OLIVER KNILL

Lecture 11: Cryptography

11.1. **Cryptology** is the science of constructing and breaking codes. It consist of **cryptography**, the creation of codes and **cryptanalysis**, the theory of cracking codes. Related in information theory is the construction of **error correcting codes**. The purpose of the later is the building of protocols allowing the transmission of information to be more secure. The goal is data corruption or data loss can be reversed by adding redundant information. Already the DNA, encoding the blueprints of life have redundancy built in.

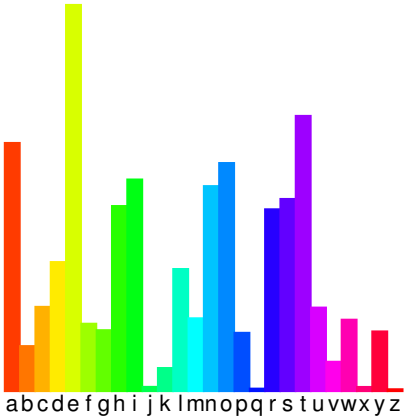
FIGURE 1. Some gadgets related to cryptology.

11.2. What kind of mathematics is involved? The theory has ties with **probability theory**. Especially in the code breaking part **statistical methods** are useful. Many codes are based on **number theory** like RSA and Diffie-Hellman. **Combinatorial** considerations come into play for example, when looking at the complexity of codes. Especially in code breaking like with plain text attacks. **Algebraic geometry** has entered through examples like **elliptic curve cryptosystems**. In general, **algebra** enters if algebraic objects like number fields are used. New branches like **quantum cryptology** use **analysis** like **Fourier theory**.

11.3. The **Caesar cipher** permutes the letters of the alphabet. We can for example replace every letter A with B, every letter B with C and so on until finally Z is replaced with A. The word "mathematics" becomes so encrypted as "nbuifnbujdt". Caesar would shift the letters by 3. The right shift just discussed was used by his Nephew Augustus. **Rot13** shifts by 13, and **Atbash cipher** reflects the alphabet, switch A with Z, B with Y etc. The last two examples are involutions: encryption is decryption. Here are examples:

Cesar:	shift three to the left	F becomes C for example
Augustus:	shift to the right	F becomes G.
Atbash:	reflect	B becomes Y and Y becomes B.
Rot13:	move to middle	A Becomes N and N becomes A.

11.4. More general ciphers are obtained by permuting the alphabet. Because of the $26! = 403291461126605635584000000 \sim 10^{27}$ permutations, it appears first that a brute force attack is not possible. But Cesar ciphers can be cracked very quickly using **statistical analysis**. If we know the frequency with which letters appear and match the frequency of a text we can figure out which letter was replaced with which.



11.5. The **Trithemius cipher** prevents this simple analysis by changing the permutation in each step. It is called a polyalphabetic substitution cipher. Instead of a simple permutation, there are many permutations. After transcoding a letter, we also change the key. Lets take a simple example. Rotate for the first letter the alphabet by 1, for the second letter, the alphabet by 2, for the third letter, the alphabet by 3 etc. The word "Mathematics" becomes now "Ncwljshbrmd". Note that the second "a" has been translated to something different than a. A frequency analysis is now more difficult. The **Viginaire cipher** adds even more complexity: instead of shifting the alphabet by 1, we can take a key like "BCNZ", then shift the first letter by 1, the second letter by 3 the third letter by 13, the fourth letter by 25 the shift the 5th letter by 1 again. While this cipher remained unbroken for long, a more sophisticated frequency analysis which involves first finding the length of the key makes the cipher breakable. With the emergence of computers, even more sophisticated versions like the German **enigma** had no chance.

Alberti	Random change of alphabet indicating switch
Trithemius	Deterministic change of alphabet
Viginere	Using key telling which alphabet to use
Enigma	Using key and deterministic alphabet change overlapped with Cesar
Hill Cipher	Use matrices to permute

11.6. Block ciphers cut text into larger chunks and scramble them. Examples are

DES	Data Encryption Standards 1973
Triple DES	Used for some electronic payments, 1998

11.7. Public key systems are based number theoretical mathematical principles like the problem of factoring integers. This has lots of relations with number theory, computer science as well as seemingly unrelated topics like algebraic geometry.

11.8. Diffie-Hellman key exchange allows Ana and Bob want to agree on a secret key over a public channel. The two palindromic friends agree on a prime number p and a base a. This information can be exchanged publicly. Ana chooses now a secret number x and sends $X = a^x$ modulo p to Bob over the channel. Bob chooses a secret number y and sends $Y = a^y$ modulo p to Ana. Ana can compute Y^x and Bob can compute X^y but both are equal to a^{xy}. This number

is their common secret. The key point is that eves dropper Eve, can not compute this number. The only information available to Eve are X and Y, as well as the base a and p. Eve knows that $X = a^x$ but can not determine x. The key difficulty in this code is the **discrete log problem**: getting x from a^x modulo p is believed to be difficult for large p.

11.9. The **Rivest-Shamir-Adleman public key system** uses a **RSA public key** (n, a) with an integer $n = pq$ and $a < (p-1)(q-1)$, where p, q are prime. Also here, n and a are public. Only the factorization of n is kept secret. Ana publishes this pair. Bob who wants to email Ana a message x, sends her $y = x^a \bmod n$. Ana, who has computed b with $ab = 1 \bmod (p-1)(q-1)$ can read the secrete email y because $y^b = x^{ab} = x^{(p-1)(q-1)} = x \bmod n$. But Eve, has no chance because the only thing Eve knows is y and (n, a). It is believed that without the **factorization** of n, it is not possible to determine x. The message has been transmitted securely.

11.10. The core difficulty is that **taking roots** in the ring $Z_n = \{0, \ldots, n-1\}$ is difficult without knowing the factorization of n. With a factorization, we can quickly take arbitrary roots. If we can take square roots, then we can also factor: assume we have a product $n = pq$ and we know how to take square roots of 1. If x solves $x^2 = 1 \bmod n$ and x is different from 1, then $x^2 - 1 = (x-1)(x+1)$ is zero modulo n. This means that p divides $(x-1)$ or $(x+1)$. To find a factor, we can take the greatest common divisor of $n, x-1$. Take $n = 77$ for example. We are given the root 34 of 1. ($34^2 = 1156$ has reminder 1 when divided by 34). The greatest common divisor of $34 - 1$ and 77 is 11 is a factor of 77. Similarly, the greatest common divisor of $34 + 1$ and 77 is 7 divides 77. Finding roots modulo a composite number and factoring the number is equally difficult.

Cipher	Used for	Difficulty	Attack
Cesar	transmitting messages	many permutations	Statistics
Viginere	transmitting messages	many permutations	Statistics
Enigma	transmitting messages	no frequency analysis	Plain text
Diffie-Helleman	agreeing on secret key	discrete log mod p	Unsafe primes
RSA	electronic commerce	factoring integers	Factoring

11.11. The simplest **error correcting scheme** just uses 3 copies of the same information. A single error can be corrected. With 3 watches for example, you know the time, even if one of the watches fails. Cockpits of airplanes have three copies important instruments. But this basic error correcting code is not efficient. It can correct single errors by tripling the size. Its efficiency is only 33 percent. A cheap way to make it more efficient is to compress the data first and then make three copies. **Data compression** is a topic by itself. Here is a simple example, the **dictionary compression**. Take dictionary with $65'536 = 2^{16}$ words for example. Every word can be encoded by two bytes. Assuming an average word length of 6, we can encode every word with 2 bytes instead of 6. There are better error correcting codes using linear algebra or algebraic geometry.

Work problems

1) We crack the Caesar cypher using statistical analysis:

Letter	Percentage	Letter	Percentage
E	11.16	M	3.01
A	8.50	H	3.00
R	7.58	G	2.47
I	7.54	B	2.07
O	7.16	F	1.81
T	6.95	Y	1.78
N	6.65	W	1.29
S	5.74	K	1.10
L	5.49	V	1.01
C	4.54	X	0.29
U	3.63	Z	0.27
D	3.38	J	0.20
P	3.17	Q	0.20

The frequency of letters is relevant for designing keyboards. The Qwerty keyboard for example has ESER and OI in prominent places.
The 'top twelve' letters help with about 80 percent of the text. You can remember the first 8 with the memonic

"A SIN TO ERR".

An other thing to look for: The **top pairs** which appear are

TH HE AN RE ER IN ON AT ND ST ES EN OF TE ED OR TI HI AS TO

The most frequent **double letters** are

"LL EE SS OO TT FF RR NN PP CC"

Example

We aim to decrypt the following text:
xf uif qfpqmf pg uif vojufe tubuft,
jo psefs up gpsn b npsf qfsgfdu vojpo,
ftubcmjti kvtujdf, jotvsf epnftujd usborvjmjuz,
qspwjef gps uif dpnnpo efgfodf,
qspnpuf uif hfofsbm xfmgbsf,
boe tfdvsf uif cmfttjoht pg mjcfsuz
up pvstfmwft boe pvs qptufsjuz,
ep psebjo boe ftubcmjti uijt dpotujuvujpo gps uif
vojufe tubuft pg bnfsjdb
Decoding:
Count the number of letters which occur. Since we have not much time, the 8 most frequent letters are listed in this text. Can you figure out the text?

f	appears	39 times
u	appears	29 times
p	appears	25 times
t	appears	20 times
s	appears	20 times
j	appears	20 times
o	appears	17 times
b	appears	14 times

2) Decrypt the following text. It belongs to a famous novel.

"ny nx f ywzym zsnajwxfqqd fhpstbqjilji, ymfy f xnslqj rfs ns utxxjxxnts tk f ltti ktwyzsj, rzxy gj ns bfsy tk f bnkj. mtbjajw qnyyqj pstbs ymj kjjqnslx tw anjbx tk xzhm f rfs rfd gj ts mnx knwxy

jsyjwnsl f sjnlmgtzwmtti, ymnx ywzym nx xt bjqq kncji ns ymj rnsix tk ymj xzwwtzsinsl kfrnqnjx, ymfy mj nx htsxnijwji ymj wnlmykzq uwtujwyd tk xtrj tsj tw tymjw tk ymjnw ifzlmyjwx.:"

3) We have seen how to encrypt messages using the Vigenère Cipher. This encryption was used for a long time and should be seen as an important marker in the development of substitution ciphers:

Julius Caesar	-70
Ahmad al-Qalqashandi	1400
Leon Battista Alberti	1467
Johannes Trithemius	1508
Blaise de Viginère	1586
Charles Babbage	1854
Friedrich Kasiski	1863
Arthur Scherbius	1920

Blaise de Vigenère was not really the inventor of the cypher.

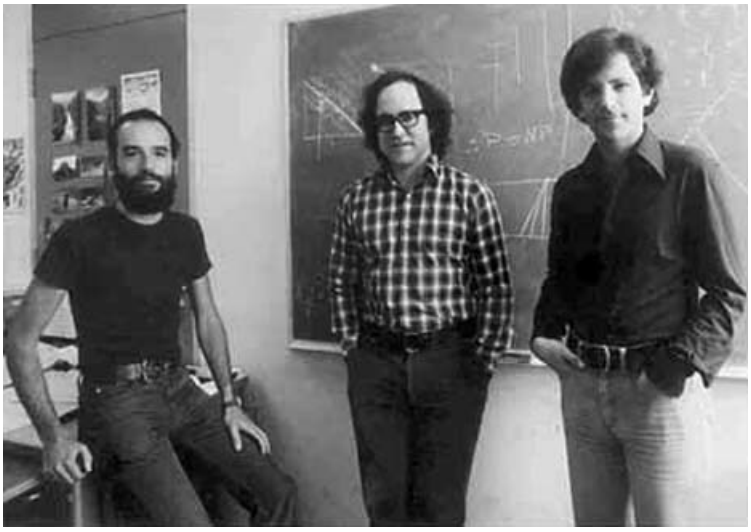
Assume we have a secret key like "ENIGMA". Given a text like "HARVARD IS COOL", we encrypt it using the following table: for the first letter, we use the line starting with E, for the second letter, we use the line starting with N etc.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Now it is your turn

HARVARD IS COOL

3) We want to understand the basic mechanism for RSA encryption.



Ron Rivest, Adi Shamir and Len Adleman.

An **RSA public key** is a pair (n, a) where n is an integer with secret factorization $n = pq$ and where $a < (p-1)(q-1)$ is such that there exists b with $ab = 1 \bmod (p-1)(q-1)$. Ana publishes this pair. If Bob wants to send a secrete message to Ana, he transmits to Ana the message

$$y = x^a \bmod n \ .$$

Ana can read the email by computing

$$y^b \bmod n \ .$$

Why does it work? We use the **Fermat's little theorem** which tells that $x^{p-1} - 1$ is is divisible by p and $x^{q-1} - 1$ is divisible by q. But this assumes p, q to be prime. For $n = pq$, we have $x^{(p-1)(q-1)} - 1$ divisible by pq.
For example take $p = 3$ and $q = 5$ Verify that $2^{(p-1)(q-1)} - 1$ is divisible by $n = pq$.

11.12. Because $y^b = x^{ab} = x \bmod n$, because $ab-1$ is divisible by $(p-1)(q-1)$ and $x^{(p-1)(q-1))} = 1 \bmod n$ we have $x^{ab} = x \bmod n$.
Ana gets the message, Bob has sent. But Eve has no chance to read it, because the only thing Eve can see is y and (n, a). It is believed that without the factorization of n, the message can not be read.

Let $(55, 13)$ be the public key of Ana. Assume Ana has the message $x = 4$ to submit. She computes $y = 4^{13} \bmod 55 = 9$. Because $55 = 11 * 5 = pq$, Ana knows $(p-1)(q-1) = 40$ and can obtain $b = 37$. With $x = 9^{37} \bmod 55$ she gets back 4.

11.13. Problem a) Assume the public key of Ana is $(n, a) = (15, 2)$. You are Bob. Send the message $x = 7$ to Ana.
Problem b) You are now Ana and have received the message from Bob. Decipher it.

Oliver Knill, knill@math.harvard.edu Math E-320, Harvard Extension School, 2010-2022