Cryptography II
— Exercises —

Luca Viganò
Institut für Informatik
Albert-Ludwigs-Universität Freiburg
Solutions of the exercises of last week

EXERCISE 1. The ciphertext

QBB JXU MEHBT YI Q IJQWU QDT QBB JXU CUD QDT MECUD CUHUBO FBQOUHI
Solutions of the exercises of last week

EXERCISE 1. The ciphertext

QBB JXU MEHBT YI Q IJQWU QDT QBB JXU CUD QDT MECUD CUHUBO FBQOUHI

has been generated by an advanced Caesar cipher with shift 16 (i.e. “A” is mapped to “Q”) and thus decrypts to

ALL THE WORLD IS A STAGE AND ALL THE MEN AND WOMEN MERELY PLAYERS

As you like it, Act 2, Scene VII.

It is a better example for the decryption by frequency analysis than

A TALE TOLD BY AN IDIOT, FULL OF SOUND AND FURY, SIGNIFYING NOTHING.

simply because the frequency of letters is closer to the statistical one, i.e. to the relative frequencies in an English text of 1000 letters
One-Time Pads

EXERCISE 2. The two texts

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

were obtained by using the same one-time pad (mod 26)
One-Time Pads

EXERCISE 2. The two texts

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

were obtained by using the same one-time pad (mod 26)

A B C D E F G H I H G F E D C B A B C D E F G H
1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8

to encrypt the plaintexts

T H E W I N T E R O F O U R D I S C O N T E N T
A N D T H E R E S T I S S I L E N C E

(respectively from Richard III, Act 1, Scene I, and Hamlet, Act 5, Scene II).

For example,

• T (=20) + A (=1) = U (=21).

• W (=23) + D (=4) = A (=27 \mod 26 = 1).
One-Time Pads (cont.)

If a one-time pad is reused, decryption could be carried out according to the following strategy (similar to the one for the Vigenère cipher).

1. Assume that the first ciphertext contains the word THE somewhere. Hence, assume that the entire message consists of a series of THE's.
2. Work out the one-time pad that would be required to turn a whole series of THE's into the first ciphertext.
3. To find out which parts of this one-time pad are correct, apply it to the second ciphertext, and see if the resulting plaintext makes sense.
4. With some luck, we will be able to discern a few fragments of words in the second plaintext, indicating that the corresponding parts of the one-time pad are correct. This in turn shows which parts of the first message should be THE.
5. By expanding the fragments we have found in the second plaintext, we can work out more of the one-time pad, and then deduce new fragments in the first plaintext.
6. By expanding these fragments in the first plaintext, we can work out more of the one-time pad, and then deduce new fragments in the second plaintext.
7. We can continue this process until we have deciphered both ciphertexts.
The Churchyard Cipher: solution

EXERCISE 3.: the Churchyard Cipher (simplified)

- \[ \text{solution} \]

- \[ \text{simplified} \]
The Churchyard Cipher: solution

EXERCISE 3.: the Churchyard Cipher (simplified)

- ★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★...
Solutions of Exercises 4 and 5

- EXERCISE 4.: Explain why two substitution ciphers, applied one after another, may provide no more security than one substitution. (Such a cipher is called the product of the two underlying ciphers.)
Solutions of Exercises 4 and 5

• **EXERCISE 4.**: Explain why two substitution ciphers, applied one after another, may provide no more security than one substitution. (Such a cipher is called the product of the two underlying ciphers.)

Two substitution ciphers $S_1$ and $S_2$, applied one after another amount to one composed substitution cipher $S(p) = S_2(S_1(p))$.

Analogously, the product two transposition ciphers $T_1$ and $T_2$ following a regular pattern is also a regular pattern, namely $T_2(T_1(p))$.

• **EXERCISE 5.**: Explain why the product of two relatively simple ciphers, such as a substitution and a transposition, can achieve a high degree of security.
Solutions of Exercises 4 and 5

• EXERCISE 4.: Explain why two substitution ciphers, applied one after another, may provide no more security than one substitution. (Such a cipher is called the product of the two underlying ciphers.)

Two substitution ciphers $S_1$ and $S_2$, applied one after another amount to one composed substitution cipher $S(p) = S_2(S_1(p))$.

Analogously, the product two transposition ciphers $T_1$ and $T_2$ following a regular pattern is also a regular pattern, namely $T_2(T_1(p))$.

• EXERCISE 5.: Explain why the product of two relatively simple ciphers, such as a substitution and a transposition, can achieve a high degree of security.

DES is a good example for understanding why the product of two relatively simple ciphers, such as a substitution and a transposition, can achieve a high degree of security.

Another example is the ADFGVX cipher.
Some other ciphers: The Polybius Chequerboard

The Greek Polybius (≈200–118 b.C.) invented the Polybius Chequerboard, a monoalphabetic cipher that converts alphabetic characters into numeric characters.

Used to signal messages by holding different combinations of torches in each hand.

Using the English alphabet:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>e</td>
</tr>
<tr>
<td>2</td>
<td>f</td>
<td>g</td>
<td>h</td>
<td>ij</td>
<td>k</td>
</tr>
<tr>
<td>3</td>
<td>l</td>
<td>m</td>
<td>n</td>
<td>o</td>
<td>p</td>
</tr>
<tr>
<td>4</td>
<td>q</td>
<td>r</td>
<td>s</td>
<td>t</td>
<td>u</td>
</tr>
<tr>
<td>5</td>
<td>v</td>
<td>w</td>
<td>x</td>
<td>y</td>
<td>z</td>
</tr>
</tbody>
</table>

Each letter may be represented by two numbers by looking up the row the letter is in and the column. For instance h=23 and r=42.

Note that i and j share the same position. But this will not cause much of a problem when decoding as it will usually be obvious from the context which was intended!
The Playfair Cipher (1854) was popularized by Lyon Playfair, first Baron Playfair of St. Andrews, but it was invented by Sir Charles Wheatstone, one of the pioneers of the electric telegraph.

The cipher replaces each pair of letters in the plaintext with another pair of letters.

1. Sender and receiver agree on a keyword, say CHARLES, and then write alphabet in square $5 \times 5$, beginning with the keyword and combining I and J.

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

2. The message is broken up into pairs of letters (digraphs), where an X is inserted between equal letters and at the end of the message if necessary.

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>MEET ME AT HAMMERSMITH BRIDGE TONIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plaintext in digraphs</td>
<td>ME ET ME AT HA MX ME RS MI TH BR ID GE TO NI GH TX</td>
</tr>
</tbody>
</table>
Some other ciphers: The Playfair Cipher (cont.)

<table>
<thead>
<tr>
<th>C</th>
<th>H</th>
<th>A</th>
<th>R</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>S</td>
<td>B</td>
<td>D</td>
<td>F</td>
</tr>
<tr>
<td>G</td>
<td>I/J</td>
<td>K</td>
<td>M</td>
<td>N</td>
</tr>
<tr>
<td>O</td>
<td>P</td>
<td>Q</td>
<td>T</td>
<td>U</td>
</tr>
<tr>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
</tbody>
</table>

3. All digraphs fall into one of three categories: both letters in the same row, or the same column, or neither.

- If both letters are in the same row, they are replaced by the letter to the immediate right of each one, e.g. MI becomes NK. If one of the letters is at the end of the row, it is replaced by the letter at the beginning, e.g. NI becomes GK.
- If both letters are in the same column, they are replaced by the letter immediately beneath each one, e.g. GE becomes OG. If one of the letters is at the bottom of the column, it is replaced by the letter at the top, e.g. VE becomes CG.
Some other ciphers: The Playfair Cipher (cont.)

<table>
<thead>
<tr>
<th>C</th>
<th>H</th>
<th>A</th>
<th>R</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>S</td>
<td>B</td>
<td>D</td>
<td>F</td>
</tr>
<tr>
<td>G</td>
<td>I/J</td>
<td>K</td>
<td>M</td>
<td>N</td>
</tr>
<tr>
<td>O</td>
<td>P</td>
<td>Q</td>
<td>T</td>
<td>U</td>
</tr>
<tr>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
</tbody>
</table>

• If the letters of the digraph are neither in the same row nor in the same column, the encipherer follows a different rule.
  – To encipher the first letter, look along its row until you reach the column containing the second letter; the letter at this intersection then replaces the first letter.
  – To encipher the second letter, look along its row until you reach the column containing the first letter; the letter at this intersection then replaces the second letter.
  – Hence, ME becomes GD and ET becomes DO.
Some other ciphers: The Playfair Cipher (cont.)

The complete encryption thus is:

<table>
<thead>
<tr>
<th>Plaintext in digraphs</th>
<th>ME ET ME AT HA MX ME RS MI TH BR ID GE TO NI GH TX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ciphertext</td>
<td>GD DO GD RQ AR KY GD HD NK PR DA MS OG UP GK IC QY</td>
</tr>
</tbody>
</table>

4. The recipient, who also knows the keyword, can easily decipher the ciphertext by simply reversing the process; for example, enciphered letters in the same row are deciphered by replacing them by the letters to their left.

The British War Office secretly adopted the technique, probably using it in the Boer War.

However, the cipher can be attacked by looking for the most frequently occurring digraphs in the ciphertext, and assuming that they represent the commonest digraphs in English: TH, HE, AN, IN, ER, RE, ES.
Some other ciphers: The ADFGVX Cipher

- The ADFGVX Cipher features both substitution and transposition.

- Encryption begins by drawing a $6 \times 6$ grid, and filling the 36 cells with a random arrangement of the 26 letters and the 10 digits.

```
<table>
<thead>
<tr>
<th>A</th>
<th>D</th>
<th>F</th>
<th>G</th>
<th>V</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>8</td>
<td>P</td>
<td>3</td>
<td>D</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>L</td>
<td>T</td>
<td>4</td>
<td>O</td>
<td>A</td>
</tr>
<tr>
<td>F</td>
<td>7</td>
<td>K</td>
<td>B</td>
<td>C</td>
<td>5</td>
</tr>
<tr>
<td>G</td>
<td>J</td>
<td>U</td>
<td>6</td>
<td>W</td>
<td>G</td>
</tr>
<tr>
<td>V</td>
<td>X</td>
<td>S</td>
<td>V</td>
<td>I</td>
<td>R</td>
</tr>
<tr>
<td>X</td>
<td>9</td>
<td>E</td>
<td>Y</td>
<td>0</td>
<td>F</td>
</tr>
</tbody>
</table>
```

Each row and column of the grid is identified by one of A, D, F, G, V, or X.

- The arrangement of the elements in the grid acts as part of the key, so the receiver needs to know the details of the grid in order to decipher messages.
Some other ciphers: The ADFGVX Cipher (cont.)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>D</th>
<th>F</th>
<th>G</th>
<th>V</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>8</td>
<td>P</td>
<td>3</td>
<td>D</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>D</td>
<td>L</td>
<td>T</td>
<td>4</td>
<td>O</td>
<td>A</td>
<td>H</td>
</tr>
<tr>
<td>F</td>
<td>7</td>
<td>K</td>
<td>B</td>
<td>C</td>
<td>5</td>
<td>Z</td>
</tr>
<tr>
<td>G</td>
<td>J</td>
<td>U</td>
<td>6</td>
<td>W</td>
<td>G</td>
<td>M</td>
</tr>
<tr>
<td>V</td>
<td>X</td>
<td>S</td>
<td>V</td>
<td>I</td>
<td>R</td>
<td>2</td>
</tr>
<tr>
<td>X</td>
<td>9</td>
<td>E</td>
<td>Y</td>
<td>0</td>
<td>F</td>
<td>Q</td>
</tr>
</tbody>
</table>

- The first stage of encryption is to take each letter of the message, locate its position in the grid and substitute it with the letters that label its row and column.

For example, 8 would be substituted by AA, and P by AD, and thus

Message       ATTACK AT 10 PM
Plaintext     A T T A C K A T 1 0 P M
Stage 1 ciphertext DV DD DD DV FG DV DD AV XG AD GX

- So far, this is a simple monoalphabetic substitution cipher, and frequency analysis would be enough to crack it.
Some other ciphers: The ADFGVX Cipher (cont.)

- However the second stage of the ADFGVX cipher is a transposition, which makes cryptanalysis much harder.

- The transposition depends on a keyword, say MARK, which must be shared with the receiver, and it is carried out according to the following recipe.

1. The letters of the keyword are written in the top row of a fresh grid.
2. The stage 1 ciphertext is written underneath it in a series of rows

\[
\begin{array}{cccc}
M & A & R & K \\
D & V & D & D \\
D & D & D & V \\
F & G & F & D \\
D & V & D & D \\
A & V & X & G \\
A & D & G & X \\
\end{array} \quad \Rightarrow \quad
\begin{array}{cccc}
A & K & M & R \\
V & D & D & D \\
D & V & D & D \\
G & D & F & F \\
V & D & D & D \\
V & G & A & X \\
D & X & A & G \\
\end{array}
\]

and the columns in the grid are then rearranged so that the letters of the keyword are in alphabetical order.
Some other ciphers: The ADFGVX Cipher (cont.)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>D</th>
<th>F</th>
<th>G</th>
<th>V</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>8</td>
<td>P</td>
<td>3</td>
<td>D</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>D</td>
<td>L</td>
<td>T</td>
<td>4</td>
<td>O</td>
<td>A</td>
<td>H</td>
</tr>
<tr>
<td>F</td>
<td>7</td>
<td>K</td>
<td>B</td>
<td>C</td>
<td>5</td>
<td>Z</td>
</tr>
<tr>
<td>G</td>
<td>J</td>
<td>U</td>
<td>6</td>
<td>W</td>
<td>G</td>
<td>M</td>
</tr>
<tr>
<td>V</td>
<td>X</td>
<td>S</td>
<td>V</td>
<td>I</td>
<td>R</td>
<td>2</td>
</tr>
<tr>
<td>X</td>
<td>9</td>
<td>E</td>
<td>Y</td>
<td>0</td>
<td>F</td>
<td>Q</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>K</th>
<th>M</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>D</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>D</td>
<td>F</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>G</td>
<td>A</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>X</td>
<td>A</td>
<td>G</td>
<td></td>
</tr>
</tbody>
</table>

4. The final ciphertext is achieved by going down each column and then writing out the letters in this new order, i.e.

Message: ATTACK AT 10 PM
Plaintext: A T T A C K A T 1 0 P M
Stage 1 ciphertext: DV DD DD DV FG DV DD AV XG AD GX
Final ciphertext: V D G V V D D V D D G X D D F D A A D D F D X G

- The final ciphertext would then be transmitted in Morse code, and the receiver would reverse the encryption process in order to retrieve the original text.
- The entire ciphertext is made up of just 6 letters, i.e. A, D, F, G, V, X, because these are the labels of the rows and columns in the initial $6 \times 6$ grid.
- Choice of A, D, F, G, V, X (instead of, say, A, B, C, D, E, F) because they are highly dissimilar when translated into Morse dots and dashes, so that the risk of errors during transmission is minimized.
The Enigma Machine

Homework: consult the books (and the web) to find out the history and details of the Enigma Machine.
RSA based on number theory

- Prime numbers: \{2, 3, 5, 7, \ldots \}.

  Fundamental theorem of arithmetic: every \( n \in N \) has a unique set of prime factors

- Multiplying numbers is easy, factoring numbers appears hard.
  We cannot factor (most) numbers with \( \geq 1024 \) bits.

- \( a, b \in Z \) are congruent modulo \( n \), if \( a \mod n = b \mod n \).
  We write this as \( a \equiv b \pmod{n} \), or simply \( a \equiv_n b \).

- \( a, b \in N \) are relatively prime if their greatest common divisor is 1.

- Fermat’s little theorem: for \( a \) and \( n \) relatively prime and \( n \) prime
  \[
  a^{n-1} \equiv_n 1
  \]
**Proof of Fermat’s Theorem**

- **Claim:** The sequence $a \mod n, 2a \mod n, \ldots, (n - 1)a \mod n$ is a permutation of the integers from 1 to $n - 1$.

  Suppose not. Then, since 0 isn’t in the sequence ($a$ and $n$ are relatively prime), we must have $xa \equiv_n ya$ for $x, y < n$. But then we can cancel $a$ (again, since $a$ and $n$ are relatively prime) and get $x \equiv_n y$ and thus $x = y$. Contradiction!

- Now observe

  $$[(a \mod n) \times \ldots \times ((n - 1)a \mod n)] = 1 \times 2 \times \ldots (n - 1)$$

  $$= (n - 1)!$$

- However $\text{LHS} = (n - 1)! \times a^{n-1} \mod n$

- Therefore $(n - 1)! \times a^{n-1} \equiv_n (n - 1)!$

- Because $n$ is prime, $(n - 1)!$ is relatively prime to $n$.

- We can thus cancel $(n - 1)!$ from congruence, yielding

  $$a^{n-1} \equiv_n 1$$
RSA algorithms

Following operations are all easy:

- **Generate a public/private key pair:**
  1. Generate two large distinct primes $p$ and $q$
  2. Compute $n = pq$ and $\phi = (p - 1)(q - 1)$
  3. Select an $e$, $1 < e < \phi$, relatively prime to $\phi$.
  4. Compute the unique integer $d$, $1 < d < \phi$ where $ed \equiv 1$ mod $\phi$.
  5. Return public key $(n, e)$ and private key $d$

- **Encryption** with key $(n, e)$
  1. Represent the message as an integer $m \in \{0, \ldots, n - 1\}$
  2. Compute $c = m^e \mod n$

- **Decryption** with key $d$: compute $m = c^d \mod n$
An RSA example

• Let $p = 47$, $q = 71$, then $n = pq = 3337$.

• Encryption key $e$ must have no factors in common with $$(p - 1)(q - 1) = 46 \times 70 = 3320.$$ 

• Choose $e = 79$ (randomly).

• Compute $d = 79^{-1} \mod 3320 = 1019$ (with “extended Euclidean algorithm”).

• Publish $e$ and $n$, keep $d$ secret, discard $p$ and $q$.

• Break message $m$ into small blocks, e.g., $m = 688\ 232\ 687\ 966\ 668$.

• First block encrypted as $688^{79} \mod 3337 = 1570 = c_1$.

• To decrypt: $1570^{1019} \mod 3337 = 688 = m_1$. 
Another RSA example

• Alice picks two giant prime numbers \( p \) and \( q \); for simplicity, let \( p = 17 \), \( q = 11 \), so that \( n = pq = 187 \).

• Alice picks encryption key \( e \), relatively prime to \( (p - 1)(q - 1) \); again for simplicity, let \( e = 7 \).

• Alice publishes \( e \) and \( n \) (her public key) in something akin to a telephone directory, so that anybody might use them to encrypt a message for Alice.

(Note that \( e \) could also be part of everybody else’s public key; what matters is that everybody has a different value of \( n \) (depending on choice of \( p \) and \( q \)).

• Encryption of a message \( M \) by \( C = M^e \mod N \).
  – Bob wants to send Alice a simple kiss: just the letter X.
  – X in ASCII is 1011000, i.e. 88 in decimal, so \( M = 88 \) and \( C = 88^7 \mod 187 \).
Another RSA example (cont.)

• To compute $C = 88^7 \mod 187$, observe that (since $7 = 4 + 2 + 1$)

\[
88^7 \mod 187 = [(88^4 \mod 187) \times (88^2 \mod 187) \times (88^1 \mod 187)] \mod 187.
\]

Hence, the ciphertext is $C = 11$

\[
\begin{align*}
88^1 \mod 187 &= 88 \mod 187 = 88 \\
88^2 \mod 187 &= 7744 \mod 187 = 77 \\
88^4 \mod 187 &= 59969536 \mod 187 = 132 \\
88^7 \mod 187 &= 988 \times 77 \times 132) \mod 187 = 894432 \mod 187 = 11
\end{align*}
\]

• Exponentials in modular arithmetic are one-way functions, so it is difficult for an eavesdropper Eve to work backward from $C = 11$ and recover the original message.
Another RSA example (cont.)

• Alice can decipher the message as she has some special information: she knows the values of $p$ and $q$, and she can thus calculate (with “extended Euclidean algorithm”) the decryption key $d$ (her private key) according to the formula

$$e \times d \equiv (p-1) \times (q-1) \mod 1$$

• In our example, $e \times d \equiv (p-1) \times (q-1) \mod 1$ is $7 \times d \equiv (16 \times 10) \mod 1$, so that $d = 23$.

• Alice then decrypts the message using the formula:

$$M = C^d \mod n$$

so that $M = 11^{23} \mod 187 = 88 = X$ in ASCII.
RSA exercises

• Consider the RSA algorithm with public keys \( n = 55 \) and \( e = 7 \).
  
  – Encipher the plaintext \( M = 10 \).
  – Break the cipher by finding \( p, q \) and \( d \).
  – Decipher the ciphertext \( C = 35 \).

• Perform encryption and decryption using the RSA algorithm for the following:

  1. \( p = 3, q = 11, d = 7, M = 5 \).
  2. \( p = 5, q = 11, e = 3, M = 9 \).
  3. \( p = 7, q = 11, e = 17, M = 8 \).
  4. \( p = 11, q = 13, e = 11, M = 7 \).
  5. \( p = 17, q = 31, e = 7, M = 2 \). Hint: decryption is not as hard as you think; use some finesse.
Message authentication

- **Problem of public-key ciphers:** since anyone can send an encrypted message to a receiver (rather than just those in possession of the secret key for a symmetric-key cipher), the receiver cannot be certain that the sender is who she claims to be or that the message is what was intended by the original sender.

- Several possible attacks to public-key communication, for example

  **Masquerade:** a message could be sent from a fraudulent source.
  
  **Content modification:** the content of a message could be modified via insertion, deletion, transposition, or reordering.
  
  **Sequence modification:** a sequence of messages could be modified using the same methods as in content modification.

- Therefore, we must develop a system for **message authentication** to complement public-key ciphers.

There are several ways this can be accomplished, for example by

- Message encryption.
- Message Authentication Code.
- Hash Functions.
Message Authentication Code (MAC)

- Suppose that two agents A and B share a common secret key $K$, and that A wants to send a message to B.

- Then a small block of data, known as a cryptographic checksum or MAC, which is a function of the message to be sent, can be encrypted using this secret key and appended to the message.

- The combined plaintext is encrypted using the main cipher and sent to B.

- B decrypts the message and extracts and decrypts the MAC.
  - If the decrypted MAC matches the MAC computed from the decrypted message, then B is assured that the message has not been altered.
  - Furthermore, B is assured that the message came from A since the secret key used to encrypt the MAC is known only by A and B.
Message Authentication Code (MAC; cont.)

- A MAC represents a function $C_K(M)$ of the message $M$ and should have the following properties:

  1. If an opponent observes $M$ and $C_K(M)$, it should be computationally infeasible to find $M'$ such that $C_K(M') = C_K(M)$.

  2. $C_K(M)$ should be uniformly distributed, that is, if $M$ and $M'$ are two randomly selected messages, then

     $$P(C_K(M') = C_K(M)) = 2^{-n}$$

     where $n$ is the bit-length of the MAC.

  3. For some known transformation $M' = f(M)$,

     $$P(C_K(M') = C_K(M)) = 2^{-n}$$
Message Authentication Code (MAC; cont.)

- One of the most widely used MAC is based on DES encryption.
  - This MAC is obtained by performing cipher block chaining mode of DES to the message to be authenticated using an initial vector 0 and the secret key.
  - The message is split into contiguous 64-bit blocks with the last block padded if necessary to 64-bits.
  - The MAC is the last output block of the result.
Hash functions

- An alternative to a MAC is a hash function.

- This is a function of all of the bits in a message so that any change in the bits results in a change in the hash code.

- Properties that a hash function $H$ should satisfy are:
  1. $H$ can be applied to a block of data of any size.
  2. $H$ produces a fixed-length output.
  3. $H(x)$ is relatively easy to compute for any input $x$.
  4. For any given code $h$, it is computationally infeasible to find $x$ such that $h = H(x)$ (one-way property).
  5. For any given block $x$, it is computationally infeasible to find $y \neq x$ such that $H(y) = H(x)$ (weak collision resistance).
  6. It is computationally infeasible to find a pair $(x, y)$ such that $H(y) = H(x)$ (strong collision resistance).
**Hash functions (cont.)**

- Advantage of using hash codes: if the hash code is encrypted using symmetric-key or public-key encryption, then authentication is ensured since presumably only the sender could have sent the correct hash code.

- One simple hash function: XOR of every block

\[ C_i = b_{i1} \oplus b_{i2} \oplus \ldots \oplus b_{im} \]

but this scheme is not useful for data security since is it easy to generate messages that have the specified hash code.

- More complicated hash codes (and MAC’s) exist, even though they are still not always free from attacks.

- A widely used hash function is **MD5** (Message Digest Algorithm, version 5), which is used, for example, by many Linux distributions to authenticate their ISO files (files that can be burned onto CD’s).

- A hash function that is similar to but is more resistant to attacks than MD5 is **SHA-1**.
Secure Hash Algorithm (SHA-1)


- When a message of any length less than $2^{64}$ bits is input, the SHA-1 produces a 160-bit output called a message digest.

- The message digest can then be input to the Digital Signature Algorithm (DSA), which generates or verifies the signature for the message.

- Signing the message digest rather than the message often improves the efficiency of the process because the message digest is usually much smaller in size than the message.

- The same hash algorithm must be used by the verifier of a digital signature as was used by the creator of the digital signature.

- The SHA-1 is called secure because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest.
Consult the books (and the web) to learn more about

- the Message Authentication Code MAC

- and the Secure Hash Algorithm SHA-1 (and the Digital Signature Algorithm DSA; see also the next classes).