IT Security

Ciphers

David Schneider

Class T3

Table of content

1 Caesar/Rot 13

The Caesar Cipher is one of the simplest cipher techniques and is also known as shift cipher.
The key is used as shift value and can be either a number or a alphabetic character.
ROT 13 has the specialty that the key (shift value) is the half of the text in clear. It can only be used if the text has a pair length.

Example Caesar:

Text:	Key:	New alphabet:	Message:
DAFE	L	LMNOPQRSTUVWXYZABCDEFGHIJK	OLQP

Example Rot 13:

Text:	Key:	New alphabet:	Message:
DAFE	2	CDEFGHIJKLMNOPQRSTUVWXYZAB	FCHG

2 Vigenère Cipher

The Vigenère Cipher is very similar to the Caesar Cipher. The difference is that the key has multiple characters which are used consecutively.

Example:

Text:	Key:	Message:
CRYPTOGRAPHIE	DAFE	CRYP TOGR APHI DAFE DAFE DAFE **FRDT WOLV DPMC**

3 Hill Cipher

Hill Cipher is based on linear algebra (matrix theory). The key is a quadratic matrix. The number of dimensions of the matrix indicates the block size. The chosen matrix needs to be invertible.
The modulo 26 in my example comes from the number of characters of the chosen alphabet (A-Z).

Text:	Key:	Message:
DAFE	$\begin{pmatrix} M & B \\ T & T \end{pmatrix} = \begin{pmatrix} 12 & 01 \\ 19 & 19 \end{pmatrix}$	KFMP (10;5;12;15)

$$\begin{pmatrix} 12 & 01 \\ 19 & 19 \end{pmatrix} \cdot \begin{pmatrix} 03(D) \\ 0(A) \end{pmatrix} = \begin{pmatrix} 36+0 \\ 57+0 \end{pmatrix} = \begin{pmatrix} 36\,mod\,(26) \\ 57\,mod\,(26) \end{pmatrix} = \begin{pmatrix} 10 \\ 5 \end{pmatrix}$$

$$\begin{pmatrix} 12 & 01 \\ 19 & 19 \end{pmatrix} \cdot \begin{pmatrix} 5(F) \\ 4(E) \end{pmatrix} = \begin{pmatrix} 60+4 \\ 95+76 \end{pmatrix} = \begin{pmatrix} 64\,mod\,(26) \\ 171\,mod\,(26) \end{pmatrix} = \begin{pmatrix} 12 \\ 15 \end{pmatrix}$$

4 Atbash

Atbash is a simple cipher that substitutes the last character with the first, the second last with the second and so on. In other words it inverses the alphabet. It can be applied to any alphabet, originally the Hebrew alphabet was used.

Text:	New alphabet:	Message:
DAFE	ZYXWVUTSRQPONMLKJIHGFEDCBA	WZUV

5 XOR

The XOR cipher uses the logical exclusive or between the key and the plain text. XOR equals true if only one of the bits is true.

Text (8bit ASCII):	Key:	Message:
DAFE (68, 65, 70, 69)	$A9_{16} = 169_{10} = 10101001_2$	íèïî

```
      01000100   01000001   01000110   01000111
XOR   10101001   10101001   10101001   10101001
      11101101   11101000   11101111   11101110

      237= í     232=è      239=ï 238=î
```

Output message depends on character encoding (here: ISO 8859-1)

6 Homophone

Homophone is also a substitution cipher. Not all the characters of a language are used as many times. This allows frequency analysis attacks. This cipher counts how often a character appears and allocates it to one or more than one cipher text symbol. The number of allocations depends on the frequency and the maximal number of homophones.

Examples for the frequency of characters[1]:

Character:	English:	German:	French:	Spanish:
A	8,2 %	6,5 %	7,6 %	12,5 %
E	12,7 %	17,4 %	14,7 %	13,7 %
J	0,2 %	0,2 %	0,5 %	0,4 %

1 Quelle: http://de.wikipedia.org/wiki/Buchstabenh%C3%A4ufigkeit (22.01.08)

7 RC4

RC4 is a stream cipher which is used for example by the protocol Secure Sockets Layer (SSL) to protect internet traffic or Wired Equivalent Privacy
(WEP) to secure wireless networks.
RC4 generates a variable sized keystream (pseudorandom bits) and applies it bit-wise to the plain text by using the XOR operation.

8 3-DES

Triple Data Encryption Standard is a block cipher (64 bit block size) developed by IBM. DES shifts each round to obtain a new 56 bit key witch results a total of 168 bit key for 3-DES. The expansion allowed to enlarge the key size without changing the algorithm.

9 RSA

RSA is a public key algorithm developed by Rivest, Shamir and Adleman. Public key means that the algorithm works asymmetric with two key, one to crypt and one to encrypt. It was the first method able to be used for ciphering and for digital signatures. The algorithm uses modulo calculation and prime factors and takes advantage of the mathematical problem, that big numbers can't be split fast into prime factors. For example a 640 bit number can be factored by a 2.2GHz-Opteron-CPU in about 30 years[2].

How does RSA works:

1. Code the message.
   HELLO in ASCII = 08 05 12 12 15
   Can be regrouped to avoid frequency analysis (see point 6)
   HELLO = 080 512 121 5

2. Choose 2 big primes p & q (RSA ~300 digits = 2048 bit)
   p=37 q=19

2 Quelle: http://www.rsa.com/rsalabs/node.asp?id=2964 (22.01.08)

3. Calculation

Calculate N: $N = p \cdot q = 37 \cdot 19 = 703$
Calculate Phi: $\phi = (p-1) \cdot (q-1) = 36 \cdot 18 = 648$
Choose e: must be smaller than N and coprime to Phi
$e = 35$
Calculate d: $e \cdot d \bmod \phi = 1$
$d = 611$

Public key (N, e) **(703, 35)**

Private key (N, d) **(703, 611)**

4. Calculate cipher message

$$C = M^e \bmod N$$

$$C1 = 080^{35} \bmod 703 = 290$$
$$C2 = 512^{35} \bmod 703 = 265$$
$$C3 = 121^{35} \bmod 703 = 581$$
$$C4 = 005^{35} \bmod 703 = 422$$

5. Decipher

$$M = C^d \bmod N$$

$$M = 290^{611} \bmod 703 = 080$$
$$M = 265^{611} \bmod 703 = 512$$
$$M = 581^{611} \bmod 703 = 121$$
$$M = 422^{611} \bmod 703 = 005$$

Verification with Mathematica:

```
Mod[{290^611, 265^611, 581^611, 422^611}, 703]
Out[10]= {80, 512, 121, 5}
```

Although I used for this example very small primes the TI-89 calculator wasn't able to figure out the solution of the modulo calculations.

10 Hash function

Hash functions are mathematical functions which reduces any kind of digital data into a small integer. The functions are one-way, which means that its not possible to reconstruct the message out of the hash value. Furthermore two different messages wont give the same hash.
Hash functions are used for checksums, fingerprints, error correction and for cryptography.
Examples: MD2, MD4, MD5, SHA, Tiger, Whirlpool and so one.

SHA

SHA or Secure Hash Algorithm is a Hash function designed by the cryptologic intelligence agency of the United States government. SHA enhanced after 1993 where the original algorithm was published and is now available in different versions.

SHA-1 added to the original algorithm a bitwise rotation to provide greater resistance to attacks. This version produces a hash value with 160-bit and can be used on a message with a maximum length of 2^{64} bits.

SHA-2 is the family name of the algorithms that followed. The variants are named after the length of hash key they produce. The four versions are SHA-224, SHA-256, SHA-384 and SHA-512. The block size is 512 bits (SHA-512 uses 1024 bit). The maximal length is the same as on the first variant of SHA 2^{64} bits apart for the 512 bit SHA which has a maximal message size of 2^{128} bit.

SHA-3 development is in progress.

Hash-time

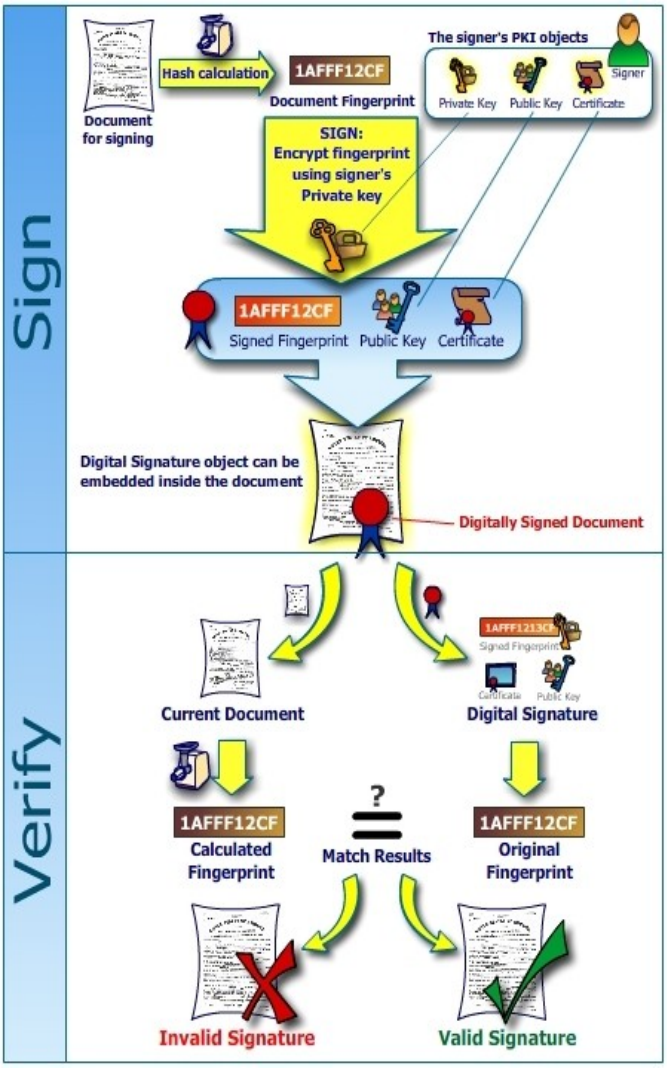
Used time to calculate hash function:

	MD5	**SHA-1**	**SHA 224**	**SHA 256**	**SHA 512**
1 MB	0.005s	0.007s	0.011s	0.012s	0.009s
5 MB	0.019s	0.034s	0.047s	0.046s	0.032s
10 MB	0.032s	0.059s	0.095s	0.097s	0.069s

For calculation the linux command time has been used. The output value "real" has been used. The values variate if the command is executed several times because of CPU usage for other processes.

11 Digital Signature

Digital Signature is also a asymmetric type of cryptography. The aim of a digital signature is to displace the security properties of a handwritten signature which means that the reader can be sure that the message comes from the person denoted.
To prove that the content of the message hasn't been changed on the way trough the internet, a hash function is used.
A hash function is a mathematical function which reduces, in our example a message, to a integer of a given size (kind of checksum). Different messages can't give the same output.
To ensure the integrity of the message, the asymmetric encryption is applied (public and private key), this time not the same way as we saw on the example of RSA. The message is now crypted with the private key and needs to be mad readable with the public key.



Img. 1: Digital Signature
http://en.wikipedia.org/wiki/Image:Digital_Signature_-_How_it_works.jpg