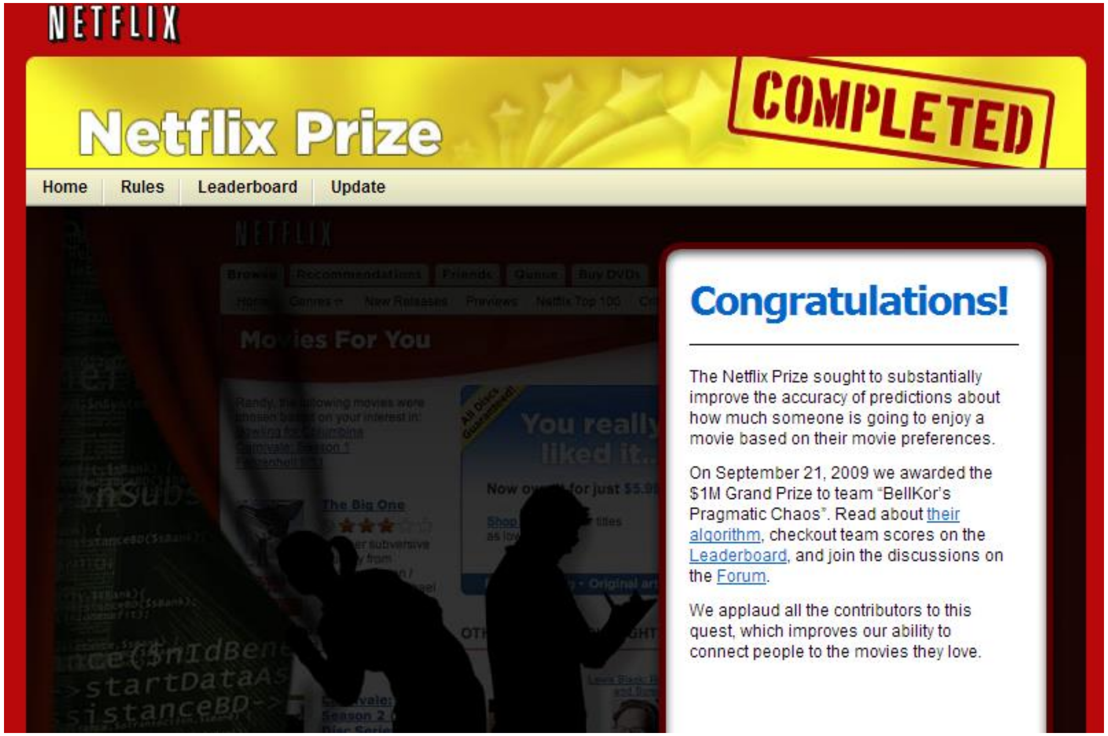
ENEE 459-C
Computer Security

Introduction
(continue from previous lecture)

UNIVERSITY OF
MARYLAND

Netflix incident



- See paper
  - http://www.cs.utexas.edu/~shmat/shmat_oak08netflix.pdf

Google chrome saving passwords

- In a previous version of Google chrome, if you saved your password so that you do not have to put it in every time you log into a website (a bad idea in general), one could retrieve the password with three clicks
- No need for the password to be displayed in plaintext
- http://www.itpro.co.uk/security/20363/google-chrome-password-access-bug-discovered

The human factor

- In 2010, Google fired an employee because he was caught snooping on user's data

Some Security Principles

Economy of mechanism

- This principle stresses **simplicity** in the **design** and **implementation** of security measures
  - Example: Avoid multiple interconnecting software modules (running in different machines) to implement a security property (e.g., input of password in one machine, checking of password in another)

Complete mediation

- The idea behind this principle is that every access to a resource must be checked for **compliance with a protection scheme**
  - As a consequence, one should be wary of performance improvement techniques that save the results of previous authorization checks, since permissions can change over time
  - For example, an online banking web site should require users to sign on again after a certain amount of time, say, 15 minutes, has elapsed

Open design

- According to this principle, the security architecture and **design** of a system should be made **publicly available**
  - Security should rely only on keeping cryptographic keys secret
  - Open design allows for a system to be scrutinized by multiple parties, which leads to the early discovery and correction of security vulnerabilities caused by design errors
  - The open design principle is the opposite of the approach known as **security by obscurity,** which tries to achieve security by keeping cryptographic algorithms secret and which has been historically used without success by several organizations

Separation of privilege

- Try to isolate software components to limit the damage that can be caused in a computer system
- E.g., when one runs a virtual machine, then an attack on some software running in the virtual machine cannot affect files in the host machine (these are different machines)

Least privilege

- Each program and user of a computer system should operate with the bare **minimum privileges necessary** to function properly
  - If this principle is enforced, abuse of privileges is restricted, and the damage caused by the compromise of a particular application or user account is minimized
  - The military concept of **need-to-know** information is an example of this principle

Least common mechanism

- In systems with multiple users, mechanisms allowing resources to be **shared by more than one user should be minimized**

Psychological acceptability

- This principle states that user interfaces should be **well designed and intuitive**, and all security-related settings should adhere to what an ordinary user might expect

Work factor

- According to this principle, the **cost of circumventing** a security mechanism should be compared with the resources of an attacker when designing a security scheme
  - A system developed to protect student grades in a university database, which may be attacked by snoopers or students trying to change their grades, probably needs less sophisticated security measures than a system built to protect military secrets, which may be attacked by government intelligence organizations

Compromise recording

- This principle states that sometimes it is more desirable to **record the details** of an intrusion than to adopt more sophisticated measures to prevent it
  - Internet-connected surveillance cameras are a typical example of an effective compromise record system that can be deployed to protect a building in lieu of reinforcing doors and windows
  - The servers in an office network may maintain logs for all accesses to files, all emails sent and received, and all web browsing sessions

Computer Security Goals

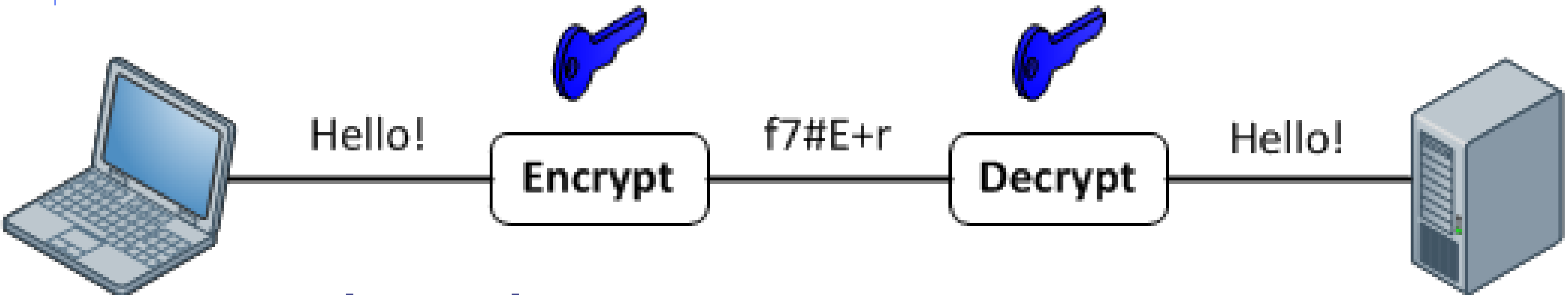
- Confidentiality
- Integrity
- Availability
- Authenticity
- Anonymity

Confidentiality

- It is the avoidance of the unauthorized disclosure of information
- It involves the protection of data, providing access for those who are allowed to see it while disallowing others from learning anything about its content
- E.g., nobody should be able to read the emails I am sending to my friends, except for my friends

Tools for confidentiality

- **Encryption:** the transformation of information using a secret, called an encryption key, so that the transformed information can only be read using another secret, called

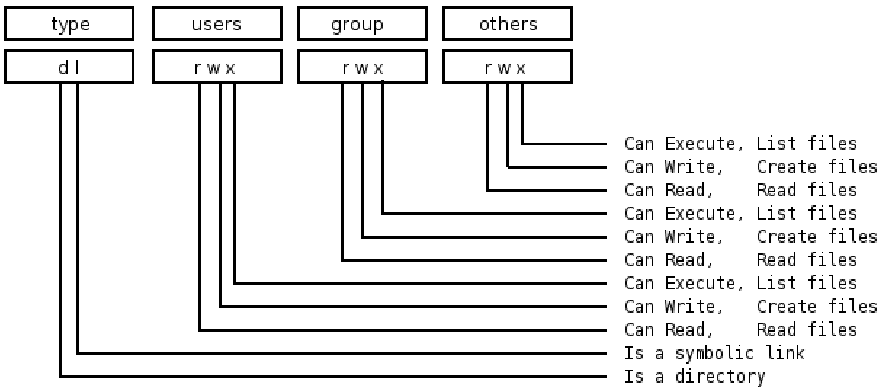


Discussion: To encrypt a message 1000 bits, create a secret key of 100 bits and XOR 100-bit blocks sequentially.

Does this reveal the content of the message?
Is this good enough?

Tools for confidentiality

- **Access control:** rules and policies that limit access to confidential information to those people and/or systems with a "need to know"
    - This need to know may be determined by identity, such as a person's name or a computer's serial number, or by a role that a person has, such as being a manager or a computer security specialist



Tools for confidentiality

- **Authentication:** the determination of the identity or role that someone has. This determination can be done in a number of different ways, but it is usually based on a combination of
  - **something the person has** (like a smart card or a radio key fob storing secret keys)
  - **something the person knows** (like a password)
  - **something the person is** (like a human with a fingerprint)

Integrity

- The property that information has not be altered in an unauthorized way
- **Tools:**
  - **Checksums:** the computation of a function that maps the contents of a file to a numerical value. A checksum function depends on the entire contents of a file and is designed in a way that even a small change to the input file (such as flipping a single bit) is highly likely to result in a different output value.
  - **Discussion:** Can we use the checksum f(x) = x mod M?

Availability

- **Availability:** the property that information is accessible and modifiable in a timely fashion by those authorized to do so
- **Tools:**
  - **Physical protections:** infrastructure meant to keep information available even in the event of physical challenges.
  - **Computational redundancies:** computers and storage devices that serve as back-ups in the case of failures
    - **E.g., error-correcting codes**

Other important Security goals

- Authenticity

- Anonymity

Authenticity

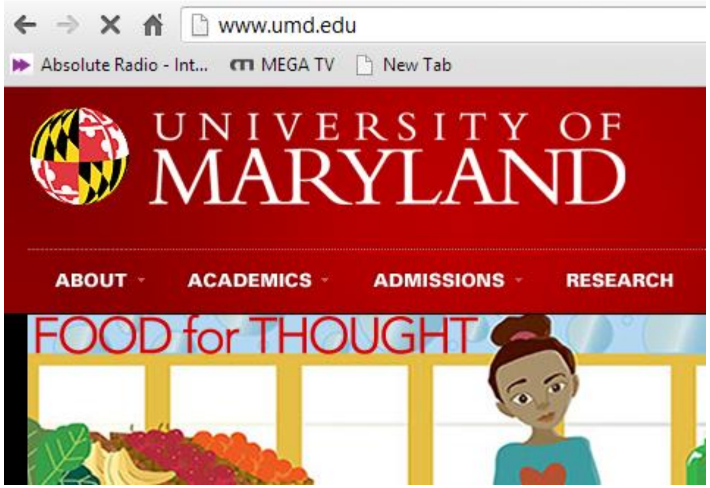
- **Authenticity** is the ability to determine that statements, policies, and permissions issued by persons or systems are genuine
- **Primary tool:**
  - **Digital signatures.** These are cryptographic computations that allow a person or system to commit to the authenticity of their documents in a unique way that achieves **nonrepudiation**, which is the property that authentic statements issued by some person or system cannot be denied

Anonymity

- **Anonymity:** the property that certain records or transactions not to be attributable to any individual
- **Tools:**
  - **Aggregation:** the combining of data from many individuals so that disclosed sums or averages cannot be tied to any individual
  - **Proxies:** trusted agents that are willing to engage in actions for an individual in a way that cannot be traced back to that person
  - **Pseudonyms:** fictional identities that can fill in for real identities in communications and transactions, but are otherwise known only to a trusted entity

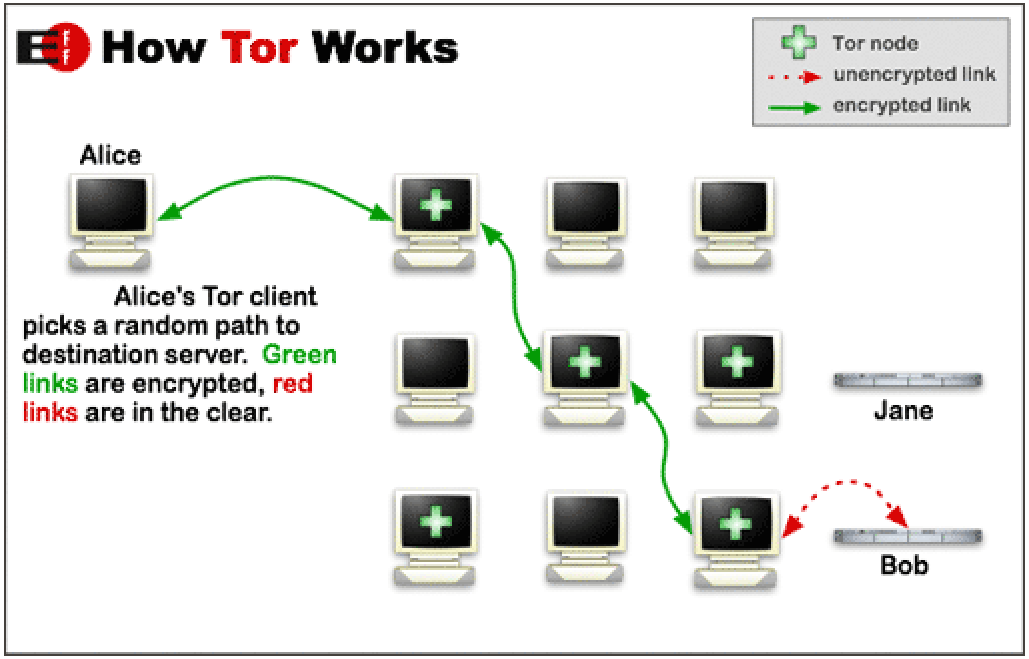
Examples: HTTPS protocol

- Confidentiality
- Integrity
- Availability
- Authenticity
- Anonymity



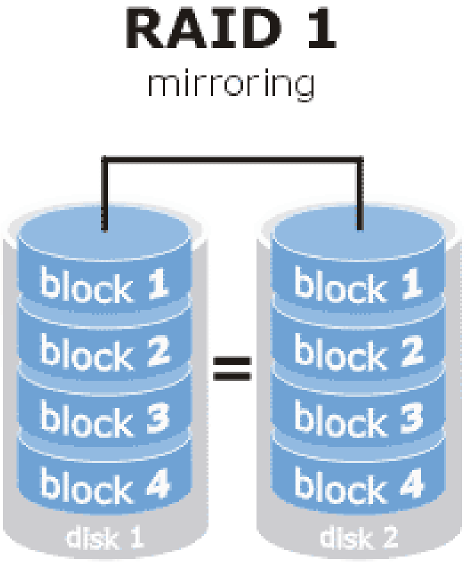
Examples: TOR protocol

- Confidentiality
- Integrity
- Availability
- Authenticity
- <u>Anonymity</u>



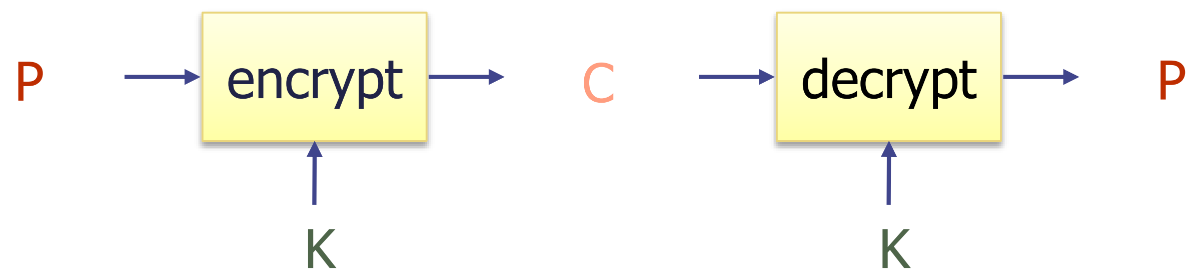
Examples: RAID technology

- Confidentiality
- Integrity
- Availability
- Authenticity
- Anonymity



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)
- 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?

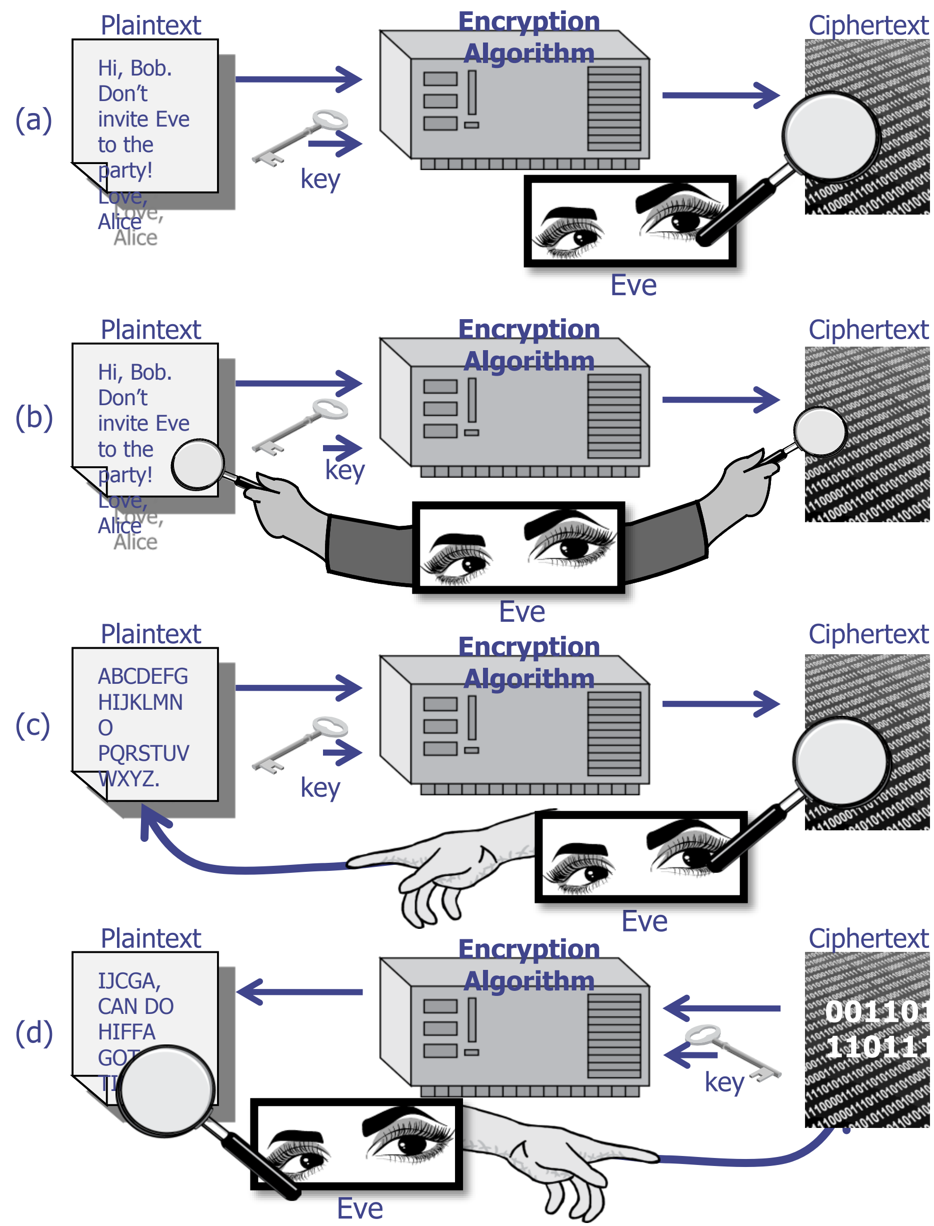


Basics

- Notation
  - Secret key K
  - Encryption function $E_K(P)$
  - Decryption function $D_K(C)$
  - Plaintext length typically the same as ciphertext length
  - Encryption and decryption are permutation functions (bijections) on the set of all n-bit arrays
- Efficiency
  - functions E_K and D_K should have efficient algorithms
- Consistency
  - Decrypting the ciphertext yields the plaintext
  - $D_K(E_K(P)) = P$

Attacks

- Attacker may have
  a) collection of ciphertexts (ciphertext only attack)
  b) collection of plaintext/ciphertext pairs (known plaintext attack)
  c) collection of plaintext/ciphertext pairs for plaintexts selected by the attacker (chosen plaintext attack)
  d) collection of plaintext/ciphertext pairs for plaintexts and ciphertexts selected by the attacker (chosen ciphertext attack or lunchtime attack)



Randomized encryption

- Encryption should be randomized
  - For the same plaintext, it should output different ciphertexts
- How can we turn a deterministic encryption scheme into a randomized one?
  - Padding input with randomness
- Decryption should however always work

Brute-Force Attack

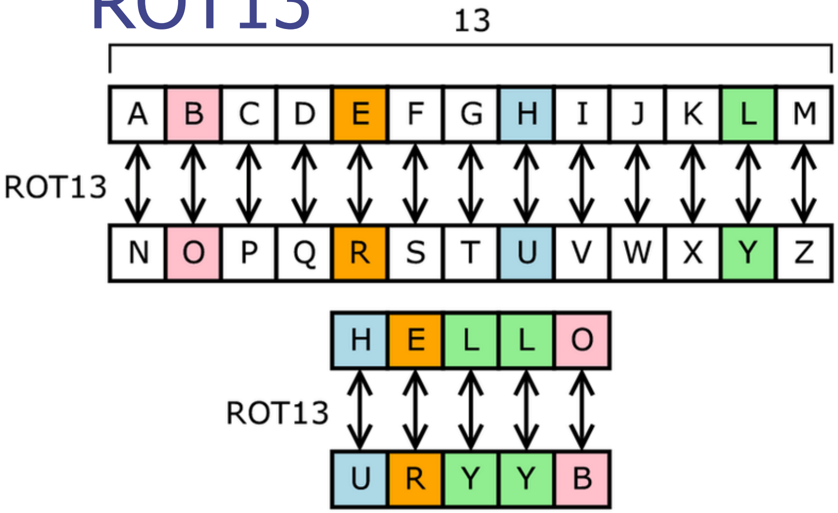
- Try all possible keys K and determine if $D_K(C)$ is a likely plaintext
  - Requires some knowledge of the structure of the plaintext (e.g., PDF file or email message)
- Key should be a sufficiently long random value to make exhaustive search attacks unfeasible

Image by Michael Cote from http://commons.wikimedia.org/wiki/File:Bingo_cards.jpg

Substitution Ciphers

- Each letter is uniquely replaced by another
- There are 26! possible substitution ciphers

- One popular substitution "cipher" for some Internet posts is ROT13



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.

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.

Semantic security

- I give you a symmetric encryption scheme (Enc,Dec,K)
- What do you need to prove in order to say that it is secure?
- A strong notion used is “semantic security”
- Informally:
  - An attacker picks messages m_i and receives ciphertexts Enc_K(m_i)
  - An attacker picks message m_0 and m_1 and sends them to the bank that has the secret key
  - The bank flips a coin b and computes t_b=Enc_K(m_b)
  - The bank sends t_b to the attacker
  - The scheme is secure if the attacker has no better chance of finding whether t_b corresponds to m_0 or m_1 than just guessing!
- This should hold even if it is repeated many (polynomial) times

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, $(k_1, k_2, \ldots, k_n)$, 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

Algorithms

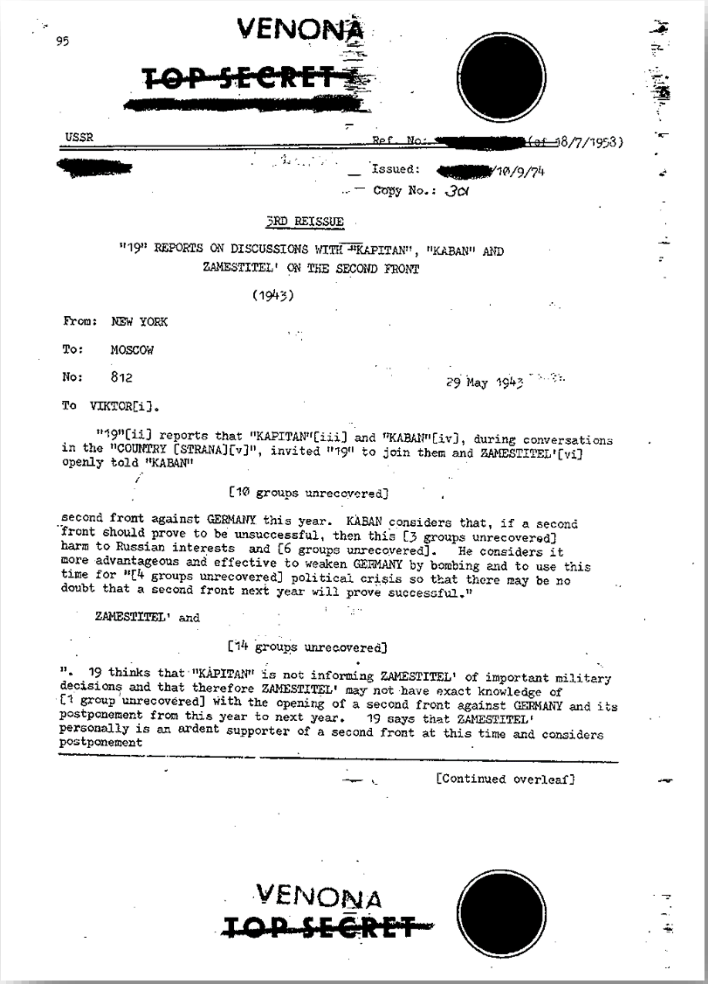
- $K \leftarrow KeyGen(n)$: Pick a random key K of n bits
- $E_K(A)$: On input plaintext A, compute ciphertext B=A XOR K
- $D_K(B)$: On input ciphertext B, compute plaintext A=B XOR K
- Correctness: B XOR K= (A XOR K) XOR K= A XOR 0 = A
- Security?

Perfect security

- For all messages m_1 and m_2 and for all ciphertexts c
- $\Pr[K \leftarrow \mathrm{KeyGen}(n): E_K(m_1)=c]=$
  $\Pr[K \leftarrow \mathrm{KeyGen}(n): E_K(m_2)=c]$
- Proof
  - Note that $Enc_K(m_1)=c$ is the event m_1 XOR K = c which is the event K = m_1 XOR c
  - K is chosen at random (irrespective of m_1 and m_2, and therefore the probability is 2^{-n}
  - Namely ciphertext does not reveal anything about the plaintext

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 compromised communications during the cold war



95

VENONA

~~TOP SECRET~~

[redacted]

USSR

Ref. No.: [redacted] (of 18/7/1953)

[redacted]

Issued: [redacted]/10/9/74

Copy No.: 301

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]

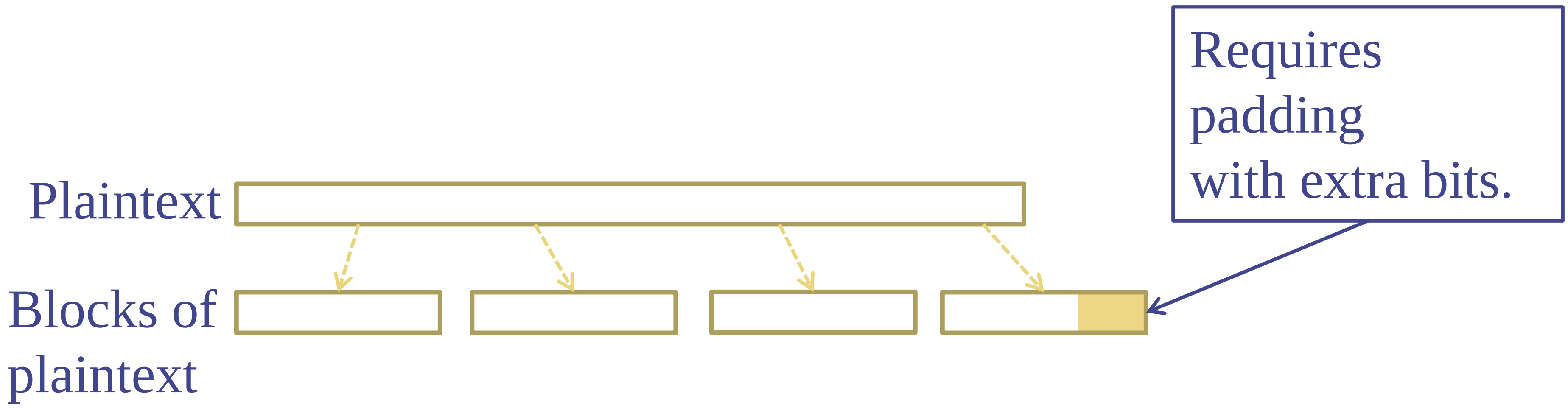
VENONA

~~TOP SECRET~~

Public domain declassified government image from
https://www.cia.gov/library/center-for-the-study-of-intelligence/csi-publications/books-and-monographs/venona-soviet-espionage-and-the-american-response-1939-1957/part2.htm

Block Ciphers

- In a **block cipher:**
  - Plaintext and ciphertext have fixed length b (e.g., 128 bits)
  - A plaintext of length n is partitioned into a sequence of m **blocks**, P[0], …, P[m–1], where $n \leq bm < n + b$
- Each message is divided into a sequence of blocks and encrypted or decrypted in terms of its blocks



Block Ciphers in Practice

- Data Encryption Standard (DES)
  - Developed by IBM and adopted by NIST in 1977
  - 64-bit blocks and 56-bit keys
  - Small key space makes exhaustive search attack feasible since late 90s
- Triple DES (3DES)
  - Nested application of DES with three different keys KA, KB, and KC
  - Effective key length is 168 bits, making exhaustive search attacks unfeasible
  - $C = E_{KC}(D_{KB}(E_{KA}(P)))$; $P = D_{KA}(E_{KB}(D_{KC}(C)))$
  - Equivalent to DES when KA=KB=KC (backward compatible)
- Advanced Encryption Standard (AES)
  - Selected by NIST in 2001 through open international competition and public discussion
  - 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**

A perfect encryption of a block

- Say you have a block of n bits
- You want to encrypt it
- You want to use the same key all the time but NOT have the problem of ONE TIME PAD (i.e., be semantically secure)
- Consider a bijective mapping T from $\{0,1\}^n$ to $\{0,1\}^n$
- The pairs are computed uniformly at random
- To encrypt x, just output T[x]
- To decrypt y, just output $T^{-1}[y]$
- Your secret key is T
- Problem with this approach: T has size ~ n 2^n
- Can you make it randomized (and semantically-secure)?
  - Encrypt x (pick random r): y=T[r] XOR x, r
  - Decrypt (y,r): y XOR $T^{-1}[r]$