Introduction to Cryptography

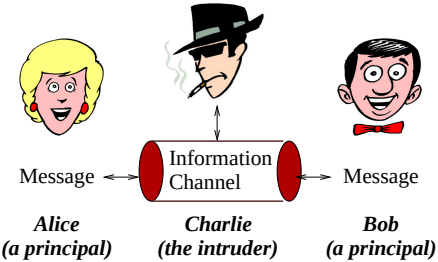
Alessandro Armando

Computer Security Laboratory (CSec)
DIBRIS, University of Genova

Computer Security
Corso di Laurea Magistrale in Ingegneria Informatica

Outline

What's it all about?

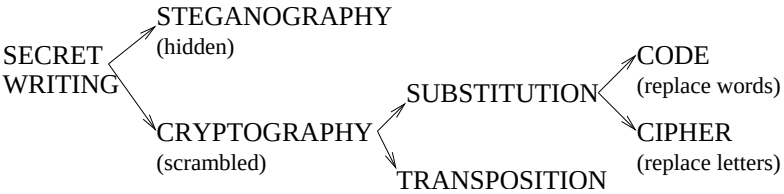


- How do we turn untrustworthy channels into trustworthy ones?

  Confidentiality: Transmitted information remains secret.
  Integrity: Information not corrupted (or alterations detected).
  Authentication: Principals know who they are speaking to.

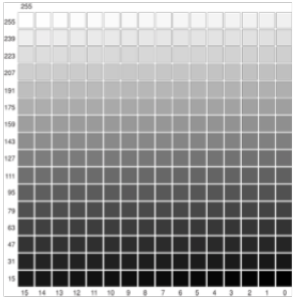
- Other goals desirable. E.g., non-repudiation.
- Cryptography is the enabling technology.

Information hiding

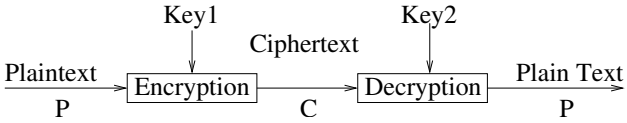


- Cryptology: the study of secret writing.
- Steganography: the science of hiding messages in other messages.
- Cryptography: the science of secret writing.

Steganography



General cryptographic schema



where $E_{key_1}(P) = C$, $D_{key_2}(C) = P$

- **Security depends on secrecy of the key, not of the algorithm**
- Symmetric algorithms
  - Key1 = Key2, or are easily derived from each other.
- Asymmetric or public key algorithms
  - Different keys, which cannot be derived from each other.
  - Public key can be published without compromising private key.
- Encryption and decryption should be easy, if keys are known.

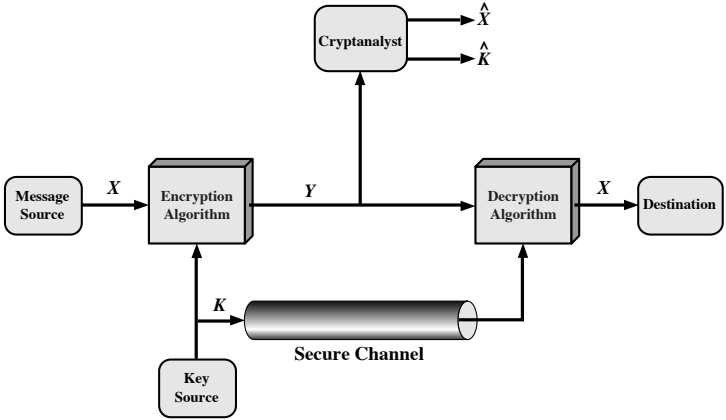
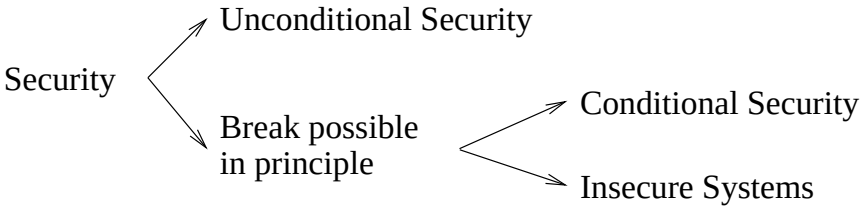


Figure 2.2 Model of Symmetric Cryptosystem

Classification of security



Unconditional Security: System is secure even if adversary has unbounded computing power since the ciphertext provides insufficient information to uniquely determine the corresponding plaintext. Security measured using information theory.

Conditional Security: System can be broken in principle, but this requires more computing power than a realistic adversary would have. Security measured using complexity theory.

Cryptoanalysis

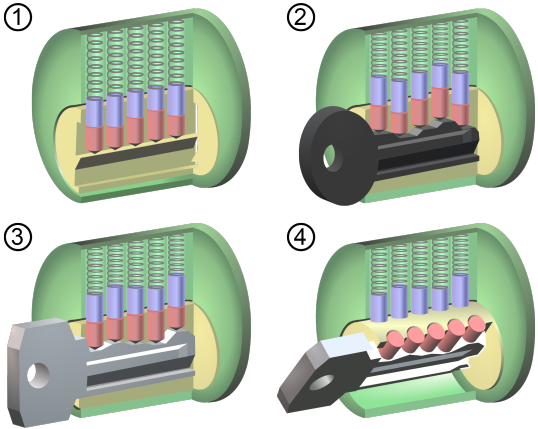
- Cryptanalysis: science of recovering the plaintext from ciphertext without the key.
- But typical objective is to recover key not just message
- General approaches:
  - brute-force attack
  - cryptanalytic attack

Brute-force Attack

- It is always possible: simply try every key
- Its cost (eavily) depends on key size
- It assumes that plaintext is known or recognisable

Key size (bits)	Number of keys	Time required at 1 decryption/μs	Time required at 10^6 decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\mu$s=35.8 min	2.15 ms
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\mu$s=1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\mu$s=5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\mu$s=5.9×10^{36} years	5.9×10^{30} years
26 chars (permut.)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\mu$s=6.4×10^{12} years	6.4×10^6 years

Analogy with Physical Security: Tumbler Key Lock



Analogy with Physical Security: Safes

- Security of safes is rated according to the degree of protection provided against an attempted burglary attack.
- Class TL-X (for $X = 15, 30, 40$):[1]

  The door successfully resist entry for [...] X minutes when attacked with common hand tools, picking tools, mechanical or portable electric tools, grinding points, carbide drills and pressure applying devices or mechanisms.

- assuming that[2]

  the laboratory receives the plans of the samples to be tested in advance so that the vault, its structure, its composition, its assembly and any weak points can be examined in depth beforehand

[1] https://www.safeandvaultstore.com/pages/burglary-ratings
[2] https://www.bunkerkit.com/en/performances/vault/

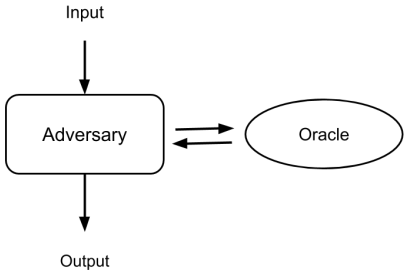
Model of Attack

We can think of the adversary as playing a game:

Input: Whatever adversary necessarily knows from the beginning, e.g., public key, distribution of plain texts, etc.

Oracle: Models information adversary can obtain during an attack. Different kinds of information characterize different types of attacks.

Output: Whatever the adversary wants to compute, e.g., secret key, partial information on plain text, etc. He wins if he succeeds.

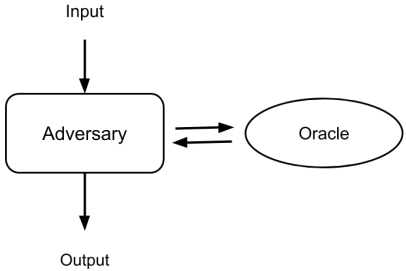


Kinds of attacks

- **Ciphertext only**
  - **Given:** $C_1 = E_K(M_1), \ldots, C_n = E_K(M_n)$
  - **Deduce:** $M_1, \ldots, M_n$ or algorithm to compute M_{n+1} from $C_{n+1} = E_K(M_{n+1})$
- **Known plaintext**
  - **Given:** $M_1, C_1 = E_K(M_1), \ldots, M_n, C_n = E_K(M_n)$
  - **Deduce:** Inverse key or algorithm to compute M_{n+1} from $C_{n+1} = E_K(M_{n+1})$
- **Chosen plaintext** Same as above but cryptanalyst may choose $M_1, \ldots, M_n$.
- **Adaptive chosen plaintext** Cryptanalyst can not only choose plaintext, but he can modify the plaintext based on encryption results.
- **Chosen ciphertext** Cryptanalyst can chose different ciphertexts to be decrypted and gets access to the decrypted plaintext.

How to build a definition of security

1. Specify an oracle (a type of attack).
2. Define what the adversary needs to do to win the game, i.e., a condition on his output.
3. The system is secure under the definition, if any efficient adversary wins the game with only negligible probability.



Outline

Encryption/decryption

- $\mathcal{A}$, the alphabet, is a finite set.
- $\mathcal{M} \subseteq \mathcal{A}^*$ is the message space. $M \in \mathcal{M}$ is a plaintext (message).
- $\mathcal{C}$ is the ciphertext space, whose alphabet may differ from $\mathcal{M}$.
- $\mathcal{K}$ denotes the key space of keys.
- Each $e \in \mathcal{K}$ determines a bijective function from $\mathcal{M}$ to $\mathcal{C}$, denoted by E_e. E_e is the encryption function (or transformation).
- For each $d \in \mathcal{K}$, D_d denotes a bijection from $\mathcal{C}$ to $\mathcal{M}$.
  D_d is the decryption function.
- Applying E_e (or D_d) is called encryption (or decryption).

Encryption/decryption (cont.)

- An encryption scheme (or cipher) consists of a set $\{E_e : e \in \mathcal{K}\}$ and a corresponding set $\{D_d : d \in \mathcal{K}\}$ with the property that for each $e \in \mathcal{K}$ there is a unique $d \in \mathcal{K}$ such that $D_d = E_e^{-1}$; i.e.,
$$D_d(E_e(m)) = m \qquad \text{for all } m \in \mathcal{M}\,.$$
- The keys e and d above form a key pair, sometimes denoted by (e, d). They can be identical (i.e., the symmetric key).
- To construct an encryption scheme requires fixing a message space $\mathcal{M}$, a ciphertext space $\mathcal{C}$, and a key space $\mathcal{K}$, as well as encryption transformations $\{E_e : e \in \mathcal{K}\}$ and corresponding decryption transformations $\{D_d : d \in \mathcal{K}\}$.

An example

Let $\mathcal{M} = \{m_1, m_2, m_3\}$ and $\mathcal{C} = \{c_1, c_2, c_3\}$. There are 3! = 6 bijections from $\mathcal{M}$ to $\mathcal{C}$. The key space $\mathcal{K} = \{1, 2, 3, 4, 5, 6\}$ specifies these transformations.

	E_1			E_2			E_3	
m_1	$\rightarrow$	c_1	m_1	$\rightarrow$	c_1	m_1	$\rightarrow$	c_2
m_2	$\rightarrow$	c_2	m_2	$\rightarrow$	c_3	m_2	$\rightarrow$	c_1
m_3	$\rightarrow$	c_3	m_3	$\rightarrow$	c_2	m_3	$\rightarrow$	c_3

	E_4			E_5			E_6	
m_1	$\rightarrow$	c_2	m_1	$\rightarrow$	c_3	m_1	$\rightarrow$	c_3
m_2	$\rightarrow$	c_3	m_2	$\rightarrow$	c_1	m_2	$\rightarrow$	c_2
m_3	$\rightarrow$	c_1	m_3	$\rightarrow$	c_2	m_3	$\rightarrow$	c_1

Suppose Alice and Bob agree on the transformation E_6. To encrypt m_1, Alice computes $E_6(m_1) = c_3$. Bob decrypts c_3 by reversing the arrows on the diagram for E_6 and observing that c_3 points to m_1.

Outline

Symmetric key encryption

- Consider an encryption scheme $\{E_e : e \in \mathcal{K}\}$ and $\{D_d : d \in \mathcal{K}\}$. The scheme is symmetric-key if for each associated pair (e, d) it is computationally "easy" to determine d knowing only e and to determine e from d. In practice $e = d$.
- Other terms: single-key, one-key, shared-key, and conventional encryption.
- sender and recipient share a common key
- all classical encryption algorithms are simmetric-key (it was the only type of encryption prior to invention of public-key in 1970's)
- by far most widely used

Block ciphers, stream ciphers, and codes

- A block cipher is an encryption scheme that breaks up the plaintext message into strings (blocks) of a fixed length t and encrypts one block at a time.
- A stream cipher is one where the block-length is 1.
- In contrast, codes work on words of varying length.

Codes

- Translation given by a ‘code-book’.

Word	Code
...	...
The	1701
secret	5603
mischiefs	4008
that	3790
I	2879
set	0524
...	...

Codes

- Translation given by a ‘code-book’.

Word	Code
...	...
The	1701
secret	5603
mischiefs	4008
that	3790
I	2879
set	0524
...	...

```
2327 6605 1702 9853 0001 0970 3190 8817 1320 0000 =
1701 5603 4008 3790 2879 0524 7946                =
2879 2870 6699 1702 3982 5550 8102 7354 0000      =
```

Codes

- Translation given by a ‘code-book’.

Word	Code
...	...
The	1701
secret	5603
mischiefs	4008
that	3790
I	2879
set	0524
...	...

```
2327 6605 1702 9853 0001 0970 3190 8817 1320 0000 = I do the wrong, and first begin to brawl.
1701 5603 4008 3790 2879 0524 7946               = The secret mischiefs that I set abroach
2879 2870 6699 1702 3982 5550 8102 7354 0000     = I lay unto the grievous charge of others.
```

(Richard III, Act I, Scene 3)

Outline

Simple Substitution ciphers

- **KHOOR ZRUOG** = **HELLO WORLD**
  Caesar cipher: each plaintext character is replaced by the character three to the right modulo 26.
- **Zl anzr vf Nqnz** = **My name is Adam**
  ROT13: shift each letter by 13 places.
  Under Unix:
  tr a-zA-Z n-za-mN-ZA-M
- **2-25-5 2-25-5** = **BYE BYE**
  Alphanumeric: substitute numbers for letters.

How hard are these to cryptanalyze? Caesar? General?

Simple Substitution ciphers

- **KHOOR ZRUOG** = **HELLO WORLD**
  Caesar cipher: each plaintext character is replaced by the character three to the right modulo 26.
- **Zl anzr vf Nqnz** = **My name is Adam**
  ROT13: shift each letter by 13 places.
  Under Unix:
  tr a-zA-Z n-za-mN-ZA-M
- **2-25-5 2-25-5** = **BYE BYE**
  Alphanumeric: substitute numbers for letters.

How hard are these to cryptanalyze? Caesar? General?

Simple Substitution ciphers

- **KHOOR ZRUOG** = **HELLO WORLD**
  Caesar cipher: each plaintext character is replaced by the character three to the right modulo 26.
- **Zl anzr vf Nqnz** = **My name is Adam**
  ROT13: shift each letter by 13 places.
  Under Unix:
  `tr a-zA-Z n-za-mN-ZA-M`
- **2-25-5 2-25-5** = **BYE BYE**
  Alphanumeric: substitute numbers for letters.

How hard are these to cryptanalyze? Caesar? General?

Simple Substitution ciphers

- **KHOOR ZRUOG** = **HELLO WORLD**
  Caesar cipher: each plaintext character is replaced by the character three to the right modulo 26.
- **Zl anzr vf Nqnz** = **My name is Adam**
  ROT13: shift each letter by 13 places.
  Under Unix:
  tr a-zA-Z n-za-mN-ZA-M
- **2-25-5 2-25-5** = **BYE BYE**
  Alphanumeric: substitute numbers for letters.

How hard are these to cryptanalyze? Caesar? General?

Simple Substitution ciphers

- **KHOOR ZRUOG** = **HELLO WORLD**
  Caesar cipher: each plaintext character is replaced by the character three to the right modulo 26.
- **Zl anzr vf Nqnz** = **My name is Adam**
  ROT13: shift each letter by 13 places.
  Under Unix:
  tr a-zA-Z n-za-mN-ZA-M
- **2-25-5 2-25-5** = **BYE BYE**
  Alphanumeric: substitute numbers for letters.

How hard are these to cryptanalyze? Caesar? General?

KEY	PHHW PH DIWHU WKH WRJD SDUWB
1	oggv og chvgt vjg vqic rctva
2	nffu nf bgufs uif uphb qbsuz
3	meet me after the toga party
4	ldds ld zesdq sgd snfz ozqsx
5	kccr kc ydrcp rfc rmey nyprw
6	jbbq jb xcqbo qeb qldx mxoqv
7	iaap ia wbpan pda pkcw lwnpu
8	hzzo hz vaozm ocz ojbv kvmot
9	gyyn gy uznyl nby niau julns
10	fxxm fx tymxk max mhzt itkmr
11	ewwl ew sxlwj lzw lgys hsjlq
12	dvvk dv rwkvi kyv kfxr grikp
13	cuuj cu qvjuh jxu jewq fqhjo
14	btti bt puitg iwt idvp epgin
15	assh as othsf hvs hcuo dofhm
16	zrrg zr nsgre gur gbtn cnegl
17	yqqf yq mrfqd ftq fasm bmdfk
18	xppe xp lqepc esp ezrl alcej
19	wood wo kpdob dro dyqk zkbdi
20	vnnc vn jocna cqn cxpj yjach
21	ummb um inbmz bpm bwoi xizbg
22	tlla tl hmaly aol avnh whyaf
23	skkz sk glzkx znk zumg vgxze
24	rjjy rj fkyjw ymj ytlf ufwyd
25	qiix qi ejxiv xli xske tevxc

Mono-alphabetic substitution ciphers

- Key idea: generalise Caesar cipher by allowing an arbitrary substitution.
- Let $\mathcal{K}$ be the set of all permutations on the alphabet $\mathcal{A}$. Define for each $e \in \mathcal{K}$ an encryption transformation E_e on strings $m = m_1 m_2 \cdots m_n \in \mathcal{M}$ as

$$E_e(m) = e(m_1)e(m_2)\cdots e(m_n) = c_1 c_2 \cdots c_n = c$$

- To decrypt c, compute the inverse permutation $d = e^{-1}$ and

$$D_d(c) = d(c_1)d(c_2)\cdots d(c_n) = m$$

- E_e is a simple substitution cipher or a mono-alphabetic substitution cipher.

Example:

Plain:	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher:	DKVQFIBJWPESCXHTMYAUOLRGZN
Plaintext:	IFWEWISHTOREPLACELETTERS
Ciphertext:	WIRFRWAJUHYFTSDVFSFUUFYA

Example: Affine Ciphers

- An *affine cipher* is a monoalphabetic substitution cipher such that

$$e(m) = (a \cdot m + b) \mod |\mathcal{A}|$$

  where a and b are positive integers and are the key of the cipher.
- For the cipher to be invertible, a and $|\mathcal{A}|$ must be *relatively prime*, i.e. the only positive integer that divides both of them must be 1.
- The decryption function is

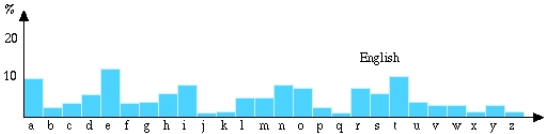
$$D(c) = a^{-1}(c - b) \mod |\mathcal{A}|,$$

  where a^{-1} is the modular multiplicative inverse of a modulo $|\mathcal{A}|$, i.e. it satisfies the equation

$$1 = a \cdot a^{-1} \mod |\mathcal{A}|$$

(In)security of substitution ciphers

- Key spaces are typically huge. 26 letters $\rightsquigarrow$ 26! possible keys.
- Yet, they can be easily cracked using frequency analysis (letters, digraphs, etc.).
- Frequencies for English based on data-mining books/articles.

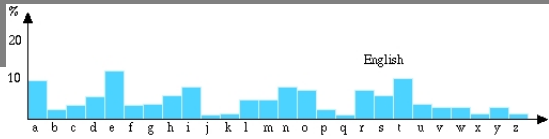


$\Rightarrow$ Serdhrapvrf sbe Ratyvfu onfrq ba qngn-zvavat obbxf/negvpyrf.

- Easy to apply, except for short, atypical texts
  From Zanzibar to Zambia and Zaire, ozone zones make zebras run zany zigzags.

$\Rightarrow$ More sophistication required to mask statistical regularities.

Example



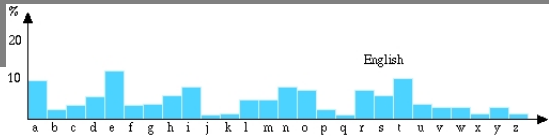
Given ciphertext:

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

- Count relative letter frequencies
- Since **P** and **Z** occur most frequently, guess they correspond to **E** and **T** respectively.
- Count relative digram frequencies.
- Since **ZW** occurs most frequently, guess it corresponds to **TH** (which is the digram occurring most frequently in English)
- Hence **ZWP** is **THE**
- By proceeding with trial and error finally get:

IT WAS DISCLOSED YESTERDAY THAT SEVERAL INFORMAL BUT DIRECT
CONTACTS HAVE BEEN MADE WITH POLITICAL REPRESENTATIVES OF THE
VIET CONG IN MOSCOW

Example



Given ciphertext:

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

- Count relative letter frequencies
- Since **P** and **Z** occur most frequently, guess they correspond to **E** and **T** respectively.
- Count relative digram frequencies.
- Since **ZW** occurs most frequently, guess it corresponds to **TH** (which is the digram occurring most frequently in English)
- Hence **ZWP** is **THE**
- By proceeding with trial and error finally get:

IT WAS DISCLOSED YESTERDAY THAT SEVERAL INFORMAL BUT DIRECT CONTACTS HAVE BEEN MADE WITH POLITICAL REPRESENTATIVES OF THE VIET CONG IN MOSCOW

Homophonic substitution ciphers

- To each $a \in \mathcal{A}$ associate a set $H(a)$ of strings of t symbols, where $H(a), a \in \mathcal{A}$ are pairwise disjoint. A homophonic substitution cipher replaces each a with a randomly chosen string from $H(a)$. To decrypt a string c of t symbols, one must determine an $a \in A$ such that $c \in H(a)$. The key for the cipher is the sets $H(a)$.
- **Example:** $\mathcal{A} = \{x, y\}$, $H(x) = \{00, 10\}$, and $H(y) = \{01, 11\}$. The plaintext xy encrypts to one of 0001, 0011, 1001, 1011.
- Rational: makes frequency analysis more difficult.
  Cost: data expansion and more work for decryption.

Polyalphabetic substitution ciphers

- Idea (Leon Alberti): conceal distribution using family of mappings.
- A polyalphabetic substitution cipher is a block cipher with block length t over alphabet $\mathcal{A}$ where:
  - the key space $\mathcal{K}$ consists of sequences of permutations over $\mathcal{A}$ of the form $(e_1, \ldots, e_t)$.
  - Encryption of $m = m_1 m_2 \cdots$ under key $e = (e_1, \cdots, e_t)$ is $E_e(m) = c_1 c_2 \cdots$, where $c_i = e_{(i \bmod t)}(m_i)$ for $i = 1, 2, \ldots$
  - Decryption key for e is $d = (e_1^{-1}, \cdots, e_t^{-1})$.

Example: Vigenère ciphers

- Permutations are defined in terms of a sequence of numbers $k_1, \ldots, k_t$ in the following way:

$$e_i(b) = (b + k_i) \mod |\mathcal{A}| \qquad \text{for all } b \in \mathcal{A} \text{ and } i = 1, \ldots, t$$

- Example: English ($n = 26$), with $k = k_1, k_2, k_3$ with $k_1 = 3$, $k_2 = 7$, and $k_3 = 10$:

$$\begin{aligned} m &= \texttt{THI SCI PHE RIS CER TAI NLY NOT SEC URE} \\ E_e(m) &= \texttt{WOS VJS SOO UPC FLB WHS QSI QVD VLM XYO} \end{aligned}$$

One-time pads (Vernam cipher)

- A one-time pad is a stream cipher defined on $\mathcal{A} = \{0, 1\}$. Message $m_1 \cdots m_n$ is encrypted by a *randomly chosen* binary key string $k_1 \cdots k_n$.

$$\begin{aligned} E_{k_1 \cdots k_n}(m_1 \cdots m_n) &= (m_1 \oplus k_1) \cdots (m_n \oplus k_n) \\ D_{k_1 \cdots k_n}(c_1 \cdots c_n) &= (c_1 \oplus k_1) \cdots (c_n \oplus k_n) \end{aligned}$$

- Example:

$$\begin{array}{rcl} m & = & 010111 \\ k & = & 110010 \\ \hline c & = & 100101 \end{array}$$

- Since every key sequence is equally likely, so is every plaintext! Unconditional (information theoretic) security, if key isn't reused!
- Moscow–Washington communication previously secured this way.
- Problem? Securely exchanging and synchronizing long keys.

One-time pads (Vernam cipher)

- A one-time pad is a stream cipher defined on $\mathcal{A} = \{0, 1\}$. Message $m_1 \cdots m_n$ is encrypted by a *randomly chosen* binary key string $k_1 \cdots k_n$.

$$\begin{aligned} E_{k_1 \cdots k_n}(m_1 \cdots m_n) &= (m_1 \oplus k_1) \cdots (m_n \oplus k_n) \\ D_{k_1 \cdots k_n}(c_1 \cdots c_n) &= (c_1 \oplus k_1) \cdots (c_n \oplus k_n) \end{aligned}$$

- Example:

$$\begin{array}{rcl} m & = & 010111 \\ k & = & 110010 \\ \hline c & = & 100101 \end{array}$$

- Since every key sequence is equally likely, so is every plaintext! Unconditional (information theoretic) security, if key isn't reused!
- Moscow–Washington communication previously secured this way.
- Problem? Securely exchanging and synchronizing long keys.

Stream Cipher Limitations

- Keys must not be reused! (Hence the name “one-time” pad.)
- Suppose Alice wishes to encrypt and send Bob two messages M_1 and M_2.
- Alice decides to encrypt both messages using the same stream cipher key S, i.e.

$$C_1 = S \oplus M_1 \qquad C_2 = S \oplus M_2$$

- An adversary, who intercepts C_1 and C_2, can compute

$$C_1 \oplus C_2 = (S \oplus M_1) \oplus (S \oplus M_2) = M_1 \oplus M_2$$

- Since English text contains redundancy, given $M_1 \oplus M_2$ the adversary can recover both M_1 and M_2 in the clear (for sufficiently long M_1 and M_2).

Stream Cipher Limitations

During WWII the Soviet Union could not produce enough one-time pads. . . to keep up with the enormous demand . . . So, they used a number of one-time pads twice, thinking it would not compromise their system. American counter-intelligence during WWII collected all incoming and outgoing international cables. Beginning in 1946, it began an intensive effort to break into the Soviet messages with the cooperation of the British and by . . . the Soviet error of using some one-time pads as two-time pads, was able, over the next 25 years, to break some 2900 messages, containing 5000 pages of the hundreds of thousands of messages that been sent between 1941 and 1946 (when the Soviets switched to a different system).[3]

[3] J. Haynes and H. Klehr. *Venona: Decoding Soviet Espionage in America.* Yale University Press, 1999.

Malleability of One-time Pads

- A cryptographic function $E(K, M)$ is *malleable* iff there exist two functions $F(X)$ and $G(X)$ such that

$$F(E(K, M)) = E(K, G(M)) \quad \text{for all keys } K \text{ and messages } M$$

- $E(K, M)) = K \oplus M$ is clearly malleable.
  Let $F(X) = G(X) = N \oplus X$ for any given N of appropriate size.

$$F(E(K, M)) = N \oplus (K \oplus M) = K \oplus (N \oplus M) = E(K, G(M))$$

- Corollary: You can turn the ciphertext $C_1 = K \oplus M_1$ of a given known message M_1 into the ciphertext $C_2 = K \oplus M_2$ of any M_2 of choice even if you do not know K!
  How? It suffices to compute the $\oplus$ of C_1 and $M_1 \oplus M_2$. In fact,

$$C_1 \oplus (M_1 \oplus M_2) = (K \oplus M_1) \oplus (M_1 \oplus M_2) = K \oplus M_2$$

Outline

Transposition ciphers

- For block length t, let $\mathcal{K}$ be the set of permutations on $\{1, \ldots, t\}$. For each $e \in \mathcal{K}$ and $m \in \mathcal{M}$
$$E_e(m) = m_{e(1)} m_{e(2)} \cdots m_{e(t)}$$
- The set of all such transformations is called a transposition cipher.
- To decrypt $c = c_1 c_2 \cdots c_t$ compute $D_d(c) = c_{d(1)} c_{d(2)} \cdots c_{d(t)}$, where d is inverse permutation.
- Letters unchanged so one can exploit frequency analysis for dipthongs, tripthongs, words, etc.

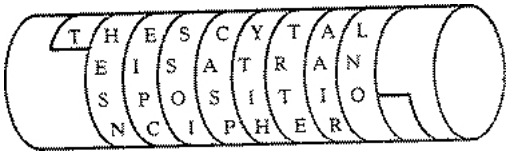
Example: transposition ciphers

- $C =$ Aduaenttlydhatoiekounletmtoihahvsekeeeleeyqonouv

A	n	d	i	n	t	h	e	e	n
d	t	h	e	l	o	v	e	y	o
u	t	a	k	e	i	s	e	q	u
a	l	t	o	t	h	e	l	o	v
e	y	o	u	m	a	k	e		

Table defines a permutation on 1, ..., 50.

- Idea goes back to Greek Scytale: wrap belt spirally around baton and write plaintext lengthwise on it.



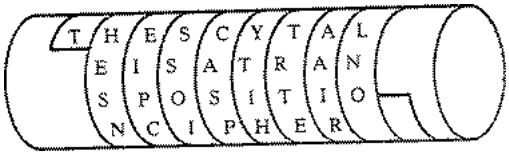
Example: transposition ciphers

- $C =$ Aduaenttlydhatoiekounletmtoihahvsekeeeleeyqonouv

A	n	d	i	n	t	h	e	e	n
d	t	h	e	l	o	v	e	y	o
u	t	a	k	e	i	s	e	q	u
a	l	t	o	t	h	e	l	o	v
e	y	o	u	m	a	k	e		

Table defines a permutation on 1, ..., 50.

- Idea goes back to Greek Scytale: wrap belt spirally around baton and write plaintext lengthwise on it.



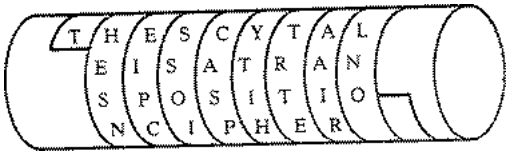
Example: transposition ciphers

- $C =$ Aduaenttlydhatoiekounletmtoihahvsekeeeleeyqonouv

A	n	d	i	n	t	h	e	e	n
d	t	h	e	l	o	v	e	y	o
u	t	a	k	e	i	s	e	q	u
a	l	t	o	t	h	e	l	o	v
e	y	o	u	m	a	k	e		

Table defines a permutation on 1, ..., 50.

- Idea goes back to Greek Scytale: wrap belt spirally around baton and write plaintext lengthwise on it.



Outline

Composite ciphers

- Ciphers based on just substitutions or transpositions are not secure
- Ciphers can be combined. However . . .
  - two substitutions are really only one more complex substitution,
  - two transpositions are really only one transposition,
  - but a substitution followed by a transposition makes a new harder cipher.
- Product ciphers chain substitution-transposition combinations.
- Difficult to do by hand $\rightsquigarrow$ invention of cipher machines.

Bibliography

- William Stallings. *Cryptography and Network Security*. Fourth Edition, Prentice Hall, 2006.
- Dieter Gollmann. *Computer Security*. Wiley, 2000.
- Bruce Schneier. *Applied Cryptography*. John Wiley & Sons, New York, 1996.
- Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.
  Available online at `http://cacr.math.uwaterloo.ca/hac/`
- Arthur E. Hutt, Seymour Bosworth, Douglas B. Hoyt. *Computer Security Handbook*. John Wiley & Sons, 1995.