Introduction to Cryptography

(203.3444/203.4444)

Dr. Orr Dunkelman

Computer Science Department
Faculty of Social Sciences
University of Haifa, Haifa 31905, Israel

October 25, 2012

Shimon Even, Dror Rawitz, Moni Shachar, Orr Dunkelman, Elad Barkan and Benny Applebaum made major contributions to these slides.

Cryptology Course

Points: 3.0

Lecturer: Orr Dunkelman

Class: Sunday 14:00–17:00, Education 408

Prerequisites: 203.2480 Probabilistic Methods

Cryptology Course (cont.)

Grade: 70% exam, 30% homework

The exercises are Magen.

Exam grades below 46 will not be combined with the grades of the exercises!

Homework: 5 exercise pages

Two-Three weeks for finishing them.

Submission: singles/pairs.

Cheating will be severely frowned upon.

WWW page: `http://www.cs.haifa.ac.il/~orrd/IntroToCrypto/`

Mailing List: Please make sure you check your email accounts, as announcements will be sent there.

Lecturer Contact Information

Lecturer: Orr Dunkelman

Office: Jacobs 408

Office Hour: Sun. 10:00–11:00

Phone: 8447

Email: orrd@cs.haifa.ac.il

WWW: http://www.cs.haifa.ac.il/∼orrd/

Topics

- Introduction to Cryptology
- Substitution Ciphers
- Shannon's Theory of Secrecy Systems
- Blockciphers
  - Data Encryption Standard (DES)
  - Advanced Encryption Standard (AES)
  - Modes of Operation
- Hashing and One-Time Signatures
- Introduction to Number Theory
- Public Key Cryptography
  - Diffie-Hellman Key Exchange
  - RSA
  - Rabin's Variant
  - Related algorithms
  - DLOG based signature schemes
- Zero-Knowledge Protocols
  - Fiat-Shamir identification scheme
- Secret Sharing

What is Cryptology

- **cryptography**: The act or art of writing in secret characters.
- **cryptanalysis**: The analysis and deciphering of secret writings.
- **cryptology**: (Webster's) the scientific study of cryptography and cryptanalysis.

In our context **cryptology** is the scientific study of protection of information.

Cryptographic Services

Cryptography supports the following services:

1. Confidentiality

2. Integrity

3. Authentication

4. Identity

5. Timeliness

6. Proof of ownership

Each has various different requirements in different circumstances, and each is supported by a wide variety of schemes.

Applications

1. Communications (encryption or authentication)
2. File and data base security
3. Electronic funds transfer
4. Electronic Commerce
5. Digital cash
6. Contract signing
7. Electronic mail
8. Authentication: Passwords, PINs
9. Secure identification, Access control
10. Secure protocols
11. Proof of knowledge

Applications (cont.)

12. Construction by collaborating parties (secret sharing)

13. Copyright protection

14. etc.

Recommended Books

Textbook:
Paar, Pelzl, *Understanding Cryptography*, Springer, 2008.
Some might prefer to take a look at:
Stinson, *Cryptography, Theory and Practice*, CRC press, 1995.
or
Stinson, *Cryptography, Theory and Practice*, second edition, Chapman Hall/CRC, 2002.
or
Stinson, *Cryptography, Theory and Practice*, third edition, Chapman Hall/CRC, 2006.[1]

[1] The second edition presents new schemes, e.g., SHA-1 and AES, but lacks various other topics presented in the first edition (secret sharing, ZK, Diffie-Hellman, etc.). The third edition adds additional material, including some of the metarial that was in the first edition. The presentation of DES in this course is closer to the presentation in the first edition, than to the newer editions.

Recommended Books (cont.)

Reference Books:

Menezes, van Oorschot, Vanstone, *Handbook of Applied Cryptography*, CRC press, 1997.

Cormen, Leiserson, Rivest, Stein, *Introduction to Algorithms*, third edition, Massachusetts Institute of Technology, 2009.

History of Cryptography

1. Steganography: Hiding information by non-cryptographic methods.

   (a) Writing with an invisible ink.

   (b) Writing in an hidden place (such as the least significant bits of the gray levels of pixels in a scanned picture).

2. An Assyrian king wrote on the head of a slave, and sent him through the enemy's lines, after the hair was grown.

History of Cryptography (cont.)

3. First cryptographic attempts: Jeremiah:
   (a) Jeremiah, 25, 26
   (b) Jeremiah, 51, 41
   (c) Jeremiah, 51, 1
4. First cryptographic attempts: Daniel: A hand written cipher for the king of Assyria, but nobody could reveal the meaning, till Daniel translated the cipher.

History of Cryptography (cont.)

5. Caesar cipher.

> “Exstant et ad Ciceronem, item ad familiares domesticis de rebus, in quibus, si qua occultius perterenda erant, per notas scripsit, id est sic structo litterarum ordine, ut nullam verbum effici posset; quae si qui investigare et persequi velit, quartem elementorum litteram, id est D pro A et perinde reliquas commutet.”
>
> “There are also letters of his [Julius Caesar’s] to Cicero, as well as to his intimates on private affairs, and in the latter, if he had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out. If anyone wishes to decipher these, and get at their meaning, he must substitute the fourth letter of the alphabet, namely D, for A, and so with the others.”

- Suetonius, ”De Vita Caesarum”, ∼ 150 A.D.

History of Cryptography (cont.)

6. 19'th century and beginning of 20'th century: The wide use of telegraph (and semaphores) made encryption necessary; transposition and substitution ciphers.
7. World war I: wide use of cryptography. Cryptanalysis (also lack of cryptanalysis) widely affected the war. The Zimmermann telegram.
8. 1930's: Enigma and other rotor machines.
9. World war II: Even wider use of cryptography and cryptanalysis.
10. Till 1970's: Usually used by governments and armies. Very limited public research and development. Used by the public primarily for quizzes.
11. 1970's: Lucifer and **DES** (by IBM).

History of Cryptography (cont.)

12. 1976: A turn point:
    (a) Merkle's puzzles.
    (b) One-time signatures.
    (c) Diffie and Hellman's **public key cryptography**.
    (d) The **RSA** cryptosystem.

13. Since then, a huge development was done in the field, including
    (a) zero-knowledge schemes,
    (b) quantum cryptography,
    (c) differential and linear cryptanalysis,
    (d) secure smartcard applications,
    (e) AES,
    (f) and many others.

History of Cryptography (cont.)

14. Since the 1990's: Widely used

    (a) Protecting cellular phone conversations and messages

    (b) Browsing the Internet: access to your bank account, secure email, browsing with https

    (c) Internet protocols: SSL, IPSEC

    (d) Wireless (802.11b/g/i, WEP), Bluetooth

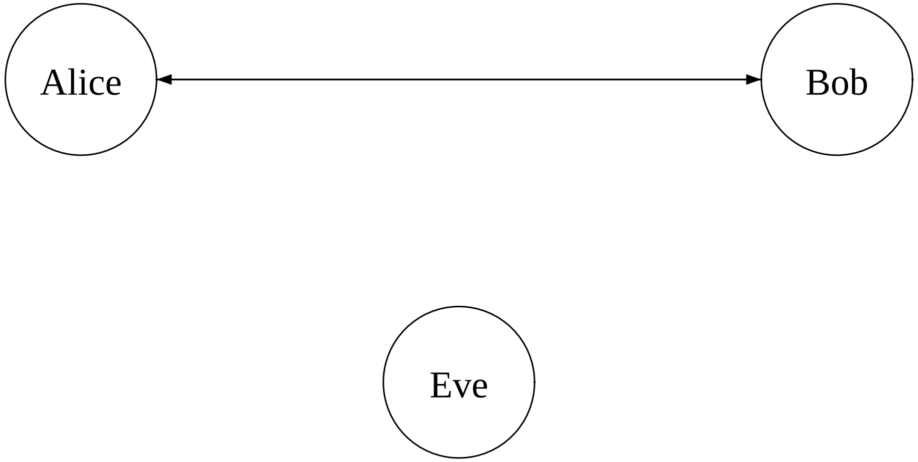
    (e) Internet applications: ssh

    (f) Other applications: disk encryption

Introduction to Cryptology

Participants

Alice and **Bob**: two parties who want to communicate securely.
Eve: an eavesdropper who wants to listen/modify their communication.



Participants (cont.)

Alice and Bob want to communicate:

- To authenticate the party they speak with.
- Eve cannot understand their messages or modify them to her advantage.

Eve wants:

- To understand or modify Alice and Bob's messages, or
- Send her own messages on their behalf.
- Eve might apply any operation that might help her.

Eve trials are called **attacks**.

Ciphers

The information (data) Alice and Bob send is called **plaintext** (or **cleartext**), and denoted by P.

The information transferred over the channel to which Eve can listen is called **ciphertext** (or **cryptogram**), and denoted by C.

The algorithm that transforms the plaintext to the ciphertext (and back) is called a **cipher** or a **cryptosystem**. The transformations of the cipher are called **encryption** and **decryption**.

Kerckhoff's Principle

We do not wish to rely only on the obscurity of the cipher being used: our communication should remain secure even if Eve knows the cipher, or found a way to steal its definition.

Therefore, in all the analysis, we assume that Eve knows the details of the cipher. The cipher has to be secure even in this case.

The only secret is assumed to be the **key** (denoted by K) which selects the exact transformation of the cipher.

Therefore, a cipher can viewed as a set of many (unkeyed) transformations which have similar structures (e.g., source code) but different in many details, and the key selects the particular instance of the transformation.

Requirements From Ciphers

1. For the legitimate users: Easy to encrypt/decrypt when the key is known.

2. For an attacker: Difficult to

   (a) encrypt/decrypt when the key is unknown,

   (b) recover the key,

   (c) get any information on the encrypted text,

   even if a lot of encrypted samples are given.

3. The above hold even if the algorithm is publicly known.

Requirements From Ciphers (cont.)

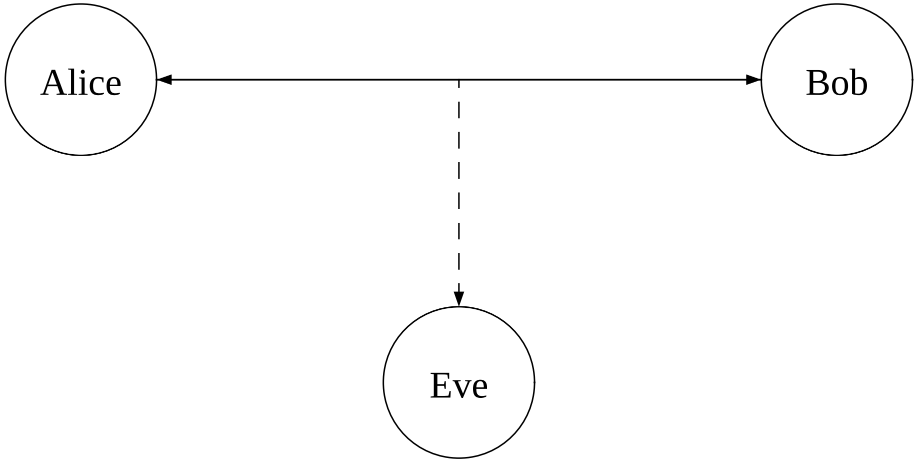
Cryptography relies on **one-way functions**, which are publicly known and easy to compute, but difficult to invert.

In particular, ciphers are designed to be easy to encrypt and decrypt when the key is known, but to be one-way when the key is the unknown input.

Passive and Active Eavesdropping

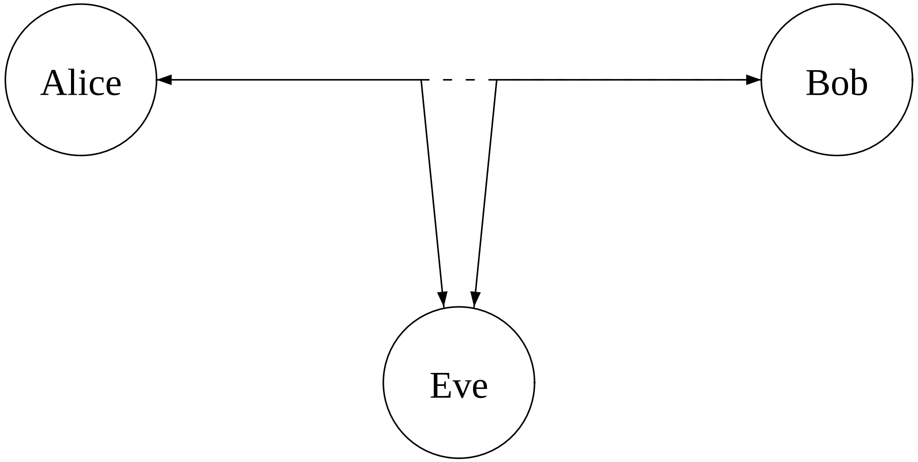
Attackers can try to get the information they need in various ways.

Passive eavesdropping: The attacker can only listen to the communication:



Passive and Active Eavesdropping (cont.)

Active eavesdropping: The attacker can **modify** the communication:



Types of Cryptanalytic Attacks

Such abilities of the attackers affect the types of attacks they can mount:

Ciphertext only attack Requires only the ciphertext, and assumes knowledge of some statistics on the plaintext (such as it is an English text). Finds either the key or the plaintext.

Known plaintext attack Finds the key using the knowledge of both **the plaintext and the ciphertext**.

Exhaustive search attack are a simple example of known plaintext attacks, applicable (in theory) to any cipher. They encrypt a plaintext under all the possible keys, and compare the results to the expected ciphertext. When the key space is too large, exhaustive search becomes infeasible.

Types of Cryptanalytic Attacks (cont.)

Chosen plaintext attack The attacker not only knows the plaintext, she can choose it to her advantage and receive the corresponding ciphertext.

Adaptive chosen plaintext attack A chosen plaintext attack in which the attacker can choose the next plaintext block depending on the ciphertext received for the previous blocks.

Chosen key attack, etc... Other more powerful, but less practical types of attacks.

As we proceed in the attacks above, the attacker receives more information, and thus can more easily find the key. However, it becomes less practical to receive the required information.

The Secret Key

We always assume that the cipher is known to the attacker, and that the security depends only on the secrecy of the key.

Each time we encrypt, the secret key is selected uniformly at random to ensure that nobody else knows it.

The keys should be selected from a large set of possible keys in order to decrease the probability of guessing the secret key, and to increase the time required for an attacker to try all keys in the set (i.e., to increase the complexity of exhaustive search).

Used Key Sizes

- 40 bits ($2^{40} = 10^{12}$ possible keys): is very common in old Internet applications due to (obsolete) export controls from the US. Totally insecure.

- 56 bits ($2^{56} = 7 \cdot 10^{16}$ possible keys): DES. Good enough in the 1970's, but insecure today.
- 64-bit keys: better, but new applications better have larger keys.
- 80-bit keys: Used in Clipper (Skipjack).
- 128-bit keys: The new standard for symmetric encryption.
- The AES (successor of DES) supports key sizes of 128, 192, and 256 bits.

Difficulty of Cryptanalysis

- Cryptanalysis is the techniques used to recover (or forge) the secret information (or a fraction of the secret information) hidden by the cryptographic algorithms.

- We usually assume that the goal of cryptanalysis is finding the secret key (although in some cases it is possible to find the plaintext but not the key).

- Theoretically, the information on the key is included even in a relatively short ciphertext, as the attacker can always perform exhaustive search to find it. However, this method might be very slow.

- The cryptanalyst may develop attacks that require long ciphertexts to reduce the time required for cryptanalysis.

- However, the main goal of ciphers are to inhibit cryptanalysis, so the cryptanalyst's job should be very difficult, if the ciphers are well developed.

Difficulty of Cryptanalysis (cont.)

Unfortunately, there are many insecure ciphers used in the industry.

Moreover, using good ciphers is not the whole solution: the developer of a system should understand how the ciphers should be used, and what are the limitations of ciphers.

For example, there are commercial applications that provide encryption:

- Some use unpublished proprietary algorithms: many of those are very weak, and can be broken instantly. In many cases, the algorithms are so simple that they can be recognized by looking at the encrypted file, and the cryptanalysis can be done without any complex computation.

Difficulty of Cryptanalysis (cont.)

- Some use standard secure ciphers, but in order to protect the user during decryption, they store a copy of the key in the beginning of the encrypted file, and they compare the copy of the key to the key the user supplies, giving an error message if they are different. Of course, just by looking in the file the key can be identified.

- Many other errors in using ciphers appear in real systems.

Therefore, in cryptography it is not sufficient to use secure algorithms. The whole system should be designed with security in mind.

Cryptographic Assumptions

The cryptographic security can rely on either

1. **Complexity theory**: The cryptographic problem may be solvable, but it takes a very long time to solve (e.g., millions of years) — the cryptosystem is **computationally secure**

2. **Information theory**: The cryptographic problem cannot be solved without additional information (even in unlimited time and space) — the cryptosystem is **unconditionally secure**

Substitution Ciphers and One-Time Pad

Caesar's Cipher

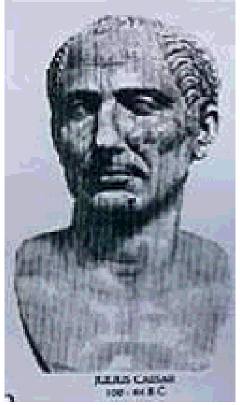
The first known algorithmic encryption.

Julius Caesar encrypted his messages by substituting each letter in the text by the third letter thereafter (cyclically):

$$\begin{array}{ccccccc} \mathrm{a} & \rightarrow & \mathrm{D} & & \mathrm{w} & \rightarrow & \mathrm{Z} \\ \mathrm{b} & \rightarrow & \mathrm{E} & \ldots & \mathrm{x} & \rightarrow & \mathrm{A} \\ \mathrm{c} & \rightarrow & \mathrm{F} & & \mathrm{y} & \rightarrow & \mathrm{B} \\ \mathrm{d} & \rightarrow & \mathrm{G} & & \mathrm{z} & \rightarrow & \mathrm{C} \end{array}$$

(notation: capital letters are used to denote ciphertext)

Thus, **caesar** is encrypted to **FDHVDU**.



Caesar's Cipher (cont.)

Weakness: Everyone who knows the encryption scheme can decrypt any message.

Caesar's Cipher (cont.)

When Augustus came to power the imperial cipher was changed to a shift of two letters.

Define a **key** known only to the sender and the receiver. The key is used as an additional input to the encryption/decryption functions $C = E_K(P)$, $P = D_K(C)$.

In Caesar's cipher $0 \leq K \leq 25$ can denote the shift of the letters (rather than $K = 3$ always).

This example is still weak, since the key space is too small.

Transposition Ciphers

Transposition ciphers are ciphers in which the order of the letters is permuted by some rule (which depends on a key).

Such ciphers were used extensively at the 19'th century and the beginning of the 20'th century.

Monoalphabetic Substitution Ciphers

Caesar's cipher have a set of 26 possible keys, which can be easily guessed and verified by attackers. The problem of Caesar's cipher is the small set of keys, and the simple permutations (cyclic rotation of letters) they use.

A major improvement is the replacement of the simple permutation by a random permutation, such that any permutation of the 26 letters is possible. The number of such permutations is enormous ($26! = 4 \cdot 10^{26}$).

Such ciphers are called **(Monoalphabetic) Substitution Ciphers**.

The key is a permutation. The cipher substitutes any letter by the corresponding letter given by the permutation. Decryption is performed similarly using the inverse permutation.

Monoalphabetic Substitution Ciphers (cont.)

Example: The key is the permutation:

abcdefghijklmnopqrstuvwxyz
PDUIRMFOHSBNCGVKTJWEYAQXZL

Encryption:

Plaintext:	monoalphabeticsubstitution
Ciphertext:	CVGVPNKOPDREHUWYDWEHEYEHVG

Decryption:

Ciphertext:	CVGVPNKOPDREHUWYDWEHEYEHVG
Plaintext:	monoalphabeticsubstitution

Security

The number of possible keys is $26! = 4 \cdot 10^{26} = 1.3 \cdot 2^{88}$. Therefore, the key can be represented with 89 bits.

Clearly, it is impractical to search all the key space exhaustively, and the probability of guessing the key is very low.

Therefore, it seems that this cipher is secure.

Are there some **algorithmic shortcuts** that can help the attacker?

A Simple Ciphertext-Only Attack

Clearly, this kind of ciphers cannot protect against known plaintext and chosen plaintext attacks. Therefore, we restrict our discussion to ciphertext-only attacks, and try to prove that even in such environments they are insecure.

However, there are algorithmic shortcuts that help the attacker using additional information.

Monoalphabetic substitution ciphers are vulnerable to ciphertext only attacks if the ciphertext and the distribution of the plaintext letters (i.e., in an English text) are known to the attacker.

The main observation is that the distribution of the letters is invariant to the permutation, and that each letter is permuted to another which get the same frequency as the original letter in the original text.

A Simple Ciphertext-Only Attack (cont.)

For example, the most frequent letter in an English text is **e**:

Letter	Frequency	Letter	Frequency	Letter	Frequency
e	12.31%	l	4.03%	b	1.62%
t	9.59%	d	3.65%	g	1.61%
a	8.05%	c	3.20%	v	0.93%
o	7.94%	u	3.10%	k	0.52%
n	7.19%	p	2.29%	q	0.20%
i	7.18%	f	2.28%	x	0.20%
s	6.59%	m	2.25%	j	0.10%
r	6.03%	w	2.03%	z	0.09%
h	5.14%	y	1.88%		

A Simple Ciphertext-Only Attack (cont.)

The most frequent English word is **the**:

Word	Frequency	Word	Frequency	Word	Frequency
the	6.421%	a	2.092%	i	0.945%
of	4.028%	in	1.778%	it	0.930%
and	3.150%	that	1.244%	for	0.770%
to	2.367%	is	1.034%	as	0.764%

Breaking Monoalphabetic Substitutions

Exercise: Solve

```
UCZCS NYEST MVKBO RTOVK
VRVKC ZOSJM UCJMO MBRJM
VESZB SMOSJ OBKYE MJTRV
VEMPY JMOMJ AMVEM HKOVJ
KTRVK CZCQV EMNMV VMJOS
ZHVER OVEMP BSZTM MSOKN
PTJCI MZ
```

The frequency of the letters in this ciphertext:

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M
Occurs	1	5	7	0	8	0	0	2	1	10	8	0	19
Letter	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Occurs	3	11	3	1	6	9	6	2	15	0	0	3	7

Vigenere Cipher

Uses Caesar's cipher with various different shifts, in order to hide the distribution of the letters. The key defines the shift used in each letter in the text.

A key word is repeated as many times as required to become the same length as the plaintext. The result is added to the plaintext as follows:

Plaintext:	`vigenerescipher`
Key:	`keykeykeykeykey`
Ciphertext:	`FMEORCBIQMMNRIP`

(a=0, b=1, . . . , z=25, mod 26).

This cipher was considered very secure in the 19'th century, and was still used in the first world war...

Vigenere Cipher (cont.)

But in 1863 Kasiski found a method to break it:

1. Find the keyword length:

   (a) If short, try 1, 2, 3, . . . , or

   (b) Find repeated strings in the ciphertext. Their distance is expected to be a multiple of the length. Compute the gcd of (most) distances.

2. Find the key letters one by one (just as in Caesar's cipher).

Exercise: Solve

```
KBPYU AXTGV HANWV IQNTT GZRUC ZFCXM
GGPOB LVPXQ GMHLG MAGNT LUJMG DCBAC
TEHJH EHIGC IDTWF FKAWY AAKCU BLATI
MAJMW FKULC NNRYY VXBCV WQRZX YFWNR
JQLNC ELMPM TDVMK RJMHR JMKRQ OXRTQ
WMHBA CUTXC RGYCG TBLIQ GRJMB PVQIQ
```

Distances between the triplets: 4, 8, 32, 36.

Vigenere Cipher (cont.)

We now repartition the text into quartets. This way, in each quartet, the first letter is encrypted by the same key, the second letter is encrypted by the same (but not necessarily the same as the first) key, etc.

```
KBPY
UAXT
GVHA
NWVI
QNTT
GZRU
CZFC
X...
```

Vigenere Cipher (cont.)

The frequency of the letters in this ciphertext in the first column:

Letter	A	C	F	G	H	I	J	K	N	P	Q	R	T	U	V	X	Y
Occurs	1	4	1	6	2	1	6	3	3	2	2	2	4	3	3	1	1

The frequency of the letters in this ciphertext in the second column:

Letter	A	B	C	E	G	I	M	N	O	P	Q	T	U	V	W	Z
Occurs	3	4	1	3	1	1	10	2	2	1	4	3	1	3	3	3

We can guess that in the second column $e \rightarrow M$ (i.e., the second letter of the Vigenere key is I).
In the first column, we might suspect that $e \rightarrow G/J$ (i.e., the first letter of the Vigenere key is either C or F).

Vigenere Cipher (cont.)

Problem:
Given two finite distribution vectors V and U of n elements how can we determine the shift of U with respect to V? Given the distribution vector of the plaintext (V), and the distribution vector of the letter in some column (U), how we can determine the shift of U with respect to V (i.e., the key)?
Solution:
We score each shift of the vectors. The *distance* between V and U can be measured by $d(V,U) = \Sigma_{i=0}^{n-1}(V_i - U_i)^2 = \Sigma V_i^2 - 2\Sigma V_i U_i + \Sigma U_i^2$, where V_i is the probability of the element i in the probability vector V. We would like a smaller distance to have a larger score, so we take the minus of the distance. Moreover, as we are interested in the relative scores of the shifts, we can eliminate the constants factors.
We define the *mutual index of coincidence* of V and U as

$$MI_c(V,U) = \sum_{i=0}^{n-1} V_i \cdot U_i$$

Vigenere Cipher (cont.)

We compute the MI_c between V and various shifts of U, and take into consideration only those with high enough MI_c value.

The difference between guessing which encrypted letter is e and checking MI_c, is that guessing e is equivalent to matching the peaks of the probability vectors, while using MI_c use more information (other entries) to find the true value.

Vigenere Cipher (cont.)

Back to our example, we define V^j to be the probability vector of the first column shifted to the left by j places, and compute the various $MI_c(V_{plaintext}, V^j)$. Where $V_{plaintext}$ is the expected probability vector related to the plaintext.

Shift	0	1	2	3	4	5	6
$MI_c(V_{plaintext}, V^j)$	0.032	0.041	0.066	0.035	0.030	0.043	0.049
Shift	7	8	9	10	11	12	13
$MI_c(V_{plaintext}, V^j)$	0.031	0.031	0.045	0.027	0.028	0.031	0.047
Shift	14	15	16	17	18	19	20
$MI_c(V_{plaintext}, V^j)$	0.036	0.048	0.041	0.046	0.037	0.037	0.035
shift	21	22	23	24	25		
$MI_c(V_{plaintext}, V^j)$	0.041	0.035	0.028	0.044	0.034		

Thus, we conclude that the first letter of the key is c.

Vigenere Cipher (cont.)

The described approach does not take into consideration biased texts, e.g., texts with “too many” z’s or only a few e’s.

Another possible approach is to compute the MI_c between pairs of columns (various positions in the key), and get the difference of their shifts. Then we can normalize the text by adding the differences of the shifts, and get a ciphertext whose all columns are shifted by the same number of letters, i.e., encrypted by a simple Caesar cipher. Finally, we solve the Caesar cipher (either by exhaustive search or using MI_c) to get the plaintext.

Finding the Key Length using the Index of Coincidence

Write the ciphertext, and below it write the ciphertext shifted by j locations; count the number of characters that are identical in the same location in both lines. The peaks are expected to be when j is a multiplication of the key length.

```
KBPYUAXTGVHANWVIQNTTGZRUCZFCXMGGPOBLVPXQGMHLGMAGNT...
....KBPYUAXTGVHANWVIQNTTGZRUCZFCXMGGPOBLVPXQGMHLGMAGNT...
....000000000000000000000100000000000000000001100000...
```

For example in the above text:

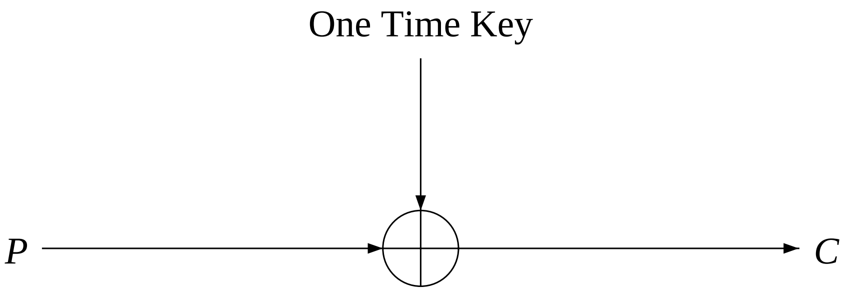
Shift	0	1	2	3	4	5	6
Index of Coincidence	180	6	4	7	18	6	5
Shift	7	8	9	10	11	12	13
Index of Coincidence	8	12	4	13	7	11	6
Shift	14	15	16	17	18	19	20
Index of Coincidence	9	2	10	7	5	4	7

Vernam Cipher - One Time Pad

A Vigenere cipher in which each key has the same length as the plaintext, and each key is uniformly selected at random and used only for one plaintext.

The attack described on the Vigenere cipher is not applicable to Vernam (why?).

One-Time Pad Over Binary Alphabets



One-Time Pad Over Binary Alphabets (cont.)

Example: Encrypting binary data using an one-time pad:

Plaintext:	`o`	`n`	`e`	`t`	`i`
In binary:	`01101111`	`01101110`	`01100101`	`01110100`	`01101001`
Key:	`01011100`	`01010001`	`11100000`	`01101001`	`01111010`
Ciphertext:	`00110011`	`00111111`	`10000101`	`00011101`	`00010011`
Plaintext:	`m`	`e`	`p`	`a`	`d`
In binary:	`01101101`	`01100101`	`01110000`	`01100001`	`01100100`
Key:	`11111001`	`11000110`	`01011010`	`10110001`	`01110011`
Ciphertext:	`10010100`	`10100011`	`00101010`	`11010001`	`00010111`

The key is randomly chosen, and is used for encryption of only one message. All the key bits are independent, and thus the ciphertext becomes random. The same ciphertext can be the encryption of any plaintext, thus an eavesdropper cannot even try to identify the correct plaintext!