The CRYPTOGRAPHY

1. **History of cryptography**

The history of cryptography dates back thousands of years. Until recent decades, it has been a history of classic cryptography — of methods of encryption that use pen and paper, or perhaps simple mechanical aids. In the early 20th century, the invention of complex mechanical and electromechanical machines, such as the Enigma rotor machine, provided more sophisticated and efficient means of encryption; and the subsequent introduction of electronics and computing has allowed elaborate schemes of still greater complexity.

The evolution of cryptography has been paralleled by the evolution of cryptanalysis — of the "breaking" of codes and ciphers. The discovery and application, early on, of frequency analysis to the reading of encrypted communications has on occasion altered the course of history. Thus the Zimmermann Telegram triggered the United States' entry into World War I; and Allied reading of Nazi Germany's ciphers may have shortened World War II by as much as two years.

Until the 1970s, secure cryptography was largely the preserve of governments. Two events have since brought it squarely into the public domain: the creation of a public encryption standard (DES); and the invention of **public-key cryptography**.

### Classical cryptography

The earliest known use of cryptography is found in non-standard hieroglyphs carved into monuments from Egypt's Old Kingdom (ca 4500+ years ago). These are not thought to be serious attempts at secret communications, however, but rather to have been attempts at mystery, intrigue, or even amusement for literate onlookers. These are examples of still other uses of cryptography, or of something that looks (impressively if misleadingly) like it. Later, Hebrew scholars made use of simple monoalphabetic substitution ciphers (such as the **Atbash cipher**) beginning perhaps around 500 to 600 BCE.

**Atbash cipher:**

<table>
<thead>
<tr>
<th><strong>plaintex</strong></th>
<th>a b c d e f g h i j k l m</th>
</tr>
</thead>
<tbody>
<tr>
<td>ciphertext</td>
<td>Z Y X W V U T S R Q P O N</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>plaintex</strong></th>
<th>n o p q r s t u v w x y z</th>
</tr>
</thead>
<tbody>
<tr>
<td>ciphertext</td>
<td>M L K J I H G F E D C B A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>plaintex</strong></th>
<th>Il sole brilla</th>
</tr>
</thead>
<tbody>
<tr>
<td>ciphertext</td>
<td>Rohlovyirooz</td>
</tr>
</tbody>
</table>

The Greeks of Classical times are said to have known of ciphers (e.g., the scytale transposition cipher claimed to have been used by the Spartan military). Herodotus tells us of secret messages physically concealed beneath wax on wooden tablets or as a tattoo on
a slave's head concealed by regrown hair, though these are not properly examples of cryptography per se as the message, once known, is directly readable; this is known as steganography. The Romans certainly did know something of cryptography (e.g., the Caesar cipher and its variations). There is ancient mention of a book about Roman military cryptography (especially Julius Caesar's); it has been, unfortunately, lost.

**Caesar cipher:**

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>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</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cipher text</td>
<td>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</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>auguridibuoncompleanno</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cipher text</td>
<td>dxjxulglexrqfrpsohdqqr</td>
</tr>
</tbody>
</table>

In India, cryptography was also well known. It is recommended in the Kama Sutra as a technique by which lovers can communicate without being discovered.

**Medieval cryptography**

It was probably religiously motivated textual analysis of the Qur'an which led to the invention of the frequency analysis technique for breaking monoalphabetic substitution ciphers sometime around 1000 CE. It was the most fundamental cryptanalytic advance until WWII. Essentially all ciphers remained vulnerable to this cryptanalytic technique until the invention of the polyalphabetic cipher by Alberti (ca 1465), and many remained so thereafter.

Cryptography became (secretly) still more important as a consequence of political competition and religious revolution. For instance, in Europe during and after the Renaissance, citizens of the various Italian states, the Papal States and the Roman Catholic Church included, were responsible for rapid proliferation of cryptographic techniques, few of which reflect understanding (or even knowledge) of Alberti's advance. 'Advanced ciphers', even after Alberti, weren't as advanced as their inventors / developers / users claimed (and probably even themselves believed); this over-optimism may be inherent in cryptography for it was then, and remains today, fundamentally difficult to really know how vulnerable your system actually is. In the absence of knowledge, guesses and hopes, as may be expected, are common.

One of the most important coding technique is the Vigenère Code. Blaise de Vigenère published in 1586 a treaty in which he proposed a code that had great fortune and it is remembered with his name. The "Verman Code", considered the theoretically perfect code derives from this cipher.
The method can be considered a generalization of the Caesar code; instead of moving the letter to be coded always of the same number of places, this letter is moved of a number of variable place, determined on the basis of a keyword, to be fixed between the sender and the receiver, and to be written under the message, character by character.

The word is called “worm” (verme) because, usually a lot shorter than the message, it must be repeated several times under this, as in the following example:

| Plaintext: | A R R I V A N O I R I N F O R Z I |
| Worm:     | V E R M E V E R M E V E R M E V E |
| Cipher text: | V V I U Z V R F U V D R W A V U M |

You can obtain the coded text, changing the clear letter of a fixed number of characters equal to the ordinal number of the letter corresponding to the worm. In fact an arithmetic addiction takes place between the ordinal of the clear text and (A=1, B=1, C=2, ... ) that one of the worm; if you reach the last letter, the z, you start again from A, according to the logic of the finite arithmetic.

To simplify this operation Mr. Vigénère proposed the use of the following square table, made up of ordinate shifted alphabets.

For example if you want to code the first R of “ARRIVANO”, the column of the R will be identified, therefore you’ll go down the column until the line corresponding to the corresponding letter of the worms (verme), in this case E: the letter you cross is the coded one, the letter V; on the other hand the second R will be coded with the letter found in the line of the R of VERME, that is with the letter I.
Compared with the single-alphabetic codes, the advantage is evident: the same letter of the plaintext is not always coded in the same way; this makes vain the use of the statistic analysis in disincryption. The person who receives the text to decode, must use the reverse method (to subtract instead of adding). Referring to the above example you have the following:

Cipher text:   VVIUZVRUFDRAVUM  
Worm:         VERMEVERMEVERMEVE  
Plaintext:    ARR I VANO I R I IFONZ I  

You can decode the second V of the cipher text (VVIUZVRUFDRAVUM) looking for it in the line of the corresponding letter of the worm VERME, the letter E; the column, where the letter V is, has the clear letter R in the top first position.

Cryptography, cryptanalysis, and secret agent/courier betrayal featured in the Babington plot during the reign of Queen Elizabeth I which led to the execution of Mary, Queen of Scots. An encrypted message from the time of the Man in the Iron Mask (decrypted just prior to 1900 by Étienne Bazeries) has shed some, regrettably non-definitive, light on the identity of that real, if legendary and unfortunate, prisoner. Cryptography, and its misuse, were involved in the plotting which led to the execution of Mata Hari and in the conniving which led to the travesty of Dreyfus’ conviction and imprisonment, both in the early 20th century. Fortunately, cryptographers were also involved in exposing the machinations which had led to Dreyfus’ problems; Mata Hari, in contrast, was shot.

**Cryptography from 1800 to World War II**

Although cryptography has a long and complex history, it wasn't until the 19th century that it developed anything more than ad hoc approaches to either encryption or cryptanalysis (the science of finding weaknesses in crypto systems). Examples of the latter include Charles Babbage's Crimean War era work on mathematical cryptanalysis of polyalphabetic ciphers, rediscovered and published somewhat later by the Prussian Friedrich Kasiski.

Understanding of cryptography at this time typically consisted of hard-won rules of thumb; see, for example, Auguste Kerckhoffs’ cryptographic writings in the latter 19th century. Edgar Allan Poe developed systematic methods solving ciphers in the 1840s. In particular he placed a notice of his abilities in the Philadelphia paper *Alexander's Weekly (Express) Messenger*, inviting submissions of ciphers, which he proceeded to solve. His success created a public stir for some months.

He later wrote an essay on methods of cryptography which proved useful in deciphering the German codes employed during World War I.
Mathematical methods proliferated in the time leading up to World War II (notably in William F. Friedman's application of statistical techniques to cryptanalysis and cipher development and in Marian Rejewski's initial break into the German Army's version of the Enigma system). Both cryptography and cryptanalysis have become far more mathematical since WWII. Even so, it has taken the wide availability of computers, and the Internet as a communications medium, to bring effective cryptography into common use by anyone other than national governments or similarly large enterprises.

**Exercises**

Decode this cipher text, using the worm IT IS

```
E   X   T   U   W   F   M   L   W   L   I   F   A   X   D   W   Z   B   V   G
I   T   I   S
```

Cipher this message, you can choose the worm

```
W   E   A   R   E   H   A   P   P   Y   T   O   W   O   R   K   W   I   T   H   Y   O   U
```

2. **RSA Encryption**

The RSA algorithm was invented in 1978 by Ron Rivest, Adi Shamir, and Leonard Adleman. Here's the relatively easy to understand math behind RSA public key encryption.

1. Find $P$ and $Q$, two large (e.g., 1024-bit) prime numbers.
2. Choose $E$ such that $E$ is greater than 1, $E$ is less than $N=PQ$, and $E$ and $Z=(P-1)(Q-1)$ are relatively prime, which means they have no prime factors in common. $E$ does not have to be prime, but it must be odd. $Z$ can't be prime because it's an even number.
3. Compute $D$ such that $(DE - 1)$ is evenly divisible by $(P-1)(Q-1)$. Mathematicians write this as $DE = 1 \mod Z$, and they call $D$ the multiplicative inverse of $E$. This is easy to do -- simply find an integer $X$ which causes $D = (XZ+1)/E$ to be an integer, then use that value of $D$.
4. The encryption function is $C = (T^E) \mod N$, where $C$ is the ciphertext (a positive integer), $T$ is the plaintext (a positive integer), and $^*$ indicates exponentiation. The message being encrypted, $T$, must be less than the modulus, $Z$.
5. The decryption function is $T = (C^D) \mod N$, where $C$ is the ciphertext (a positive integer), $T$ is the plaintext (a positive integer), and $^*$ indicates exponentiation.

Your public key is the pair $(N, E)$. Your private key is the number $D$ (reveal it to no one).
Z is the **modulus**, E is the **public exponent**, D is the **secret exponent**.

You can publish your public key freely, because there are no known easy methods of calculating D, P, or Q given only (N, E) (your public key). If P and Q are each 1024 bits long, the sun will burn out before the most powerful computers presently in existence can factor your modulus into P and Q.

**EXAMPLE** (with not large prime numbers)

\[
P=11 \quad Q=17 \quad N=P \times Q = 187 \quad Z=(P-1)(Q-1)=160
\]

\[1<E<PQ \quad E=3 \quad (\text{odd and relatively prime with } 160)\]

I find D such that DE-1 is divisible by 160, \( D=107 \) in fact \( 107 \times 3 = 2 \times 160 \)

\[
T \text{ (plaintext)} = 123
\]

\[
C \text{ (ciphertext)} = 123^3 \mod 187 = 1860867 \mod 187 = 30
\]

\[
C \text{ (ciphertext)} = 30
\]

\[
T \text{ (plaintext)} = 30^{107} \mod 187 = 123
\]

**Public key** = 187 , 3  \quad **Private key** = 107

**RSA Algorithm**

In cryptography, **RSA** is an algorithm for public-key encryption. It was the first algorithm known to be suitable for signing as well as encryption, and one of the first great advances in public key cryptography. RSA is still widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys.

The algorithm was described in 1977 by Ron Rivest, Aid Shamir and Len Adleman at MIT; the letters RSA are the initials of their surnames.

Clifford Cocks, a British mathematician working for GCHQ, described an equivalent system in an internal document in 1973. Given the relatively expensive computers needed to implement it at the time it was mostly considered a curiosity and, as far as is publicly known, was never deployed. His discovery, however, was not revealed until 1997 due to its top-secret classification.

The algorithm was patented by MIT in 1983 in the United States of America. It expired on 21 September 2000. Since the algorithm had been published prior to patent application, regulations in much of the rest of the world precluded patents elsewhere. Had Cocks' work been publicly known, a patent in the US would not have been possible either.

**Key generation**

Suppose a user Monica wishes to allow Dawid to send her a private message over an insecure transmission medium. She takes the following steps to generate a public key and a private key:
1. Choose two large prime numbers \( p \) and \( q \) such that \( p \neq q \), randomly and independently of each other.

2. Compute \( n=pq \).

3. Compute the totient. \( \phi(n) = (p-1)(q-1) \)

4. Choose an integer \( e \) such that \( 1 < e < \phi(N) \) which is coprime to \( \phi(n) \).

5. Compute \( d \) such that \( de \equiv 1(\text{mod } \phi(n)) \).
   - prime numbers can be probabilistically tested for using Fermat's little theorem:
     \( a^{(p-1)} \equiv 1(\text{mod } p) \), if \( p \) is prime; testing with a few values for \( a \) gives an excellent probability that \( p \) is prime. (Carmichael numbers are composite numbers that can pass the test for all \( a \) with \( \gcd(a,n) = 1 \), but they are exceedingly rare.)
   - (Steps 4 and 5 can be performed with the extended Euclidean algorithm; see modular arithmetic.)
   - (Step 5, rewritten, can also be found by finding integer \( x \) which causes
     \[
     d = \frac{x(p-1)(q-1)+1}{e}
     \]
     to be an integer, then using the value of \( d(\text{mod}(p-1)(q-1)) \);
   - (From step 2) PKCS#1 v2.1 uses \( \lambda = \text{LCM}(p-1,q-1) \) instead of \( \phi = (p-1)(q-1) \).

The **public key** consists of
- \( n \), the modulus, and
- \( e \), the public exponent (sometimes encryption exponent).

The **private key** consists of
- \( n \), the modulus, which is public and appears in the public key, and
- \( d \), the private exponent (sometimes decryption exponent), which must be kept secret.

Usually, a different form of the **private key** (including CRT parameters) is stored:
- \( p \) and \( q \), the primes from the key generation,
- \( d \text{ mod } (p-1) \) and \( d \text{ mod } (q-1) \) (often known as \( dmp1 \) and \( dmq1 \))
- \( (1/q) \text{ mod } p \) (often known as \( iqmp \))

This form allows faster decryption and signing using the Chinese Remainder Theorem (CRT). In this form, all of the parts of the private key must be kept secret.

Monica transmits the public key to Dawid, and keeps the private key secret. \( p \) and \( q \) are sensitive since they are the factors of \( n \), and allow computation of \( d \) given \( e \). If \( p \) and \( q \) are not stored in the CRT form of the private key, they are securely deleted along with the other intermediate values from the key generation.
Encrypting messages
Suppose Bob wishes to send a message \( M \) to Alice. He turns \( M \) into a number \( m < n \), using some previously agreed-upon reversible protocol known as a padding scheme. Bob now has \( m \), and knows \( n \) and \( e \), which Alice has announced. He then computes the ciphertext \( c \) corresponding to \( m \):
\[
c = m^e \mod n
\]
This can be done quickly using the method of exponentiation by squaring. Bob then transmits \( c \) to Alice.

Decrypting messages
Alice receives \( c \) from Bob, and knows her private key \( d \). She can recover \( m \) from \( c \) by the following procedure:
\[
m = c^d \mod n
\]
Given \( m \), she can recover the original message \( M \). The decryption procedure works because
\[
c^d \equiv (m^e)^d \equiv m^{ed} \mod n.
\]
Now, since \( ed \equiv 1 \) (mod \( p-1 \)) and \( ed \equiv 1 \) (mod \( q-1 \)), Fermat’s little theorem yields
\[
m^{ed} \equiv m \pmod p \quad \text{and} \quad m^{ed} \equiv m \pmod q
\]
Since \( p \) and \( q \) are distinct prime numbers, applying the Chinese remainder theorem to these two congruences yields
\[
m^{ed} \equiv m \pmod {pq}, \quad \text{Thus,} \quad c^d \equiv m \pmod n.
\]

The RSA algorithm in “MATHEMATICA”
The application package “Mathematica” produced by Wolfram Research, is a very powerful instrument for the numerical and algebraic calculus, able to handle symbols and produce graphs. This program has a powerful interpreted language too. Below a procedure is given to implement the RSA algorithm through “Mathematica”.

FirstPrimeAbove[n_] := Block[{k}, k = n; While[!PrimeQ[k], k = k + 1]; Return[k]]

In[1]:=
p = FirstPrimeAbove[123456789123456789]
Out[1]= 123456789123456823

Out[2]= 987654321987654321

In[2]:=
q = FirstPrimeAbove[987654321987654321]
Out[2]= 987654321987654329

I load the function that find the prime number immediately next to \( n \)
I produce 2 prime numbers: \( p \) and \( q \)
I calculate their product \( n \).

I calculate the Eulero function of \( n \).

I factorize \( z \) to find \( d \), that must be prime with \( z \).

I build \( d \) (public key) doing the product of the prime numbers not belonging to the factorization of \( z \).

I check \( d \) and \( z \) are prime between them.

I calculate \( e \) (private key), that is the inverse of \( d \) in the finite arithmetic module \( z \).

I load the \textit{encode} and \textit{decode} functions that change a message of a text respectively in a sequence of numbers and viceversa.

I turn the text message in an integer number.

I assign the number to \( m \).

I code \( m \) with the formula: \( m^d \mod n \)

I decode the ciphered message with the formula: \( m^e \mod n \)

I turn the number just found in the plain text.

Class 3G - School year 2005/2006
and Cetoretta Valentina, Pasquarella Elisa, Silvia Tartuferi (Class 5F)
The teacher: Rosella Natalini.