Introduction to Cryptography

Outline

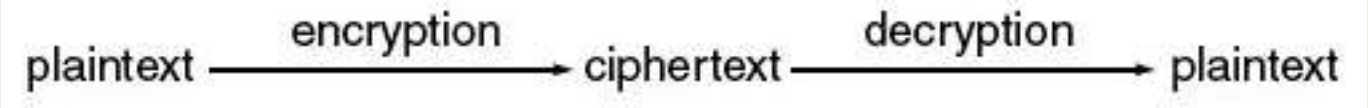
2. Introduction to Cryptography.

- What Is Cryptography?
- Breaking an Encryption Scheme.
- Types of Cryptographic Functions.
- Secret Key Cryptography.
- Public Key Cryptography.
- Hash Algorithms.

Cryptography

κρυπτο γραφη

The art of secret writing.
The art of mangling information into apparent unintelligibility
in a manner allowing a secret method of unmangling.



Fundamental Tenet of Cryptography

If lots of smart people have failed to solve a problem, then it probably won't be solved (soon).

Cryptography Definitions

- Messages:
  - Plaintext
  - Ciphertext
- Ingredients:
  - Algorithm(s)
  - Key(s)
- Players:
  - *Cryptographer*: invents clever algorithms
  - *Cryptanalyst*: breaks clever algorithms

Cryptography

- Algorithm
- Key(s) = secret value(s)
- OK for good algorithm to be public
  - Not OK to use bad algorithm
  - “Sunlight is the best disinfectant”
  - Algorithm without key does not help unmangle the information

Computational Issues

- Algorithm should be reasonably efficient
- Security depends on how hard it is to break
- Combination lock
  - 3 number sequence (2R, 1L, 0R), #s between 1-40
    - Possible combinations:

Computational Issues

- Algorithm should be reasonably efficient
- Security depends on how hard it is to break
- Combination lock
  - 3 number sequence (2R, 1L, 0R), #s between 1-40
    - Possible combinations: $40^3 = 64{,}000$
    - 10 seconds per sequence: 178 hours (/ 2 = 89)

Computational Issues

- Algorithm should be reasonably efficient
- Security depends on how hard it is to break
- Combination lock
  - 3 number sequence (2R, 1L, 0R), #s between 1-40
    - Possible combinations: $40^3 = 64{,}000$
    - 10 seconds per sequence: 178 hours (/ 2 = 89)

Computational Issues

- Algorithm should be reasonably efficient
- Security depends on how hard it is to break
- Combination lock
  - 3 number sequence (2R, 1L, 0R), #s between 1-40
    - Possible combinations: $40^3 = 64{,}000$
    - 10 seconds per sequence: 178 hours (/ 2 = 89)
  - 4 number sequence, 13 seconds per sequence

Computational Issues

- Algorithm should be reasonably efficient
- Security depends on how hard it is to break
- Combination lock
  - 3 number sequence (2R, 1L, 0R), #s between 1-40
    - Possible combinations: $40^3 = 64{,}000$
    - 10 seconds per sequence: 178 hours (/ 2 = 89)
  - 4 number sequence, 13 seconds per sequence
    - $40^4 = 2{,}560{,}000$
    - 9,244 hours

Computational Issues

- Algorithm should be reasonably efficient
- Security depends on how hard it is to break
- Combination lock
  - 3 number sequence (2R, 1L, 0R), #s between 1-40
    - Possible combinations: $40^3 = 64{,}000$
    - 10 seconds per sequence: 178 hours (/ 2 = 89)
  - 4 number sequence, 13 seconds per sequence
    - $40^4 = 2{,}560{,}000$
    - 9,244 hours

wikiHow
to do anything
Home » Categories » Home and Garden » Home Improvements and Repairs » Tools » Locks and Keys

How to Crack a "Master Lock" Combination Lock

125 authors | 664 revisions | Last updated: November 21, 2012

Publication Issues

- Public or secret algorithms?

Publication Issues

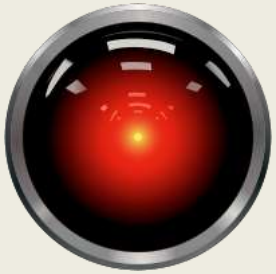
- Public or secret algorithms:
  - OK for good algorithm to be public
  - “Sunlight is the best disinfectant”
  - Reverse engineering, leaks
  - “Free consulting” by cryptanalysts
- Generally:
  - Commercial: public
  - Military: secret

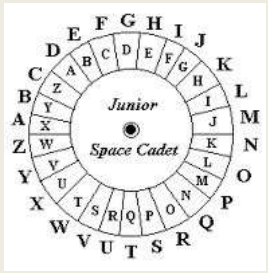
Secret Codes

- Substitution Ciphers
  - Caeser Cipher
  - Vigenere Cipher
  - Simple? Substitution Cipher

Secret Codes

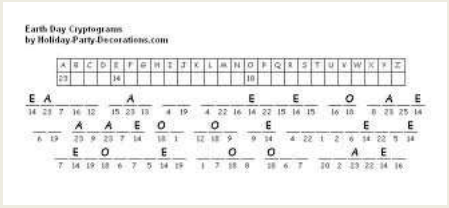
- Caesar cipher
  - Substitute letter 3 letters further on
    - DOZEN → GRCHQ
- Captain Midnight Secret Decoder Rings
  - Substitute letter *n* letters further on (*n* = 1..25)
    - HAL → IBM (*n* = 1)





Secret Codes

- Caesar cipher
  - Substitute letter 3 letters further on
    - DOZEN → GRCHQ
- Captain Midnight Secret Decoder Rings
  - Substitute letter *n* letters further on (*n* = 1..25)
    - HAL → IBM (*n* = 1)
- Monoalphabetic cipher
  - Arbitrary mappings (26! = $4.03291461 \times 10^{26}$)
    - 1 ms / try → 10M years … but: letter frequencies



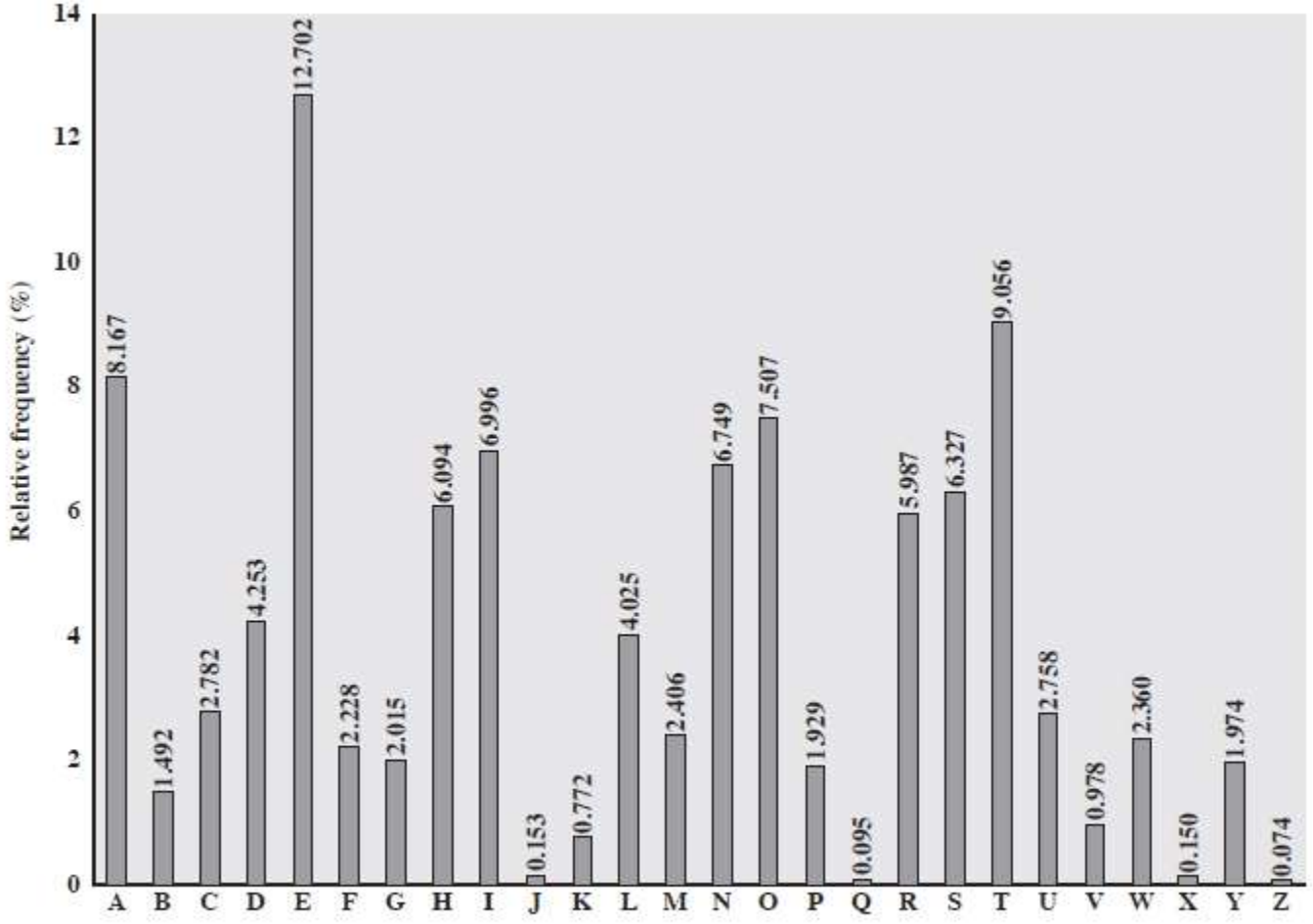


Figure 2.5 Relative Frequency of Letters in English Text

[from *Stallings, Cryptography & Network Security*]

Secret Codes

- Polyalphabetic cipher
  - Where the same character can be mapped onto different characters
  - E.g. Vigenere Cipher
  - Can Frequency Analysis work here?

- Vigenere Cipher Encryption
  - Plaintext: “GEEKS” Key: “FIRST”
  - Ciphertext: ?

	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

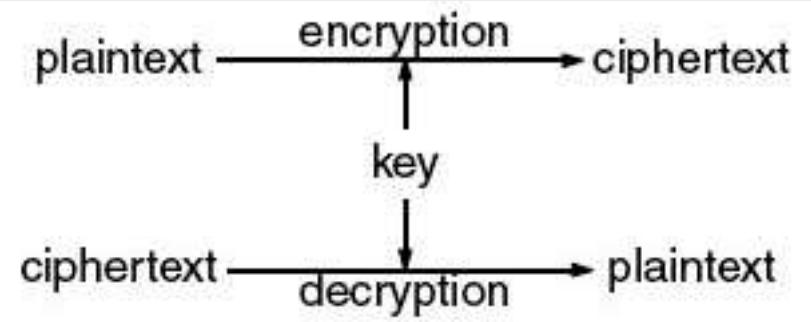
Breaking an Encryption Scheme

- Ciphertext only
  - Try all possible keys, look for intelligibility
  - Need sufficiently long ciphertext
  - XYZ = ? The hot cat was sad but you may now sit and use her big red pen.
- Known plaintext
  - (plaintext, ciphertext) pair(s)
- Chosen plaintext
  - Have plaintext encrypted, compare expected values

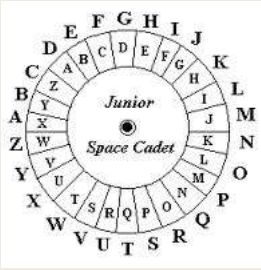
Types of Cryptography

- Public Key
  - Two keys: public & private
- Symmetric Key (aka “Secret Key”)
  - One key: secret (but possibly shared)
- Hash Functions
  - No keys

Symmetric Key Cryptography



AKA *secret key cryptography*
AKA *conventional cryptography*



Symmetric Key Applications

- Transmission over insecure channel
  - Shared secret (transmitter, receiver)
- Secure storage on insecure media
- Authentication
  - *Strong* authentication: prove knowledge without revealing key

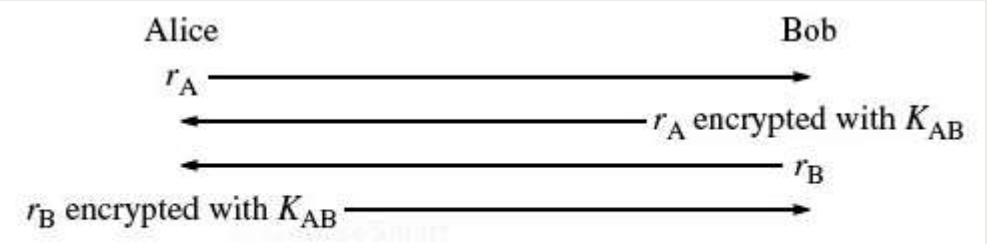


Figure 2-1. Challenge–response authentication with shared secret

A simple example

- K_{AB} = +3 (Caesar cipher), known by Alice & Bob
- r_A = “marco”
  - r_A encrypted with K_{AB}:
- r_B = “polo”
  - r_A encrypted with K_{AB}:

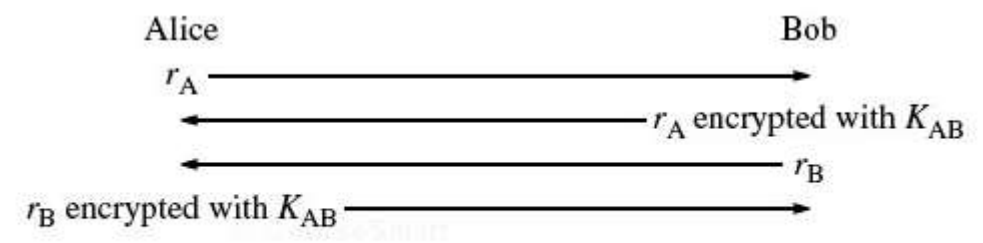


Figure 2-1. Challenge–response authentication with shared secret

A simple example

- K_{AB} = +3 (Caesar cipher), known by Alice & Bob
- r_A = “marco”
  - r_A encrypted with K_{AB}: “pdufr”
- r_B = “polo”
  - r_A encrypted with K_{AB}: “sror”

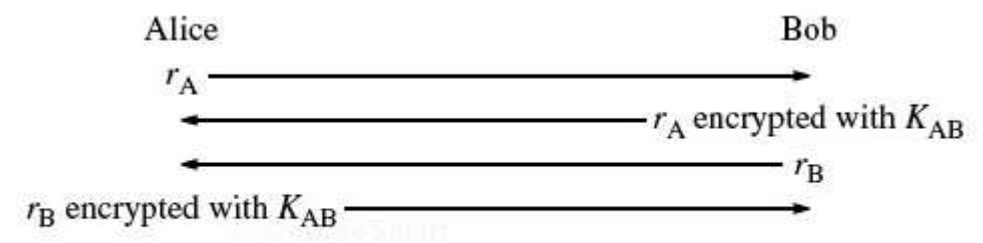


Figure 2-1. Challenge–response authentication with shared secret

A simple example

- K_{AB} = +3 (Caesar cipher), known by Alice & Bob
- r_A = “marco”
  - r_A encrypted with K_{AB}: “pdufr”
- r_B = “polo”
  - r_A encrypted with K_{AB}: “sror”
- (“marco”, “pdufr”), (“polo”, “sror”)

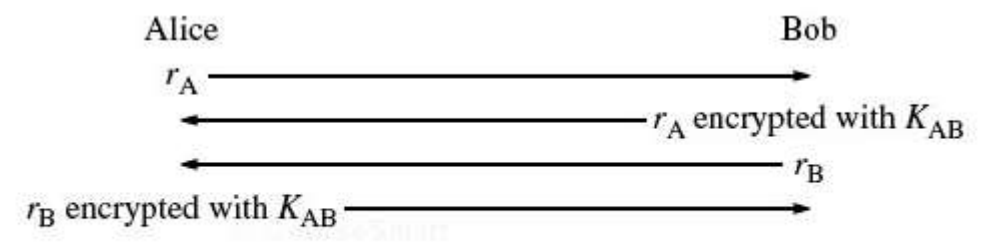
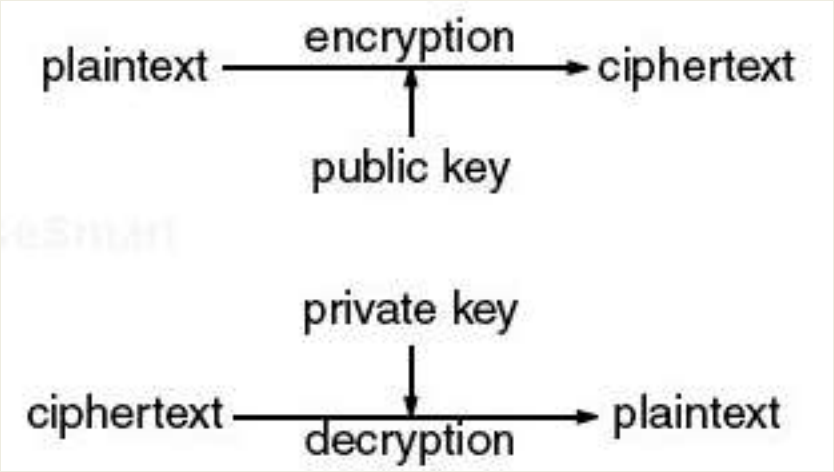


Figure 2-1. Challenge–response authentication with shared secret

Public Key Cryptography

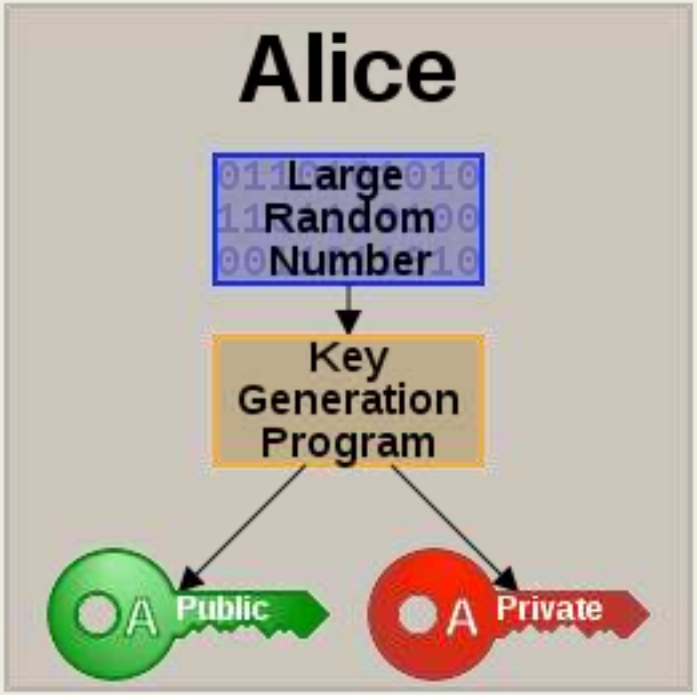


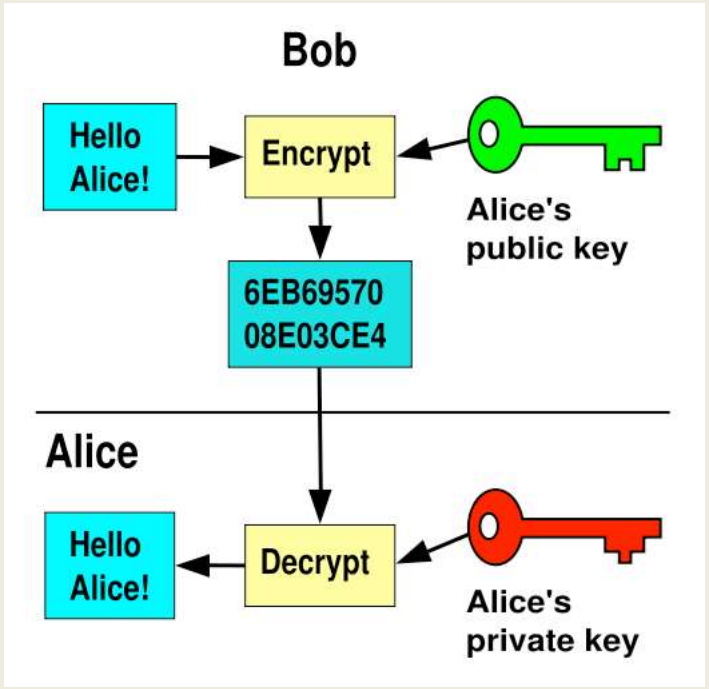
AKA *asymmetric cryptography*

AKA *unconventional cryptography* (?)

Public key: published, ideally known widely
Private key (NOT “secret key”): not published

Public Key Cryptography





http://en.wikipedia.org/wiki/Public_key_cryptography

Public Key Cryptography Issues

- Efficiency
  - Public key cryptographic algorithms are orders of magnitude slower than symmetric key algorithms
- Hybrid model
  - Public key used to establish temporary shared key
  - Symmetric key used for remainder of communication

SSL / TLS

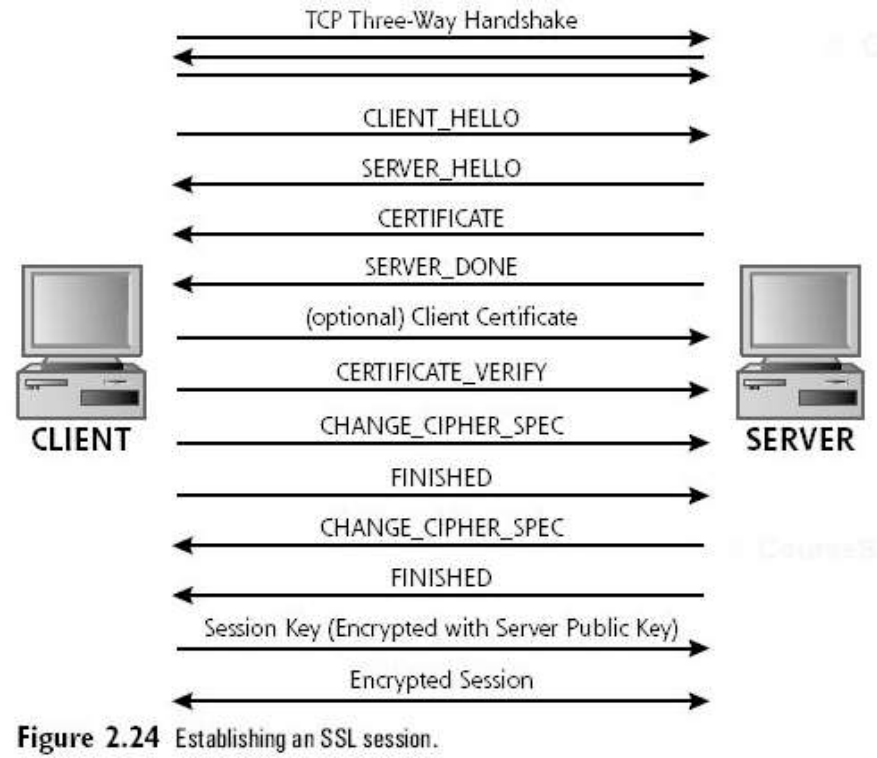
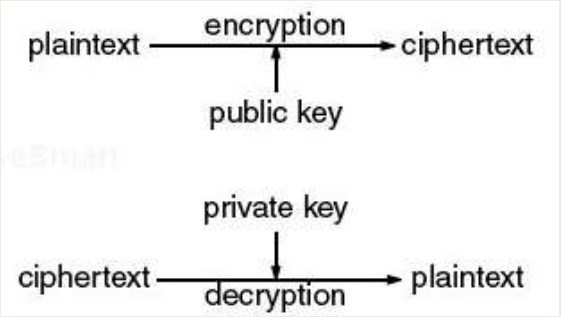


Figure 2.24 Establishing an SSL session.

- CLIENT_HELLO:
  - Available crypto & compression algorithms
  - Highest SSL/TLS protocol version
  - SSL Session ID
  - Random data
- SERVER_HELLO
  - Specific crypto & compression
  - Specific SSL version
  - SSL Session ID
  - Random data
- CERTIFICATE
  - Server's public encryption key
- Session Key
  - Server's public encryption key + random data

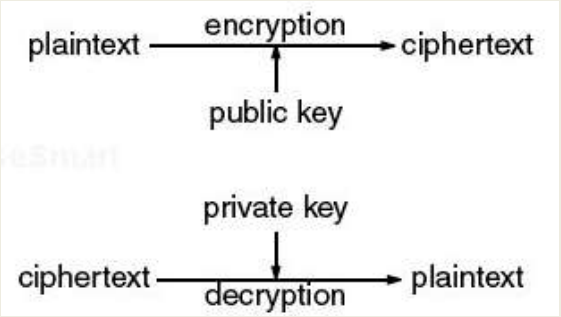
Secure Storage on Insecure Media

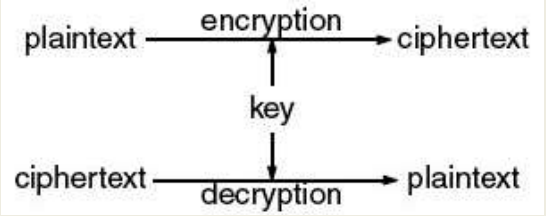
- Option 1:
  - Encrypt with public key
  - Decrypt with private key



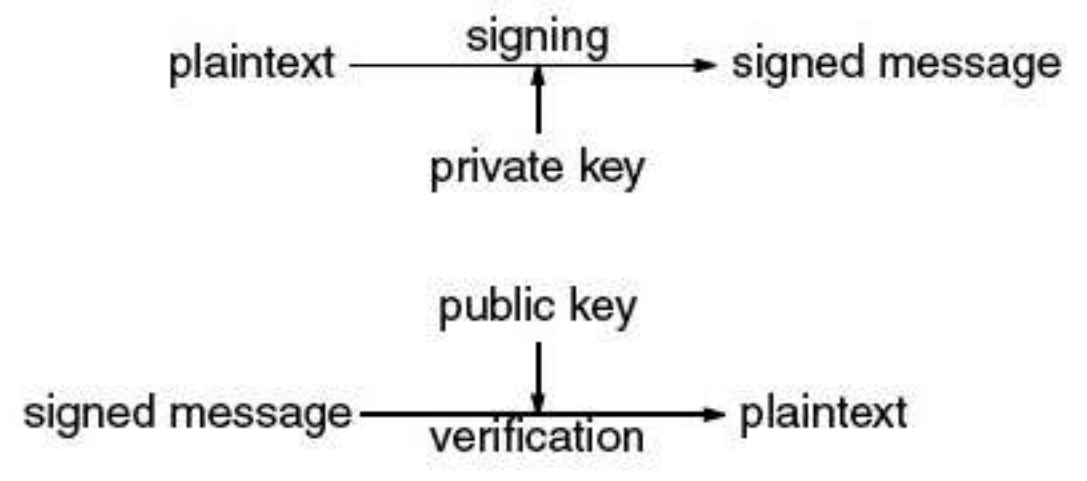
Secure Storage on Insecure Media

- Option 1:
  - Encrypt with public key
  - Decrypt with private key
- Option 2:
  - Generate random secret key
  - Encrypt with that secret key
  - Encrypt secret key with public key



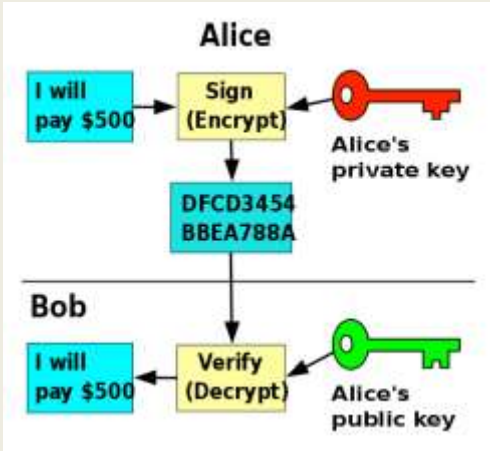


Digital Signatures



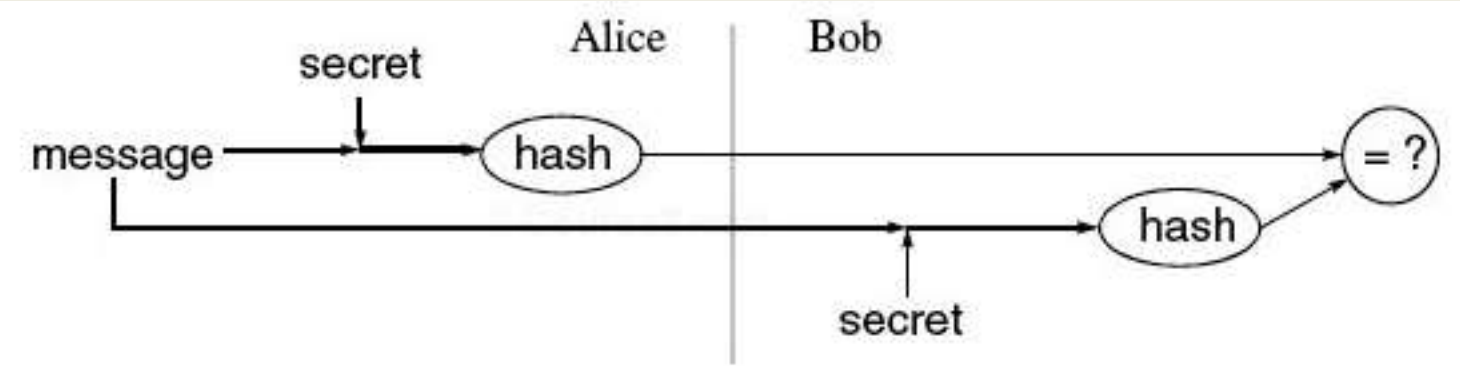
Asymmetry:
Signature can only be *generated* by owner/knower of private key
Signature can be *verified* by anyone via public key

Non-repudiation:
Sender cannot prove message (signature) was not sent
Key may have been stolen



Message Integrity

- Keyed hash, shared secret

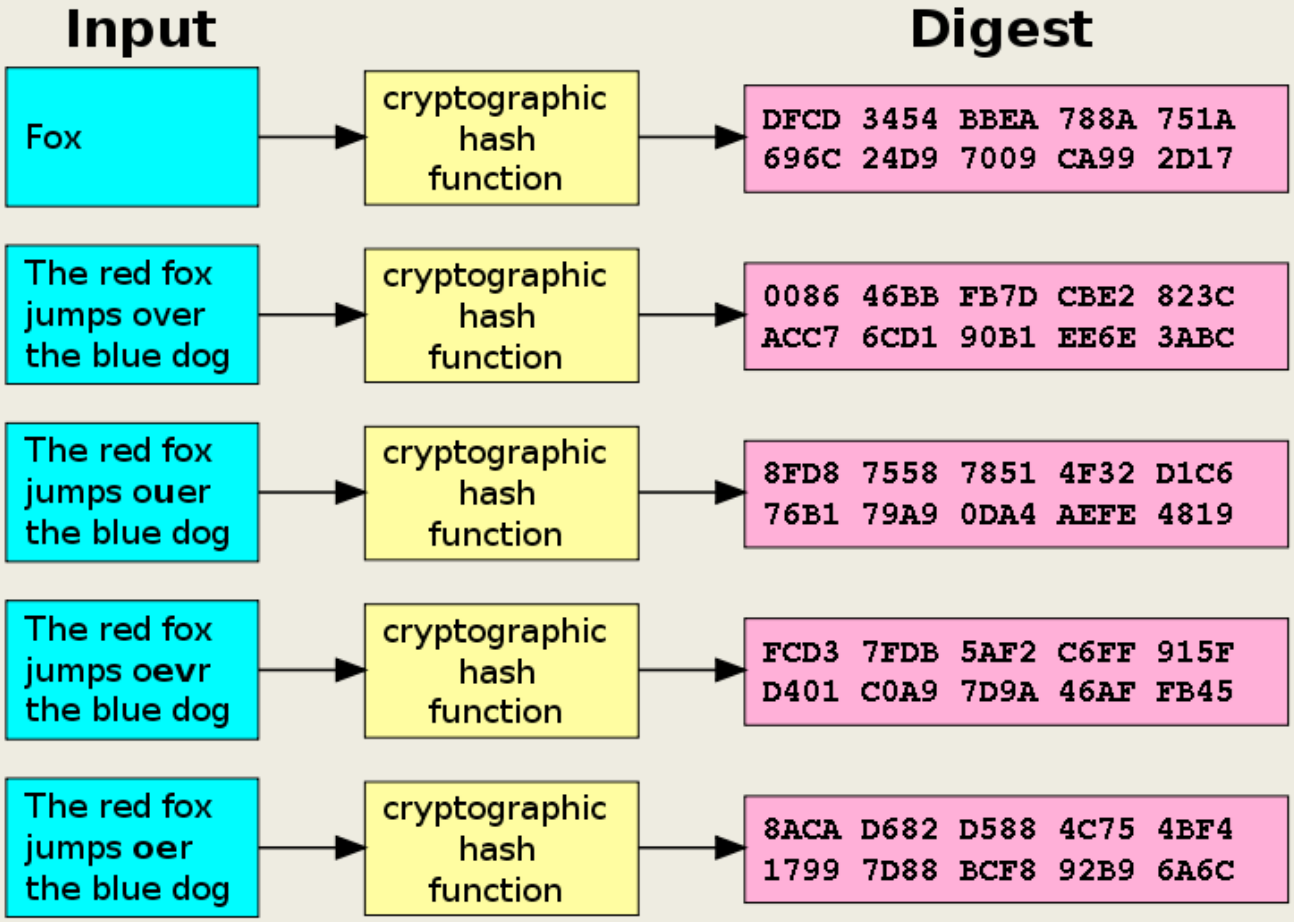


Hash Algorithms

- Message digests / one-way transformations
  - easy to compute a hash value for any given message
  - infeasible to find a message that has a given hash
  - infeasible to modify a message without hash being changed
  - infeasible to find two different messages with the same hash

http://en.wikipedia.org/wiki/Cryptographic_hash_function

Hash Algorithms



http://en.wikipedia.org/wiki/Cryptographic_hash_function

An example

This example uses the common unix utility "**md5sum**", which hashes the data on stdin to a 128 bit hash, displayed as 32 hex digits.

Assume the password is "mysecretpass" and both the client and the server know this.

The client connects to the server.

The server makes up some random data, say "sldkfjdslfkjweifj".
The server sends this data to client.

The client concatenates the random data with the password, resulting in
"sldkfjdslfkjweifjmysecretpass"

The client computes the MD5 hash of this value:

```
$ echo sldkfjdslfkjweifjmysecretpass | md5sum
4fab7ebffd7ef35d88494edb647bac37
```

The client sends "4fab7ebffd7ef35d88494edb647bac37" to the server.

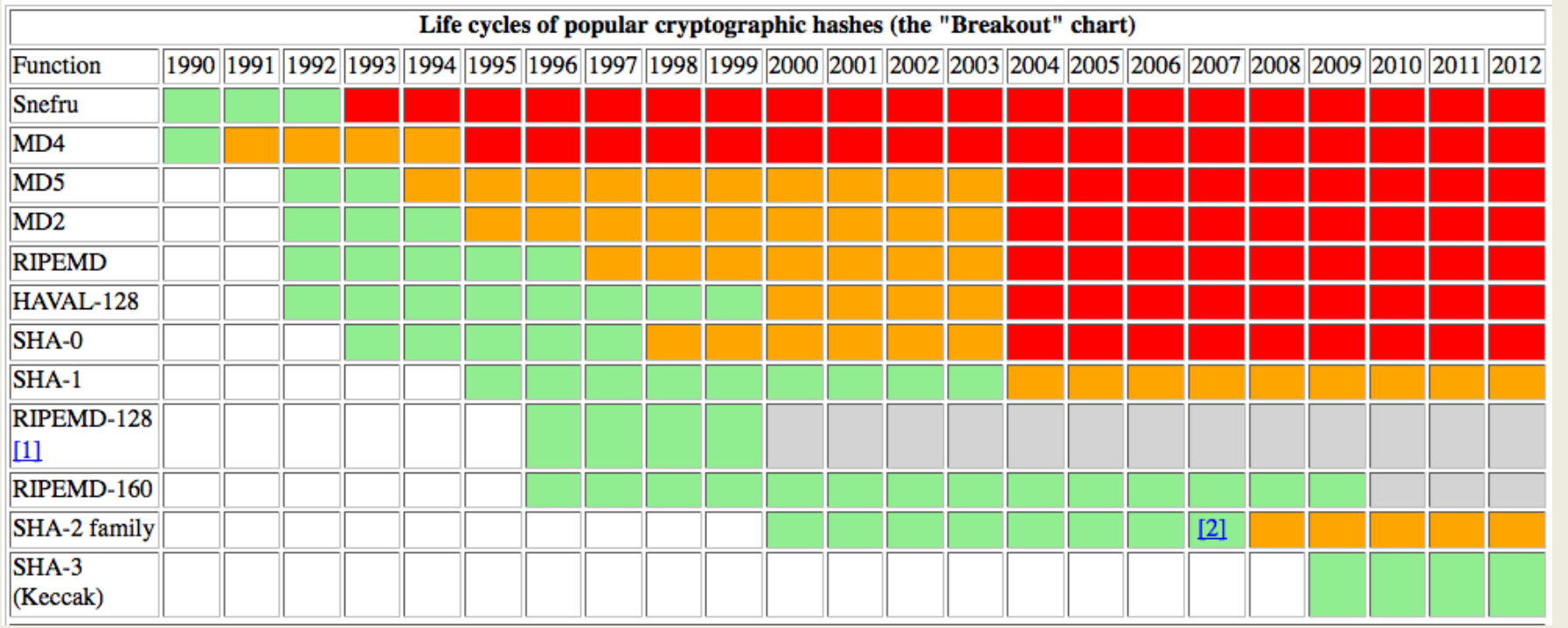
[The server confirms that it gets the same value when it runs echo "sldkfjdslfkjweifjmysecretpass" | md5sum]

http://www.hcsw.org/reading/chalresp.txt

md5sum on Linux

```
[joemcc@uw1-320-17 ~]$ cat > test1.txt
Testing 1, 2, 3
[joemcc@uw1-320-17 ~]$ md5sum test1.txt
6a8c5c1973dd8ed2df1260297964cd64  test1.txt
[joemcc@uw1-320-17 ~]$ cp test1.txt test2.txt
[joemcc@uw1-320-17 ~]$ md5sum test2.txt
6a8c5c1973dd8ed2df1260297964cd64  test2.txt
[joemcc@uw1-320-17 ~]$ cat > test3.txt
Testing 1, 2, 4
[joemcc@uw1-320-17 ~]$ md5sum test3.txt
5e361a608a1f63b154f259dba0f452dc  test3.txt
[joemcc@uw1-320-17 ~]$ echo "Testing 1, 2, 3" | md5sum
6a8c5c1973dd8ed2df1260297964cd64  -
[joemcc@uw1-320-17 ~]$
```

Cryptographic Hash Lifecycle



Life cycles of popular cryptographic hashes (the "Breakout" chart)

Function	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012
Snefru																							
MD4																							
MD5																							
MD2																							
RIPEMD																							
HAVAL-128																							
SHA-0																							
SHA-1																							
RIPEMD-128 [1]																							
RIPEMD-160																							
SHA-2 family																		[2]					
SHA-3 (Keccak)																							

http://valerieaurora.org/hash.html

[via http://www.schneier.com/blog/archives/2011/06/the_life_cycle.html]

Password Hashing

- Only hashes (not passwords) are stored
- Passwords can still be guessed (dictionary)

http://nakedsecurity.sophos.com/2012/06/06/linkedin-confirms-hack-over-60-of-stolen-passwords-already-cracked/