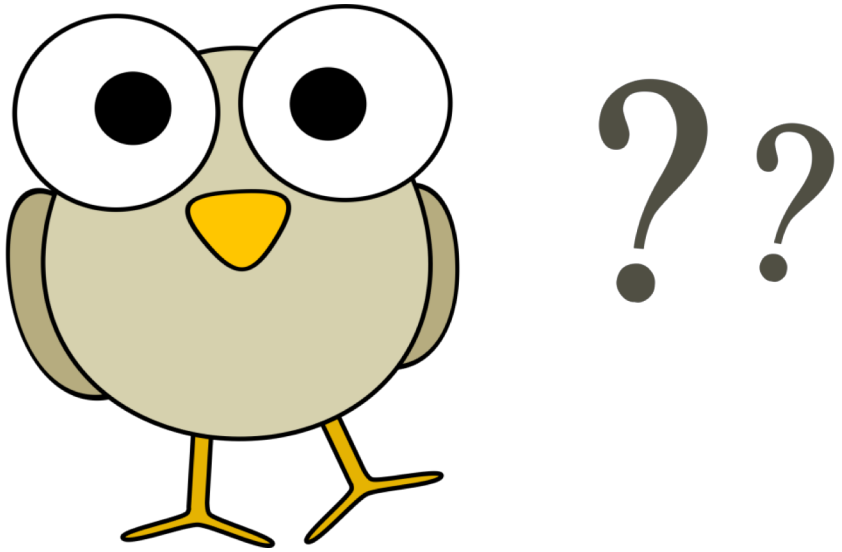
Digital Signatures

- a digital signature can indicate the authenticity of a file or a piece of code
- When you attempt to install a piece of signed code, the operating system will inspect the certificate and file
  - notify you if the certificate and hash are not acceptable
- Digital signatures provide an effective tool
  - coupled with strong hash functions and symmetric encryption, help ensure that a file is precisely what the originator stored for download

TOOLS DERIVED FROM CRYPTOGRAPHY

Tool	Uses
Secret key (symmetric) encryption	Protecting confidentiality and integrity of data at rest or in transit
Public key (asymmetric) encryption	Exchanging (symmetric) encryption keys Signing data to show authenticity and proof of origin
Error detection codes	Detect changes in data
Hash codes and functions (forms of error detection codes)	Detect changes in data
Cryptographic hash functions	Detect changes in data, using a function that only the data owner can compute (so an outsider cannot change both data and the hash code result to conceal the fact of the change)
Error correction codes	Detect and repair errors in data
Digital signatures	Attest to the authenticity of data
Digital certificates	Allow parties to exchange cryptographic keys with confidence of the identities of both parties

- Questions?



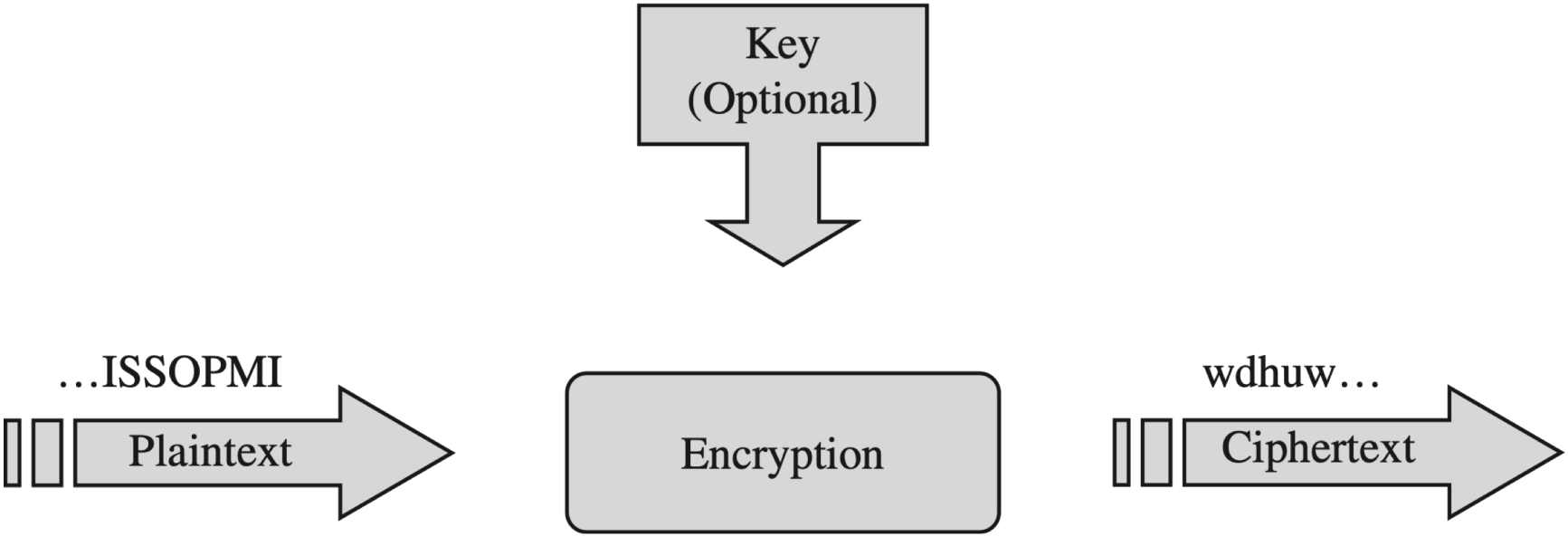
Stream Ciphers

- An alternative approach to using block cipher
- In stream ciphers, each byte of the data stream is encrypted separately
  - This is as opposed to block ciphers

Stream Ciphers

- Key stream
  - Pseudo-random sequence of bits S = S[0], S[1], S[2], …
  - Can be generated on-line one bit (or byte) at the time
  - Successive elements of the keystream generated based on an internal state
- Stream cipher
  - XOR the plaintext with the key stream C[i] = S[i] ⊕ P[i]
  - Suitable for plaintext of arbitrary length generated on the fly, e.g., media stream

Stream Ciphers



Key Stream Generation

- RC4
  - Designed in 1987 by Ron Rivest for RSA Security
  - Trade secret until 1994
  - Uses keys with up to 2,048 bits
  - Simple algorithm

Key Stream Generation

- Block cipher in counter mode (CTR)
  - Use a block cipher with block size b
  - The secret key is a pair (K,t), where K a is key and t (counter) is a b-bit value
  - The key stream is the concatenation of ciphertexts
  - $$E_K(t), E_K(t + 1), E_K(t + 2), \ldots$$
  - Can use a shorter counter concatenated with a random value
  - Synchronous stream cipher

- Questions?