Applied Cryptography

Marcelo Martins
linkedin.com/in/marcelomartins

Agenda

- What is cryptography?
- Hash
- Symmetric cryptography
- Asymmetric cryptography
- Public Key Infrastructure (PKI)
- Attacks
- Steganography
- Best practices: domestic use
- References

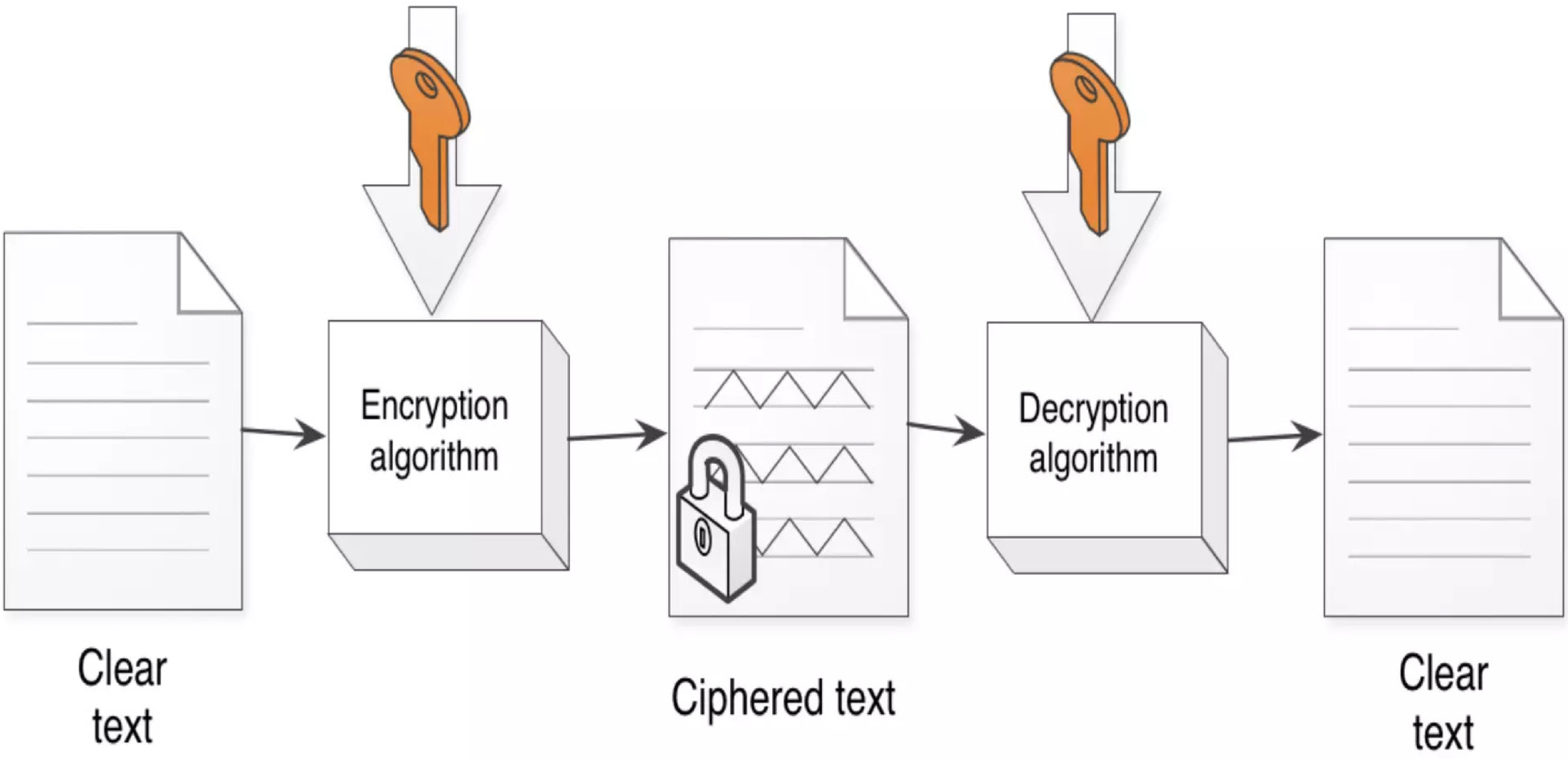
What is cryptography?

- Cryptography or cryptology (from Greek kryptós, "hidden, secret"; and graphein, "writing") is the practice and study of techniques for secure communication in the presence of third parties called intruders or adversaries

- The conversion of information from a readable state to apparent nonsense is called encryption

- Decryption is the reverse, in other words, moving from the unintelligible ciphertext back to plaintext

What is cryptography?



What is cryptography?

- Caesar Crypt or Shift Crypt
  - Used by Julius Caesar to exchange messages with his generals in the field
  - It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet
  - For example, with a right shift of 3

```
Clear:    ABCDEFGHIJKLMNOPQRSTUVWXYZ
Ciphered: DEFGHIJKLMNOPQRSTUVWXYZABC

the quick brown fox jumps over the lazy dog
WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ
```

What is cryptography?

- Caesar Crypt or Shift Crypt
  - The attacks
    - The attacker makes some replacements even not knowing if Caesar Shift was used
    - The attacker knows that Caesar Shift was used, but doesn't know the number of letters to be incremented

What is cryptography?

- Attacks at Caesar Crypt or Shift Crypt

Decrement	Candidates
Original	exxego ex srgi
1	dwwdfn dw rqfh
2	cvvcem cv qpeg
3	buubdl bu podf
4	attack at once
5	zsszbj zs nmbd
6	yrryai yr mlac
...	
23	haahjr ha vujl
24	gzzgiq gz utik
25	fyyfhp fy tshj

What is cryptography?

- Information Security Objectives
  - Confidentiality
    - Only those authorized shall access the information
  - Integrity
    - The information shall not be modified by non-authorized parties
  - Authenticity and Non-repudiation
    - It shall be possible to assure the sender sent that message and that it cannot be refuted

What is cryptography?

- Cryptography ≠ Steganography!
  - Steganography (from Greek steganos, “covered”; and graphein, “writing”) is the practice of concealing the existence of a message inside another message
- While cryptography conceals the meaning of the message, steganography hides the message itself
  - In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size.

Agenda

- What is cryptography?
- Hash
- Symmetric cryptography
- Asymmetric cryptography
- Public Key Infrastructure (PKI)
- Attacks
- Steganography
- Best practices: domestic use
- References

Hash

- It is a sequence of bits generated from a dispersion algorithm, usually represented in hexadecimal format, that allows visualization in letters and numbers (0 … 9, A … F), representing ½ byte each

- Hashing is the transformation of a large amount of information in a small amount of information

- It is not possible to revert a one way hash and get the original message back

Hash

- Hash function example

$$h(m) = h^{-1} \oplus (m \otimes p)$$

$$h(m) = \sum_{i=0}^{|m|} m_i \otimes p_i$$

$$h(m) = h^{-1} \otimes (m \otimes p)$$

$$h(m) = \prod_{i=0}^{|m|} m_i \otimes p_i$$

Hash

- Most used hash functions

Name	Output bits	Created	Collision*	Vulnerable*
MD4	128	1990	Yes	Yes
MD5	128	1991	Yes	Yes
SHA-1	160	1995	Yes	Yes
SHA-224	224	2001	No	No
SHA-256	256	2001	No	No
SHA-384	384	2001	No	No
SHA-512	512	2001	No	No
WHIRLPOOL	512	2004	No	No

Hash

- SHA-512 example

  - SHA512("The quick brown fox jumps over the lazy dog")

    07e547d9 586f6a73 f73fbac0 435ed769 51218fb7
    d0c8d788 a309d785 436bbb64 2e93a252 a954f239
    12547d1e 8a3b5ed6 e1bfd709 7821233f a0538f3d
    b854fee6

  - SHA512("The quick brown fox jumps over the lazy cog")

    3eeee1d0 e11733ef 152a6c29 503b3ae2 0c4f1f3c
    da4cb26f 1bc1a41f 91c7fe4a b3bd8649 4049e201
    c4bd5155 f31ecb7a 3c860684 3c4cc8df cab7da11
    c8ae5045

Hash

- SHA-1 example

  - SHA1("The quick brown fox jumps over the lazy dog")

    `2fd4e1c6 7a2d28fc ed849ee1 bb76e739 1b93eb12`

  - SHA1("The quick brown fox jumps over the lazy **c**og")

    `de9f2c7f d25e1b3a fad3e85a 0bd17d9b 100db4b3`

  - SHA1("")

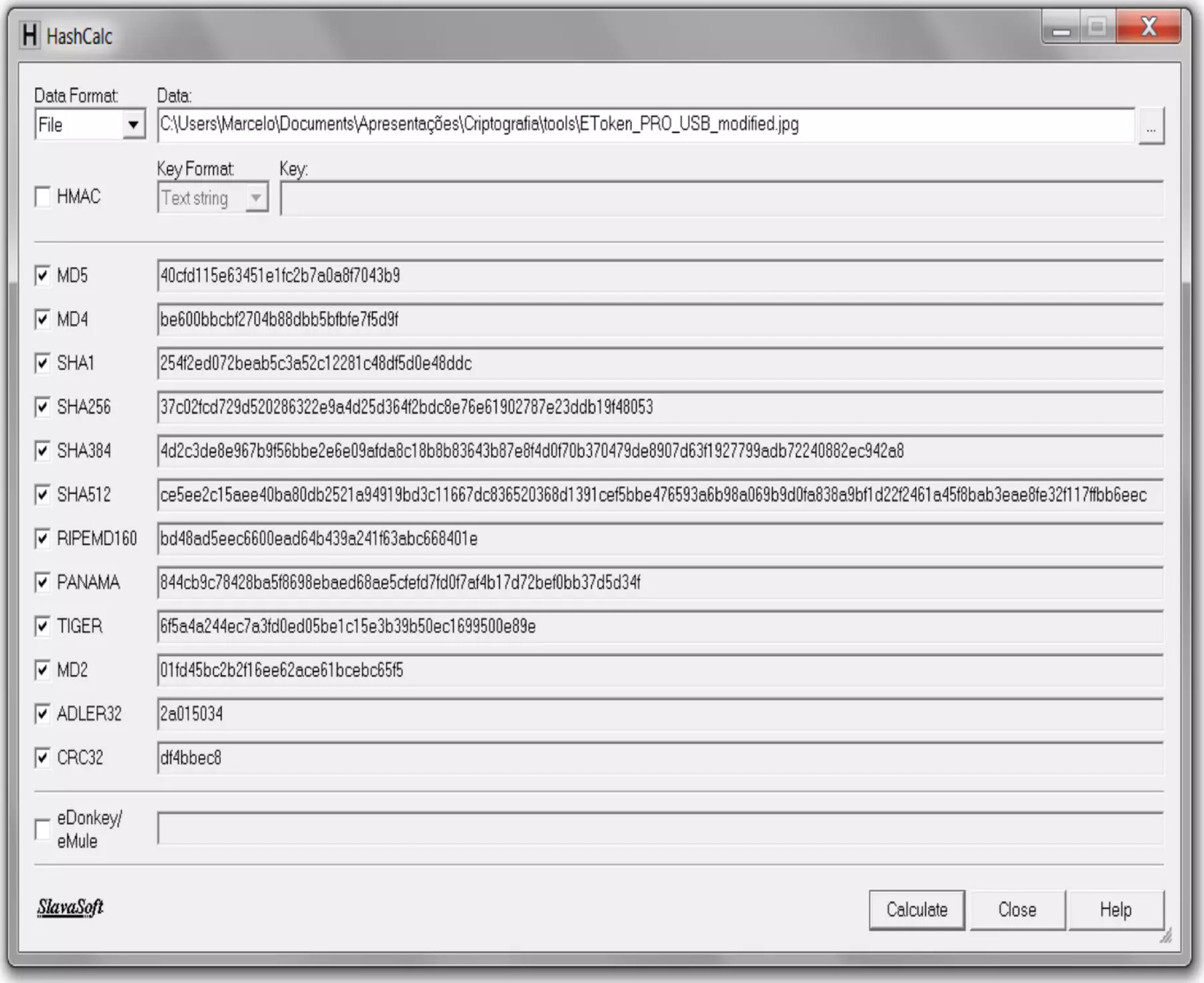
    `da39a3ee 5e6b4b0d 3255bfef 95601890 afd80709`

Changing one bit produces a totally different hash

Hash

- Independent applications
  - HashTab (free for private use, Windows)
    - http://implbits.com/products/hashtab/
    - MD5, HAVAL, MD2, SHA (1, 256, 384, 512)
  - File Checksum Tool (free for private and commercial use, Windows)
    - http://www.krylack.com/file-checksum-tool/
    - Adler32, CRC32, MD2, MD4, MD5, RIPEMD (128, 256, 320), SHA (1, 256, 384, 512), Tiger e Whirlpool
  - HashCalc (free for private and commercial use, Windows)
    - http://www.slavasoft.com/hashcalc/
    - Adler32, CRC32, MD2, MD4, MD5, RIPEMD (128, 256, 320), SHA (1, 256, 384, 512), Tiger e Panama

Hash


HashCalc
Data Format:
File
Data:
C:\Users\Marcelo\Documents\Apresentações\Criptografia\tools\EToken_PRO_USB_modified.jpg
...
HMAC
Key Format:
Text string
Key:
MD5
40cfd115e63451e1fc2b7a0a8f7043b9
MD4
be600bbcbf2704b88dbb5bfbfe7f5d9f
SHA1
254f2ed072beab5c3a52c12281c48df5d0e48ddc
SHA256
37c02fcd729d520286322e9a4d25d364f2bdc8e76e61902787e23ddb19f48053
SHA384
4d2c3de8e967b9f56bbe2e6e09afda8c18b8b83643b87e8f4d0f70b370479de8907d63f1927799adb72240882ec942a8
SHA512
ce5ee2c15aee40ba80db2521a94919bd3c11667dc836520368d1391cef5bbe476593a6b98a069b9d0fa838a9bf1d22f2461a45f8bab3eae8fe32f117ffbb6eec
RIPEMD160
bd48ad5eec6600ead64b439a241f63abc668401e
PANAMA
844cb9c78428ba5f8698ebaed68ae5cfefd7fd0f7af4b17d72bef0bb37d5d34f
TIGER
6f5a4a244ec7a3fd0ed05be1c15e3b39b50ec1699500e89e
MD2
01fd45bc2b2f16ee62ace61bcebc65f5
ADLER32
2a015034
CRC32
df4bbec8
eDonkey/
eMule
SlavaSoft
Calculate
Close
Help

Hash

- File integrity verification
  - Monitors or verifies changes in files

  - Tripwire (Windows/Unix, commercial)
  - AIDE (Unix, freeware)
  - Yafic (Unix, freeware)
  - AFICK (Windows/Unix, freeware)
  - nCircle File Integrity Monitoring (Win/Unix, commercial)
  - Advanced Checksum Verifier (Windows, commercial)
  - Slavasoft FSUM (Windows, freeware)
  - Chkrootkit (Unix, freeware)
  - Samhain (Unix, freeware)

Hash

- File integrity verification
  - macOS

    - `$ shasum test.jpg` (defaults to SHA1)
    - `a9b602d039d302867df743ab7dd056e3644bd208 test.jpg`

    - `$ shasum -a512 test.jpg`
    - `e0d4128da441d17ac02c039878a4ac1fae437656b51807b85c0238deefcfe96bebaedc285edbc3e5d4e18b315b0d1bce7a47dce130b39645d2372e6003c19fc4  test.jpg`

Agenda

- What is cryptography?
- Hash
- Symmetric cryptography
- Asymmetric cryptography
- Public Key Infrastructure (PKI)
- Attacks
- Steganography
- Best practices: domestic use

Symmetric Cryptography

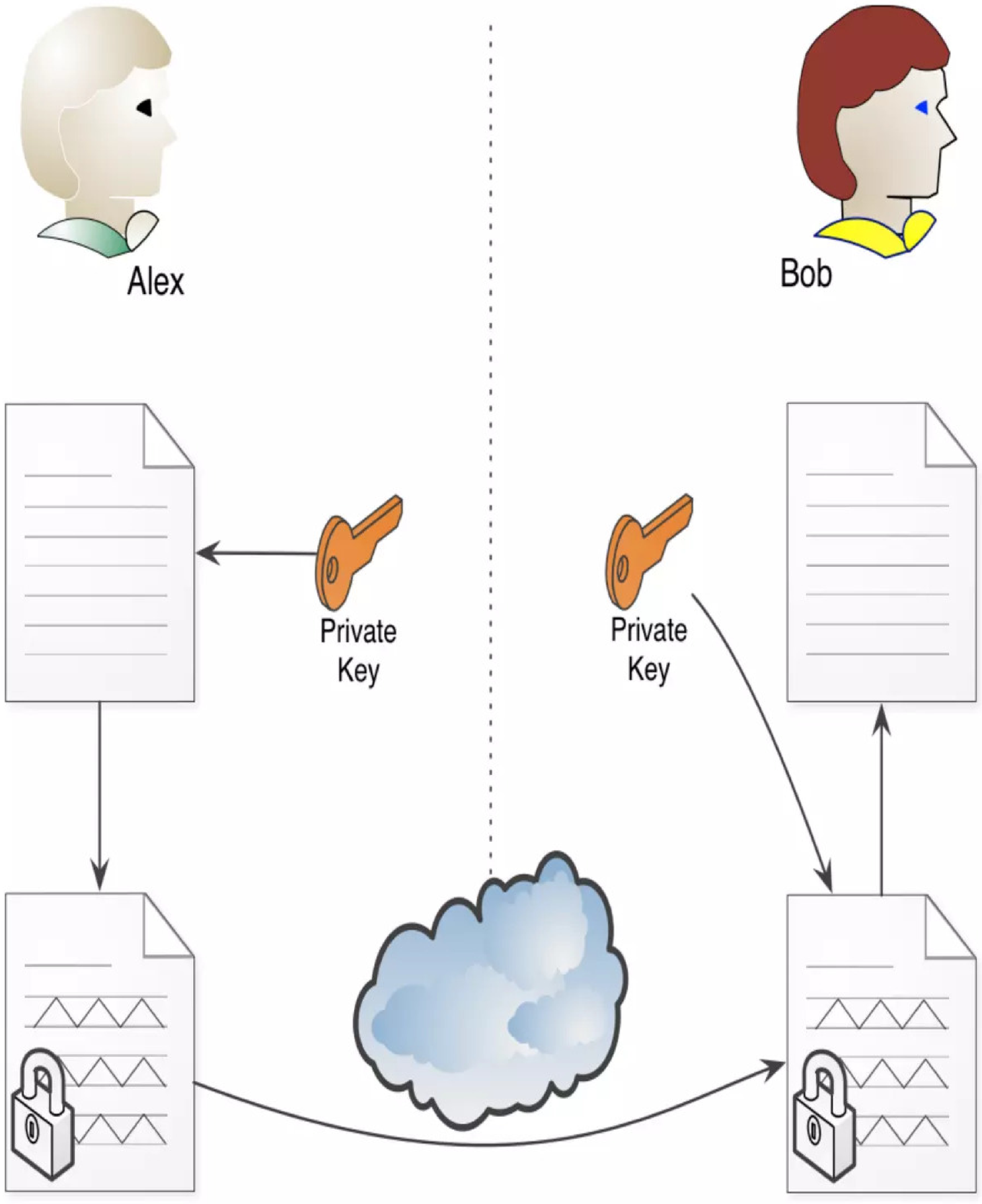
- Only one key is used to encrypt and decrypt the message

- This key is called symmetric key, private key, or shared key

- Most used algorithms are: AES, Blowfish, DES, Triple DES, Serpent, Twofish

DES and 3DES should not be used anymore

Symmetric Cryptography



Symmetric Cryptography

- Symmetric cryptography security depends on
  - Security of the private key
  - Strength of the password of the private key (just like any password)
  - Security when exchanging private keys
  - Security of the cryptographic method (various types of attack)

Symmetric Cryptography

- Problems
  - How to transmit the key to the recipient using insecure media? Phone? Smoke sign? Unencrypted E-mail?
  - For 5 people to exchange messages using symmetric keys, how many keys would be necessary?

- Necessary keys

$$\frac{n*(n-1)}{2}$$

  - 5 users

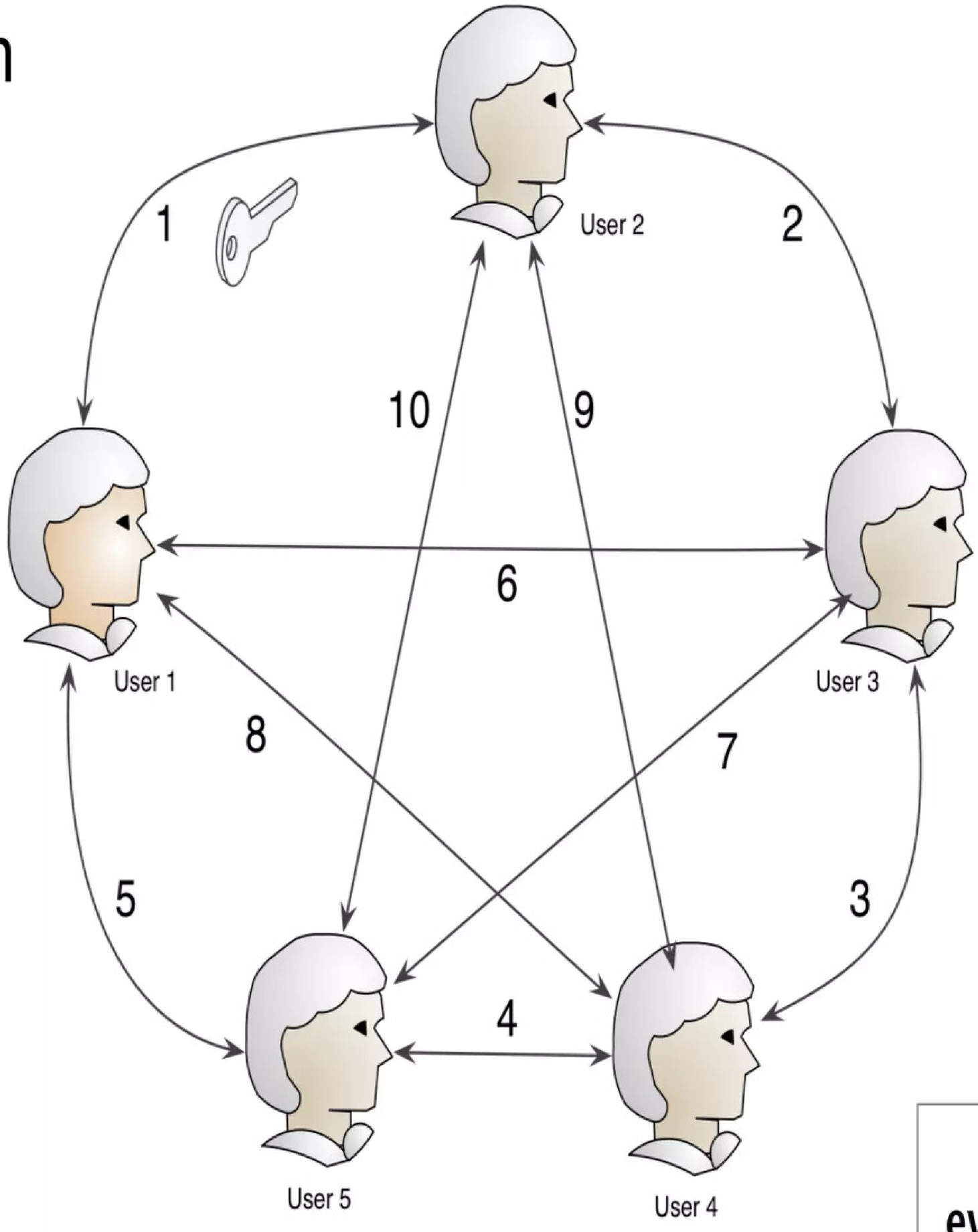
$$\frac{5*(5-1)}{2}=10$$

  - 20 users

$$\frac{20*(20-1)}{2}=190$$

Symmetric Cryptography

- Problem



One key for every two users

Agenda

- What is cryptography?
- Hash
- Symmetric cryptography
- Asymmetric cryptography
- Public Key Infrastructure (PKI)
- Attacks
- Steganography
- Best practices: domestic use
- References

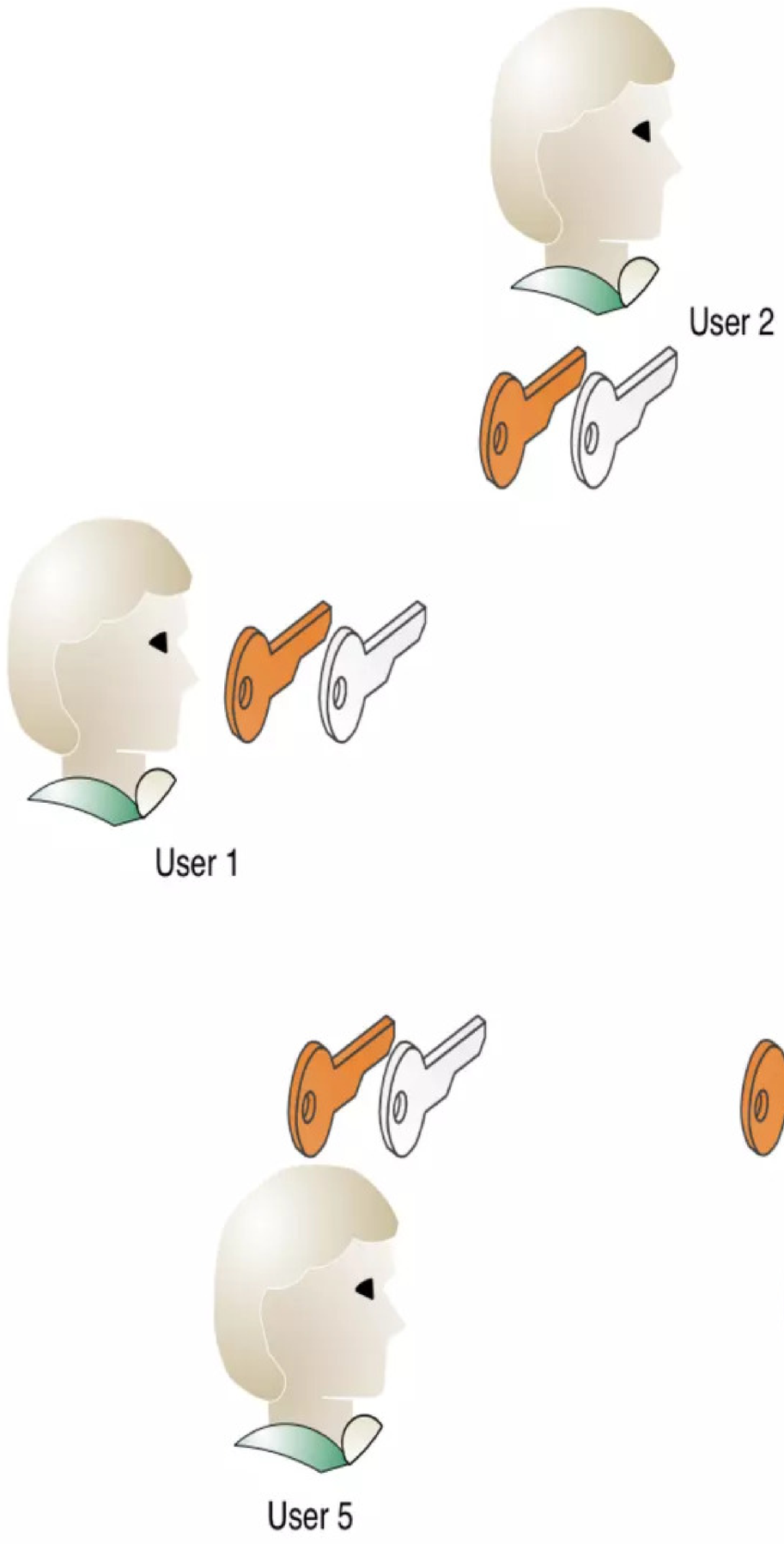
Asymmetric Cryptography

- Every user possesses a pair of keys: the public key and the private key

- The public key is available for everyone and the private key is kept secret

- What is done with one key is “undone” with the other, from the same pair

Asymmetric Cryptography



Every user has a pair of keys

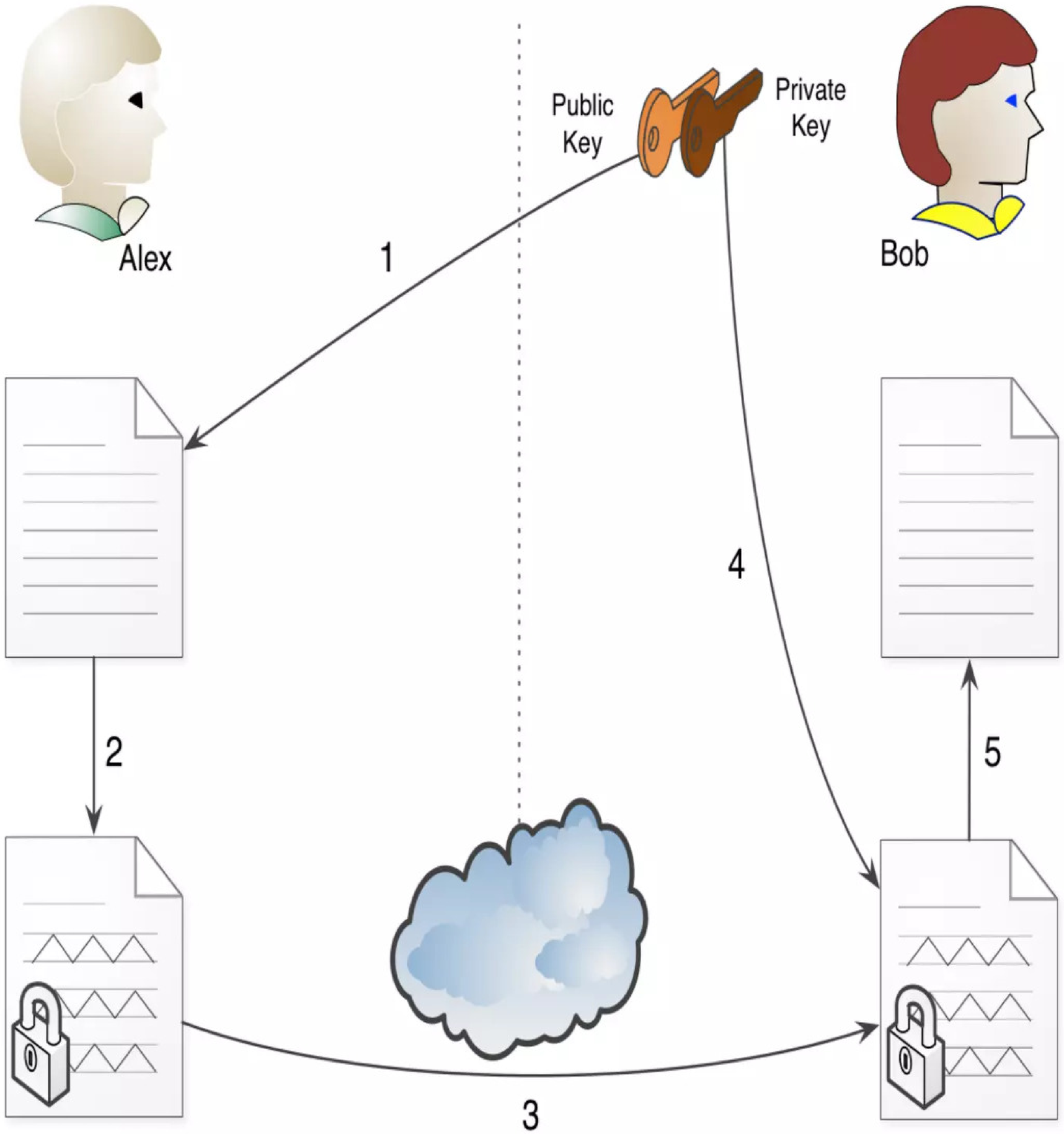
Asymmetric Cryptography

- Asymmetric key example (PGP)

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.3ia

mQCNAzR9bCAAAAEEAMbG7w5W0EJ/L4oAlOgsWwzVmS145eUF6knmOx/UUNBMdNRD
DGgJO3z3aAq4nE4yR+3hSO2auiZlB604e006SPV3ZuLpCHuzaet59dHJGmT9adgx
DMgaPv5q9CCsZa9B1lhh/SIV2eU4U17FDWfU8QWrimn+nLi/y+kli63F4U8VAAUX
tEJDZW50cm8gZGUgQXRlbmRpbWVudG8gYSBJbmNpZGVudGVzIGRlIFNlZ3VyYW5j
YSA8Y2Fpc0BjYWlzLnJucC5icj6JAJUDBRA0fWwg6SWLrcXhTxUBAU/WBACclOR+
MC4kGNAKsR7+HujUYC90BYB8QVFcg/jYAWXDMGYxFeCtvw9FQDUtk55B+/i7tdaX
2c4kZrlsjoc3dYROdB7OKDwIxw1pxmeuBBkT+I34i47YNcyRB15otW5sS9BJDb4X
jw5YqDDhRCvevUCTczUvAIKSKnzg4Yl1I/JGaQ== =DjRv
-----END PGP PUBLIC KEY BLOCK-----
```

Asymmetric Cryptography



Asymmetric Cryptography

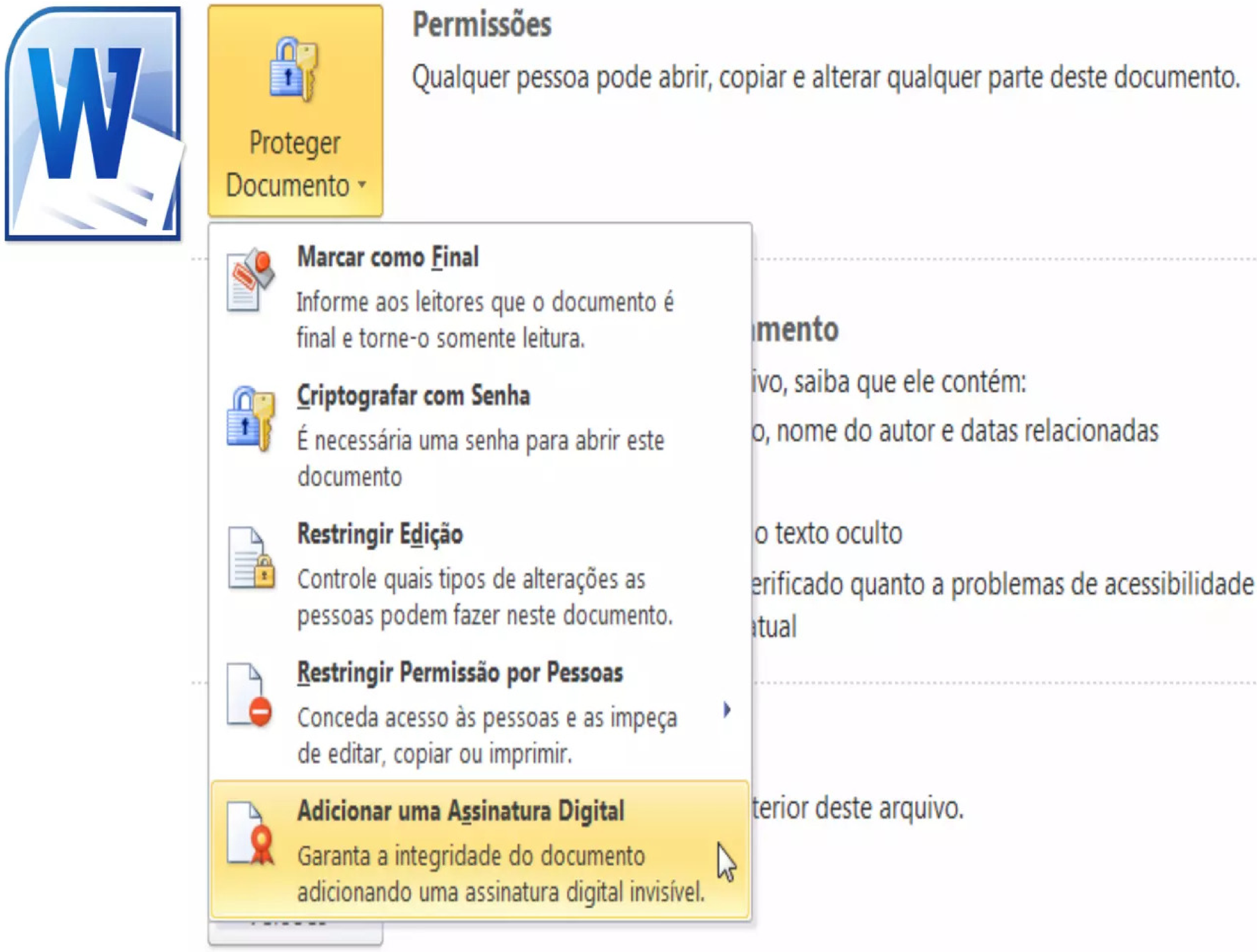
- Encryption and decryption
  1. Public key from the receiver is obtained
  2. The message is encrypted
  3. The message is sent over insecure media
  4. The private key is used to decrypt the message
  5. The original message is recovered

Asymmetric Cryptography

- Digital signature
  - A cryptographic process used to assure integrity, authenticity and non-repudiation of a message or file
  - It is possible to digitally sign e-mails and files in various formats like pdf, doc, ppt, dll, exe, etc.
  - The digital signature, alone, cannot protect the content of the message. It can only assure its integrity and sender

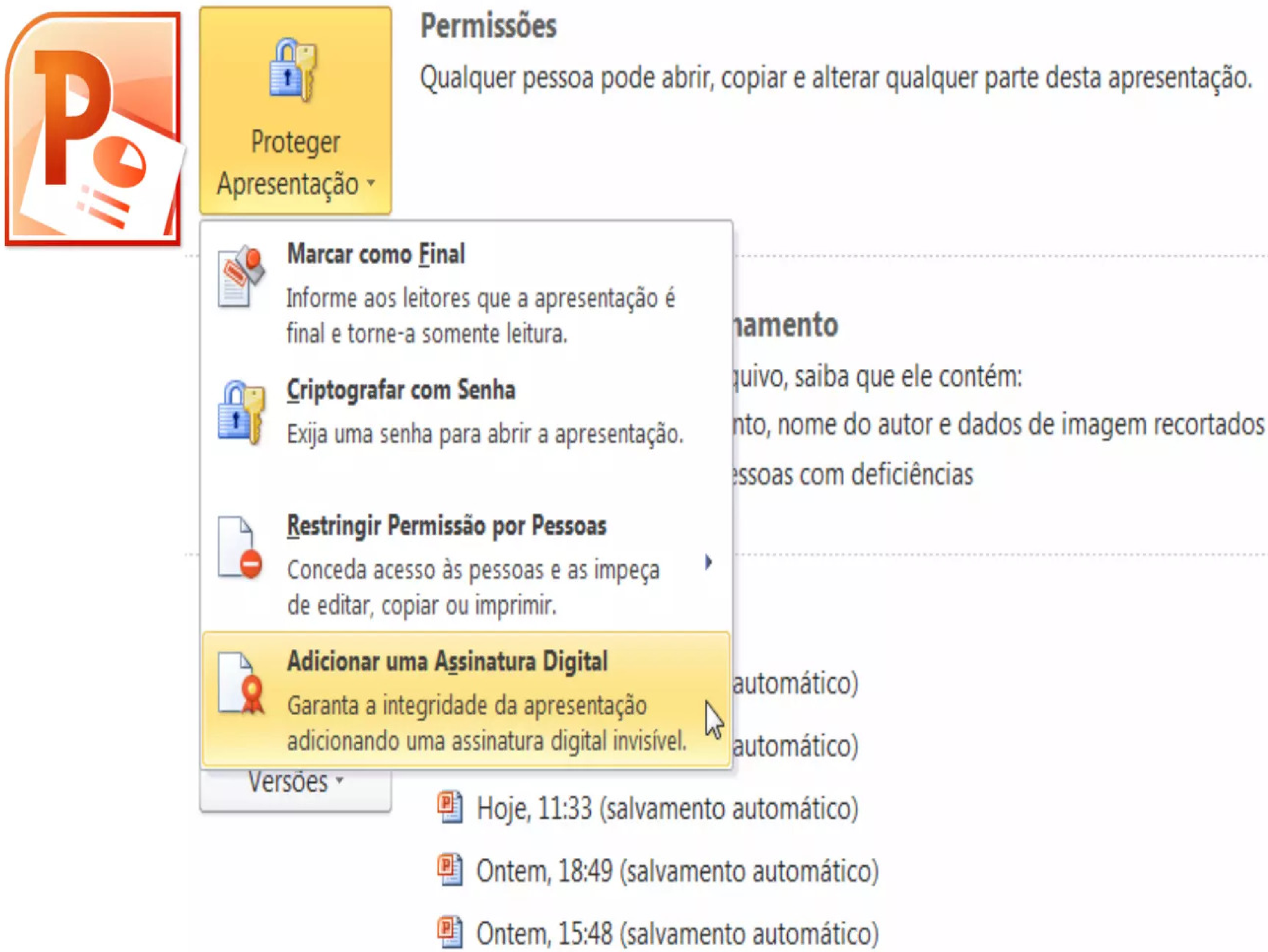
Asymmetric Cryptography

- Digital signature



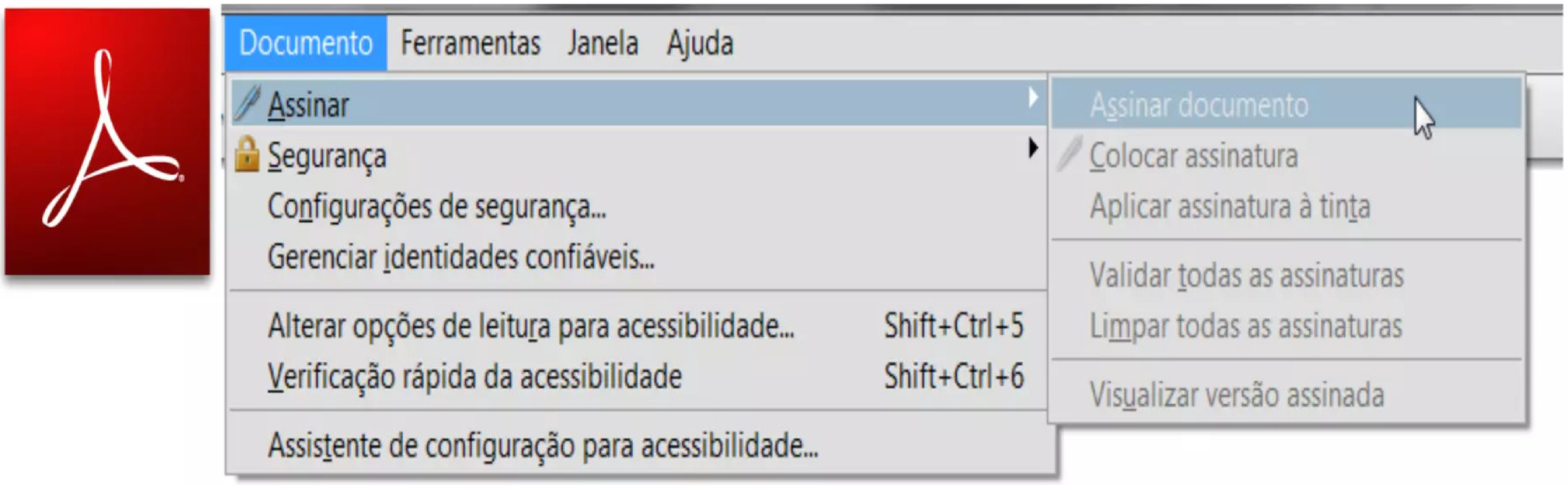
Asymmetric Cryptography

- Digital signature



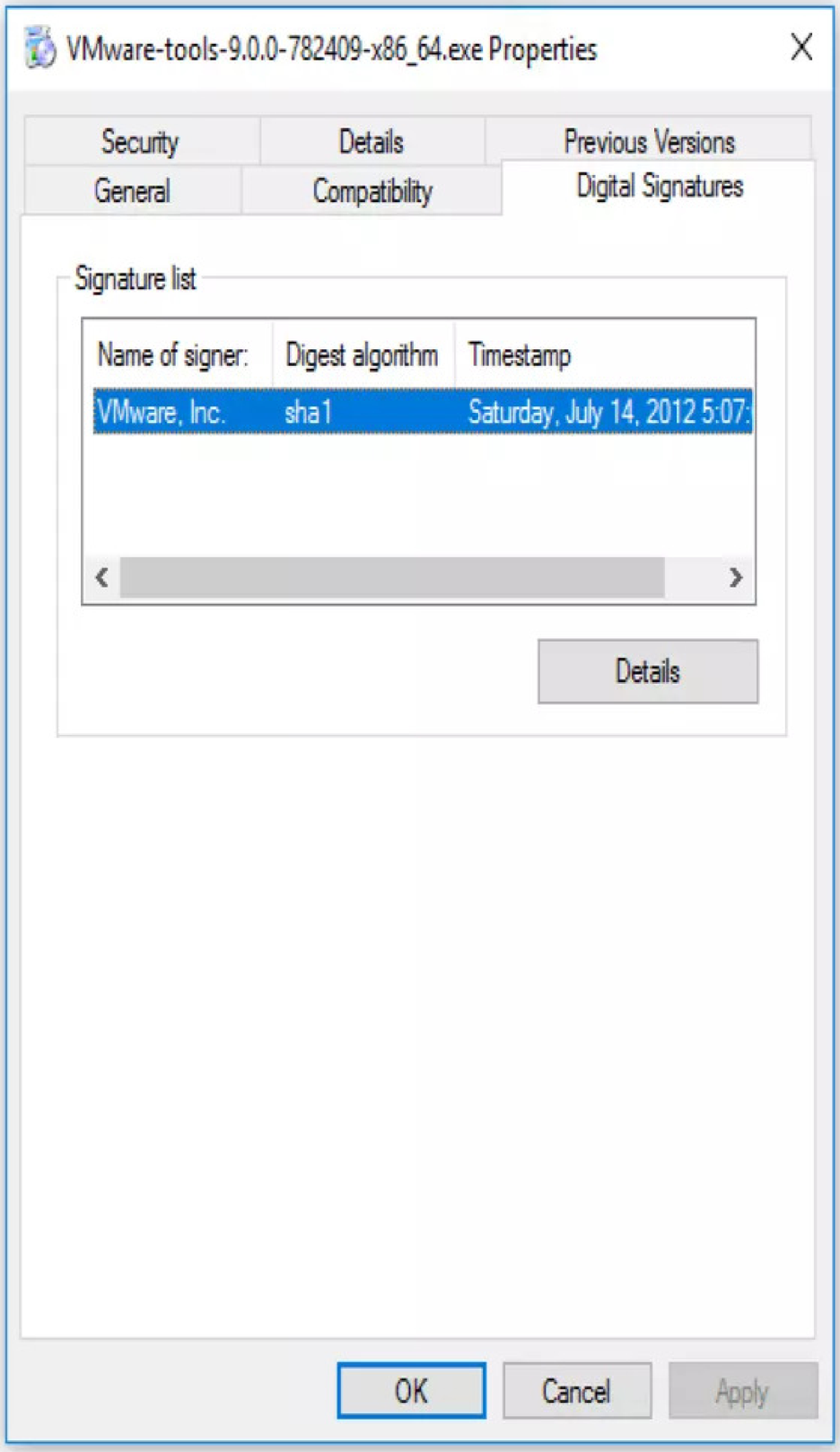
Asymmetric Cryptography

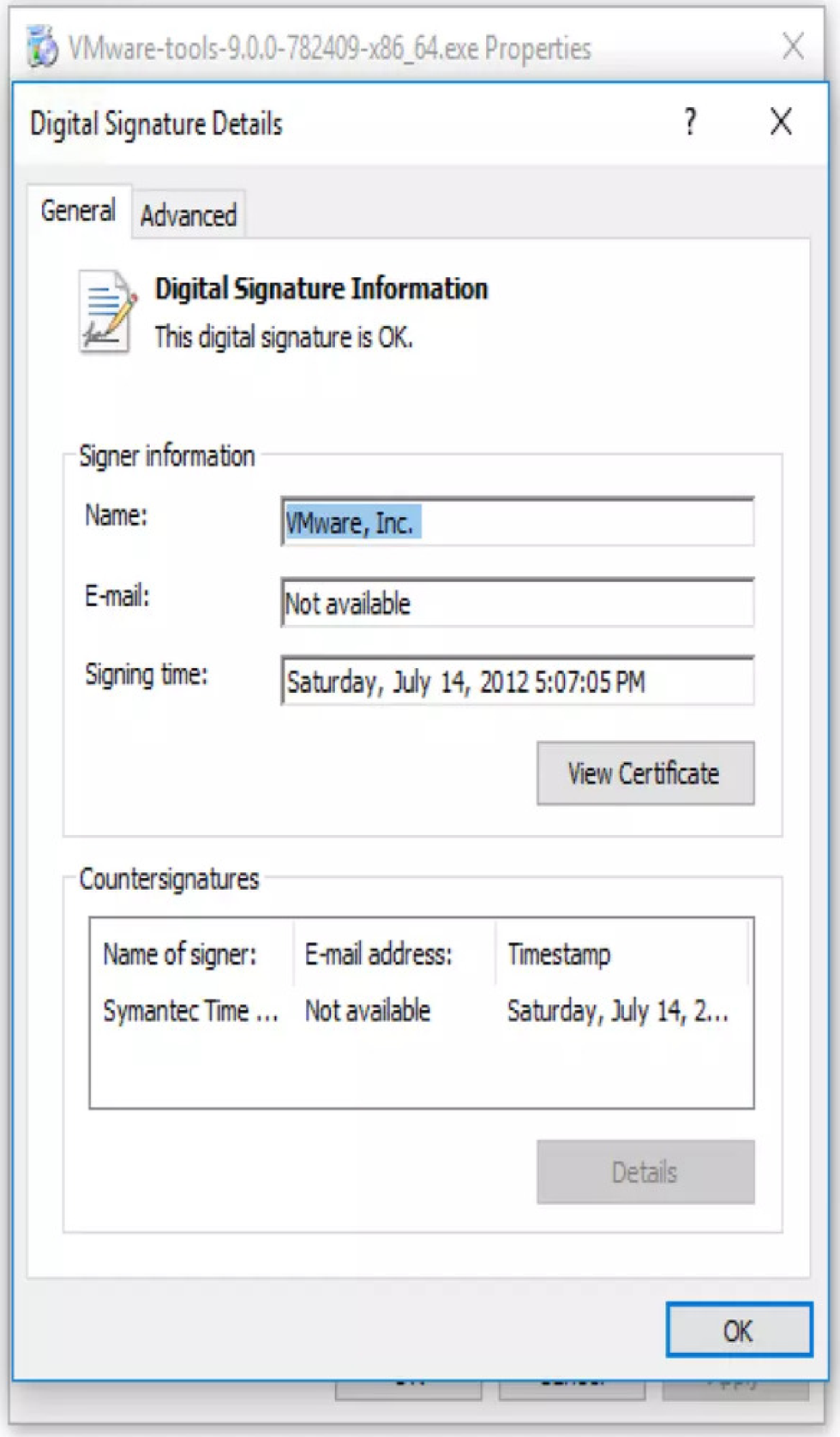
- Digital signature



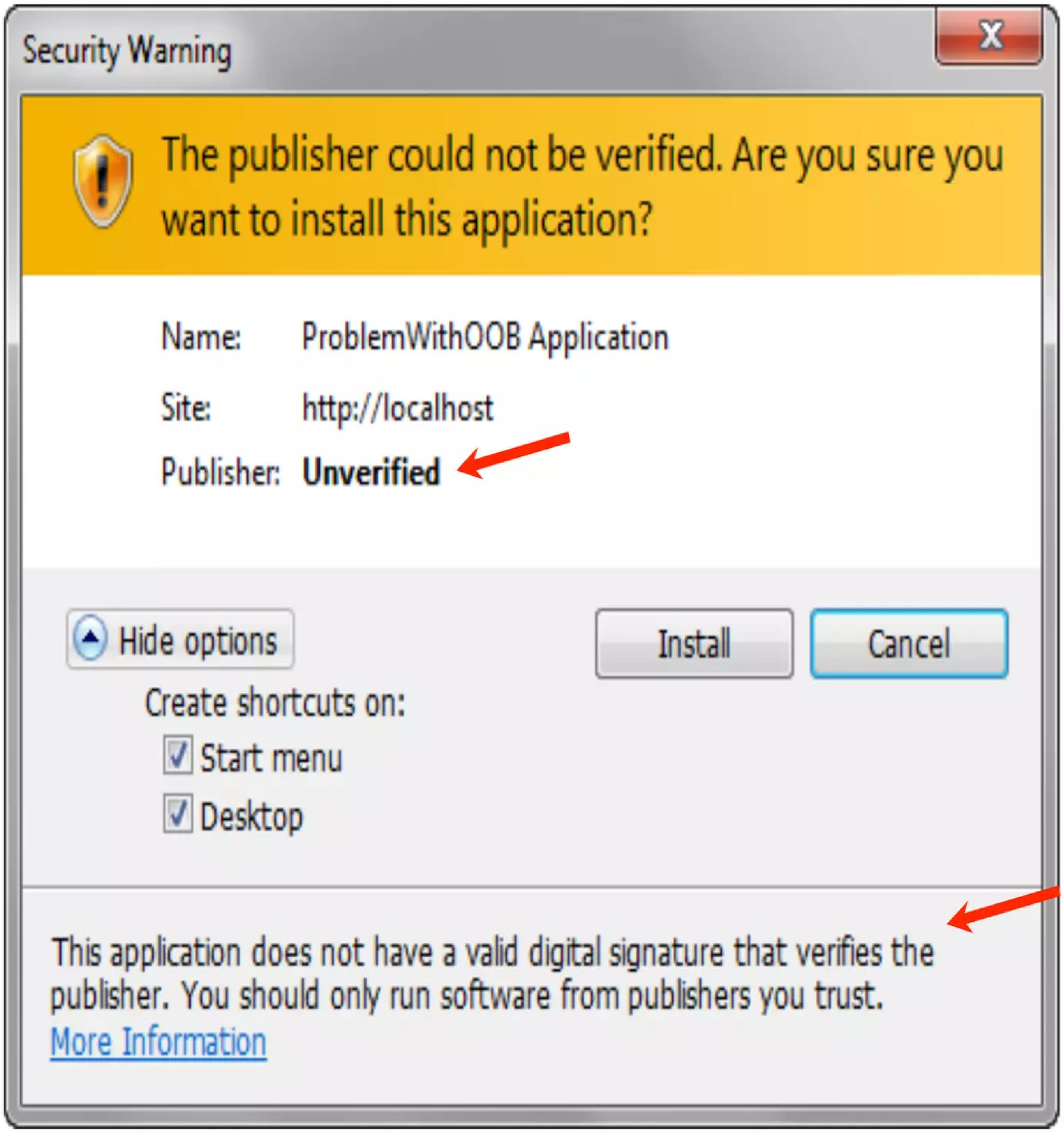
Asymmetric Cryptography

- Digital signature

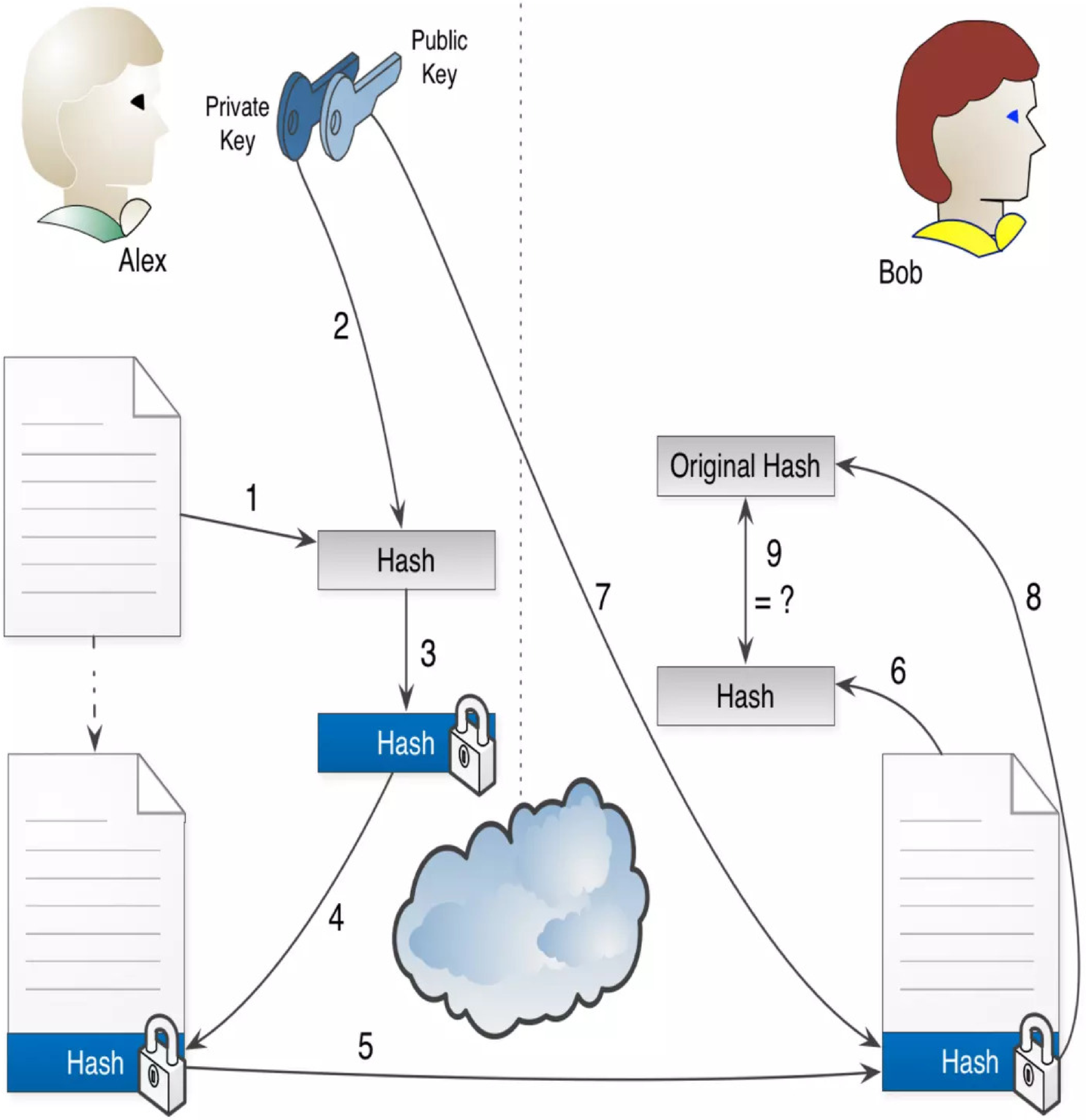




Digital Signature



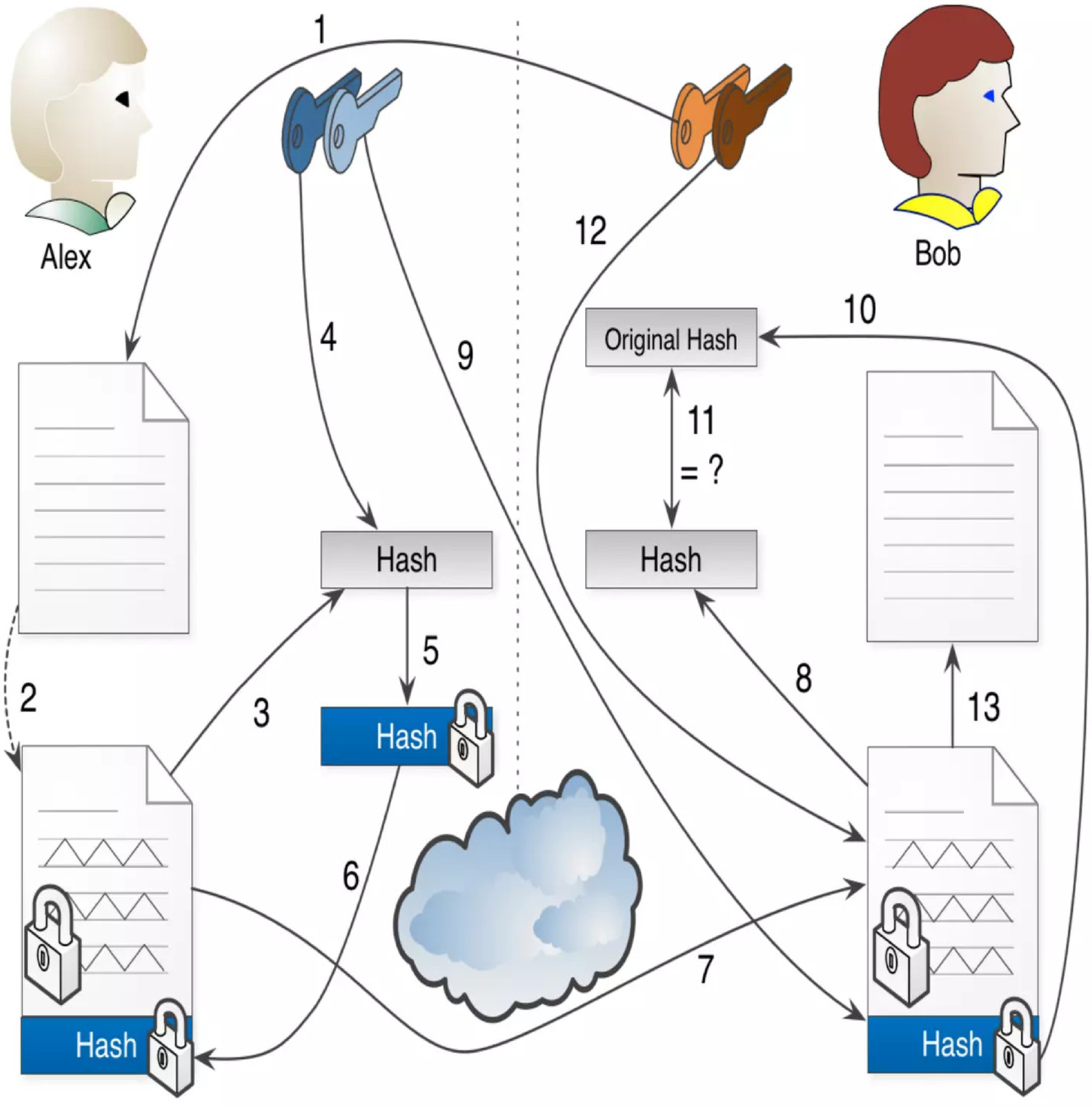
Digital Signature



Digital Signature

- Encryption and decryption
  1. A hash is generated from the message
  2. The hash is encrypted using the sender's private key
  3. The encrypted hash is obtained
  4. The encrypted hash is added to the message
  5. The message is sent over insecure media
  6. Another hash is generated from the message
  7. The sender's public key is used to decrypt the hash previously sent
  8. The original hash is obtained
  9. Comparison between both hashes; if they match the message wasn't tempered with

Encryption + Digital Signature

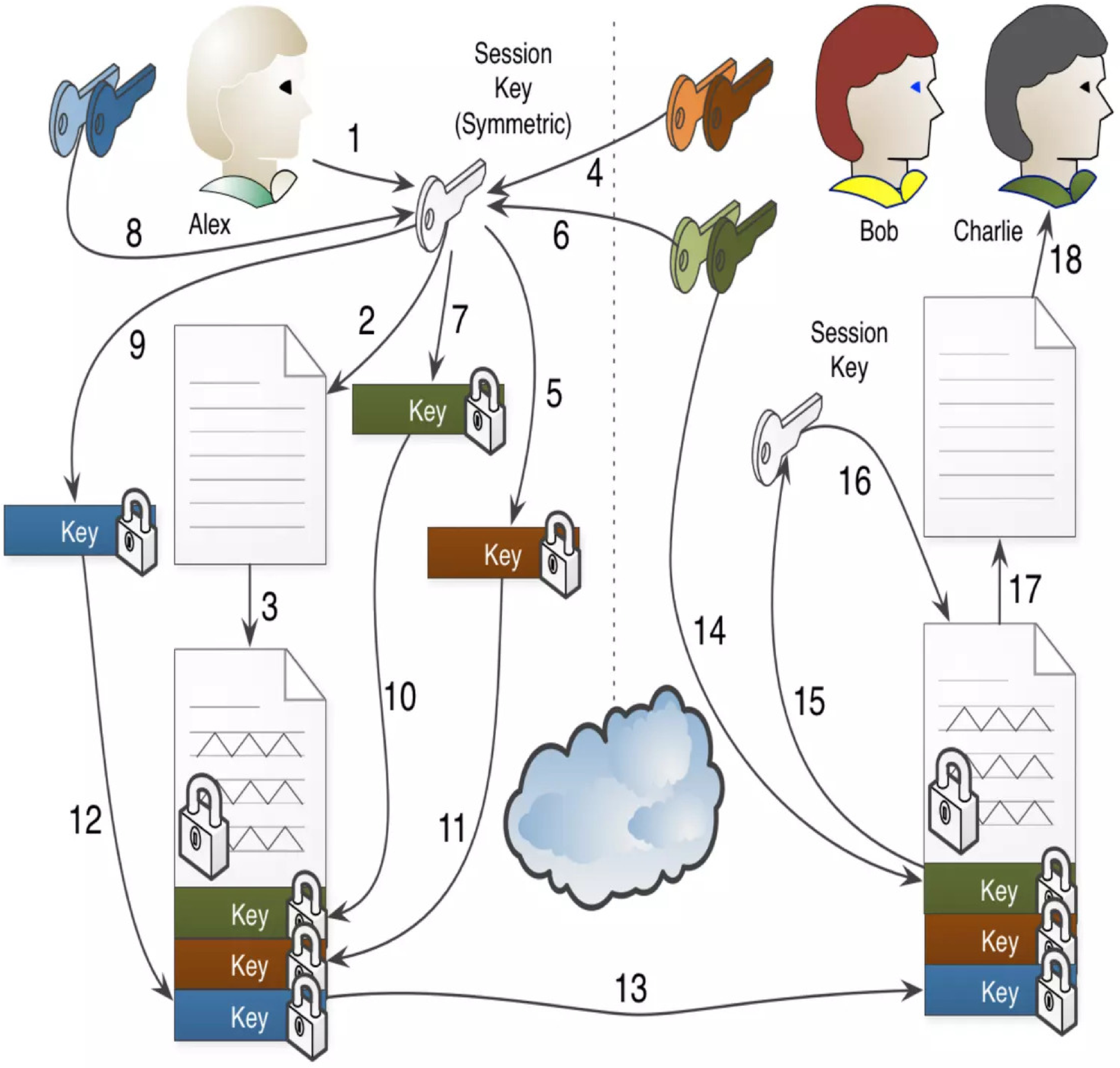


Simplified version

Encryption + Digital Signature

- Encryption and decryption
  1. Public key from the receiver is obtained
  2. The message is encrypted with this public key
  3. A hash is generated from the message
  4. The hash is encrypted using the sender's private key
  5. The encrypted hash is obtained
  6. The encrypted hash is added to the message
  7. The message is sent over insecure media
  8. Another hash is generated from the message
  9. The sender's public key is used to decrypt the hash previously sent
  10. The original hash is obtained
  11. Comparison between both hashes; if they match the message wasn't tempered with
  12. Receiver's private key is used to decrypt the message
  13. The original message is recovered

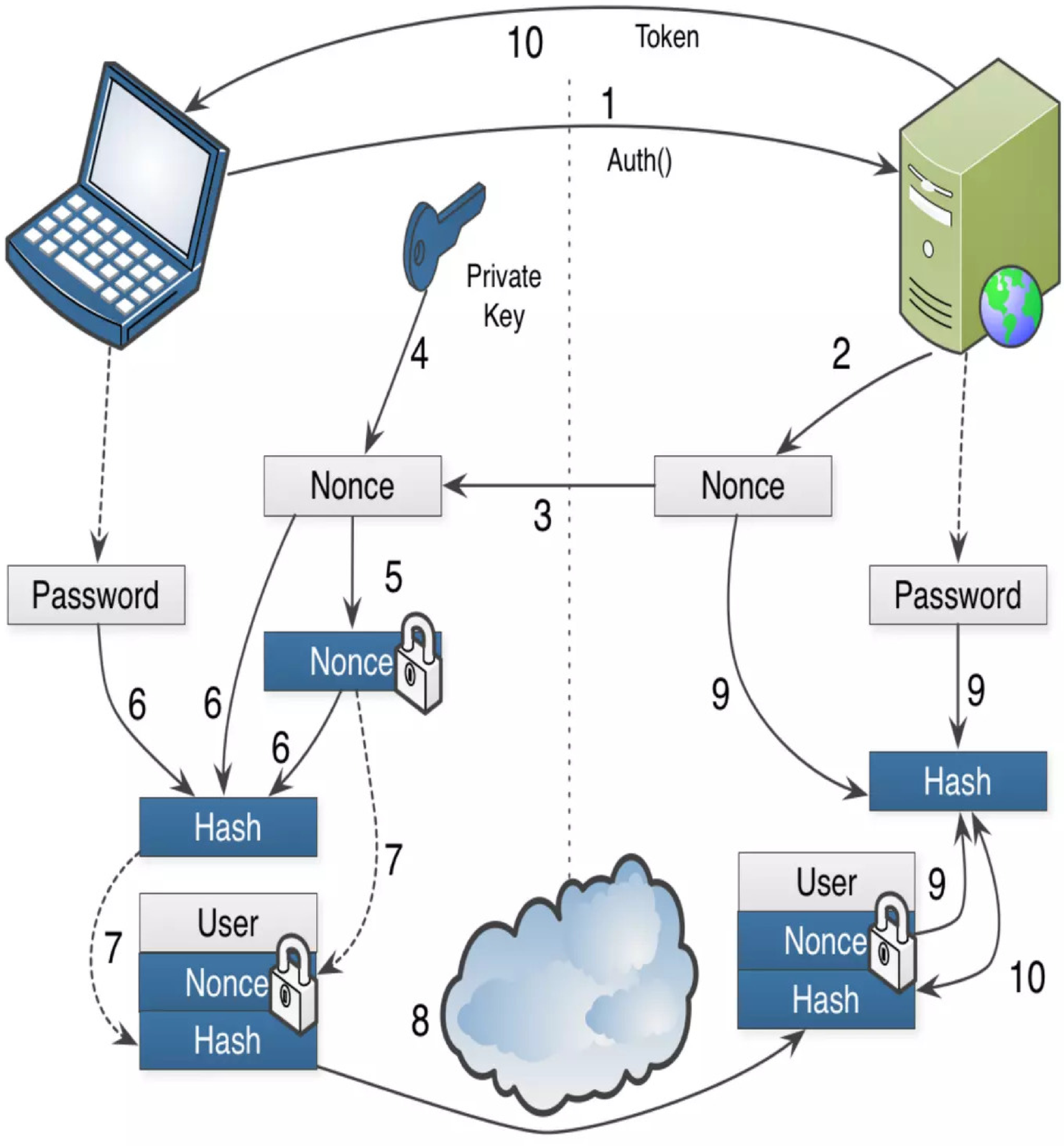
Bulk Encryption



Bulk Encryption

- Encryption and decryption
  1. A symmetric key (session key) is randomly generated
  2. The message is encrypted using the session key
  3. The encrypted message is obtained
  4. Public key from Bob is obtained
  5. A copy of the session key is encrypted using Bob's public key
  6. Public key from Charlie is obtained
  7. A copy of the session key is encrypted using Charlie's public key
  8. Public key from sender is obtained
  9. The session key is encrypted using Sender's public key
  10. Encrypted session key is added to the message
  11. Encrypted session key is added to the message
  12. Encrypted session key is added to the message
  13. The message is sent over insecure media
  14. Charlie uses his private key to obtain the session key
  15. The original key is obtained
  16. The session key is used to decrypt the message
  17. The original message is obtained

Authentication (Nonce)

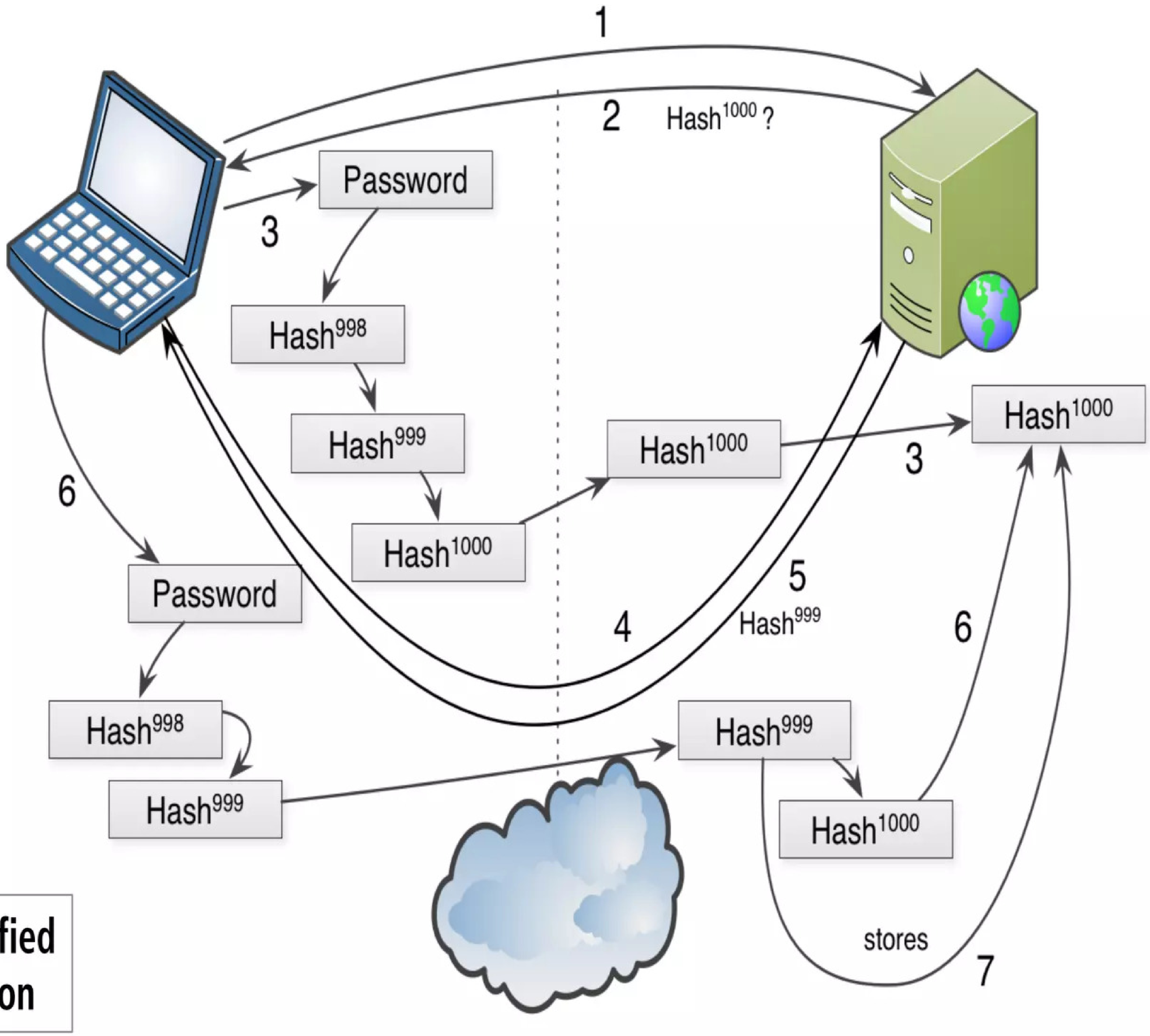


Simplified version

Authentication (Nonce)

- Hashing
  1. Client tries to authenticate
  2. Server generates a Nonce
  3. Nonce is sent to the client
  4. Client uses its private key to encrypt Nonce
  5. Encrypted Nonce is obtained
  6. A hash is generated from Nonce, Encrypted Nonce and client's password
  7. The new Hash is added to a new payload, with the User and the Nonce
  8. The payload is sent to the server
  9. Server generated a hash using original Nonce, client's stored password and the Encrypted Nonce received
  10. Comparison between both Hashes

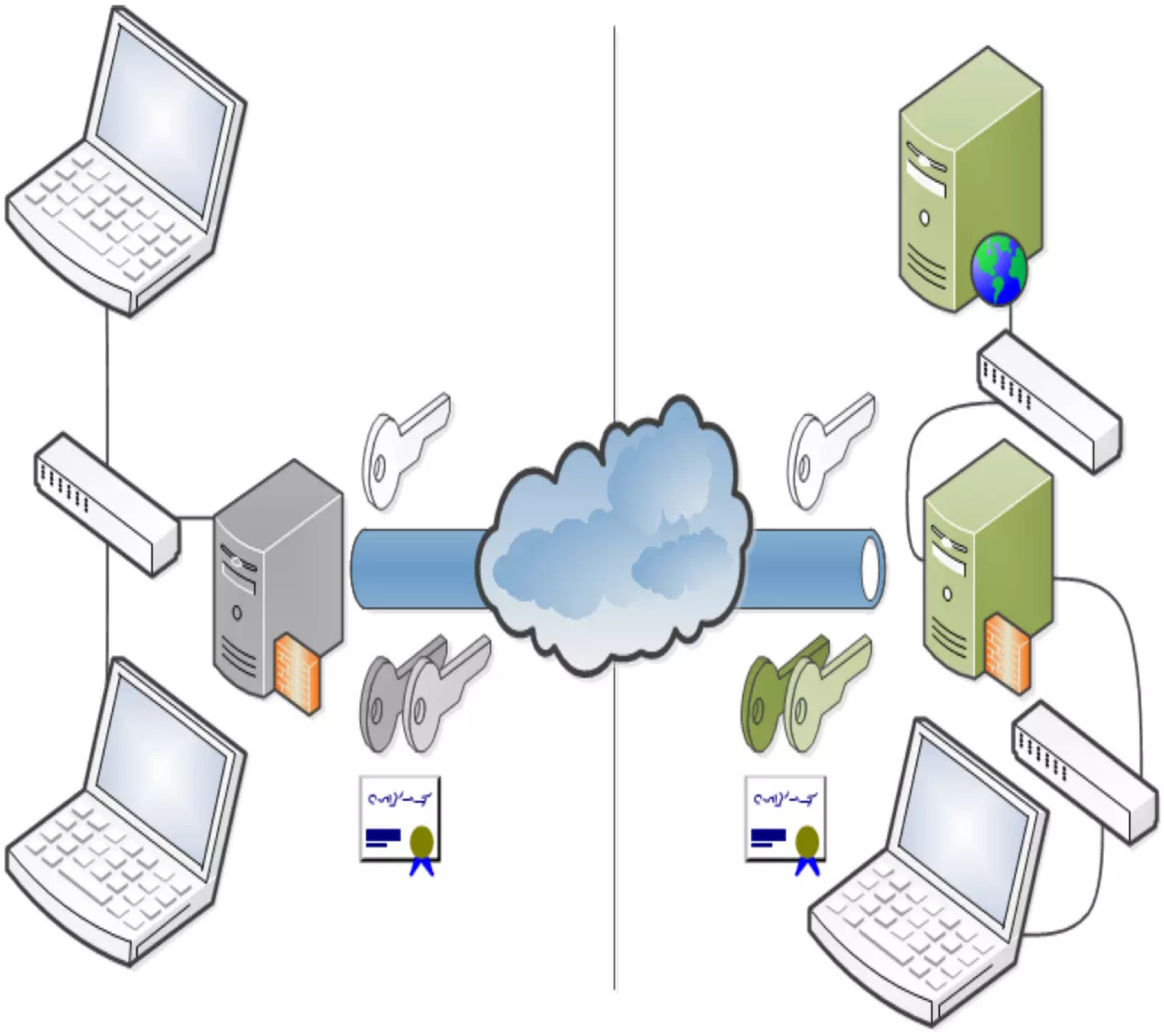
Authentication (Hash chain, S/KEY)



Authentication (Hash chain, S/KEY)

- Hashing
  1. Client tries to authenticate
  2. Server asks for 1000th hash of client's password
  3. The 1000th hash is generated and sent to the server. The server has only the 1000th hash of the user password, not the password. If they match, the user is authenticated
  4. Client attempts a new authentication
  5. This time the server asks for 999th hash of client's password
  6. The 999th hash is generated and sent to the server. The server has only the 1000th hash of the user password. The server must hash the 999th hash one more time to compute 1000th hash. If they match, the user is authenticated.
  7. The server then stores the 999th hash in its database. Next time the server will ask for the 998th hash. Security comes from the fact that the server is always asking for the previous hash, that only who knows the password can generate.

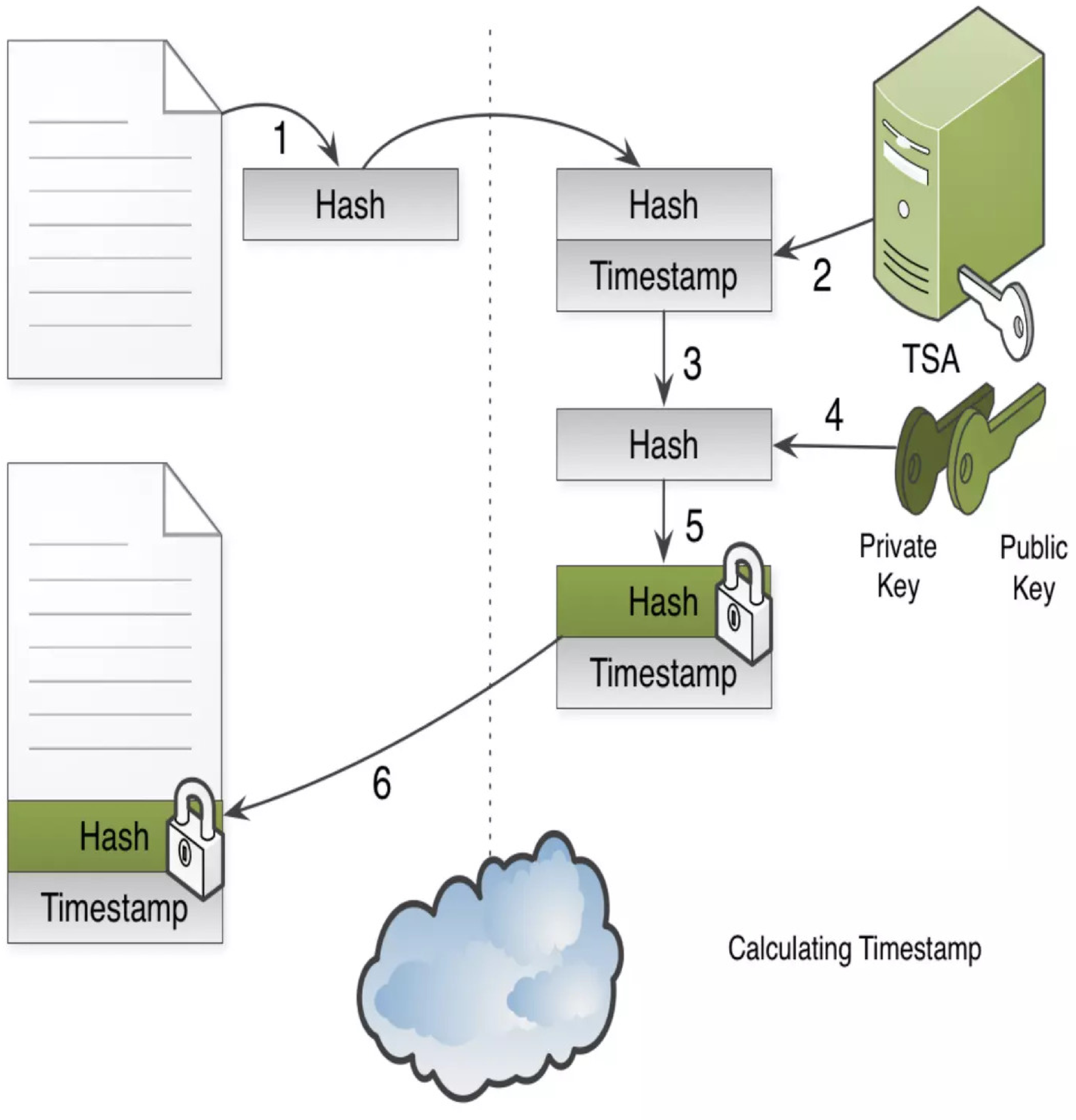
VPN



Authentication
(WPA / WPA2: *Personal* or *Enterprise*)



Trusted Timestamps

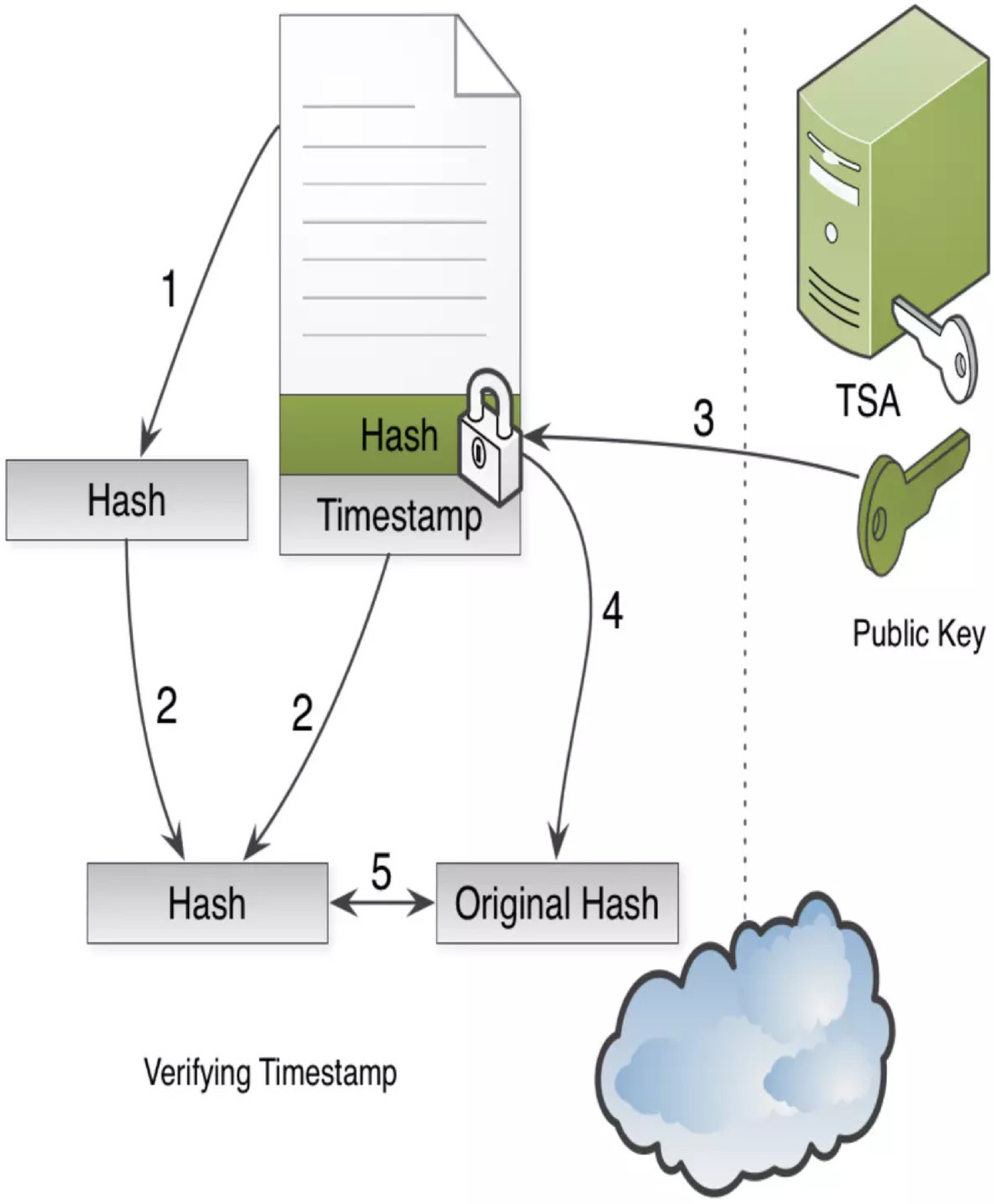


Simplified version

Trusted Timestamps

- Timestamp calculation
  1. A hash is generated from the message
  2. TSA server provides accurate time
  3. Another hash is generated from the hash and the timestamp
  4. The hash and the timestamp are encrypted using TSA's private key
  5. The timestamp is added to the encrypted hash
  6. The hash + timestamp is sent to the client and added to the message

Trusted Timestamps



Simplified version

Trusted Timestamps

- Timestamp verification
  1. A hash is generated from the message
  2. The timestamp added to the message is recovered and another hash is generated from the first hash and the timestamp
  3. Public key from TSA is obtained
  4. The original hash generated by the TSA is obtained
  5. Comparison between both hashes

Asymmetric Cryptography

- External storage for the private key





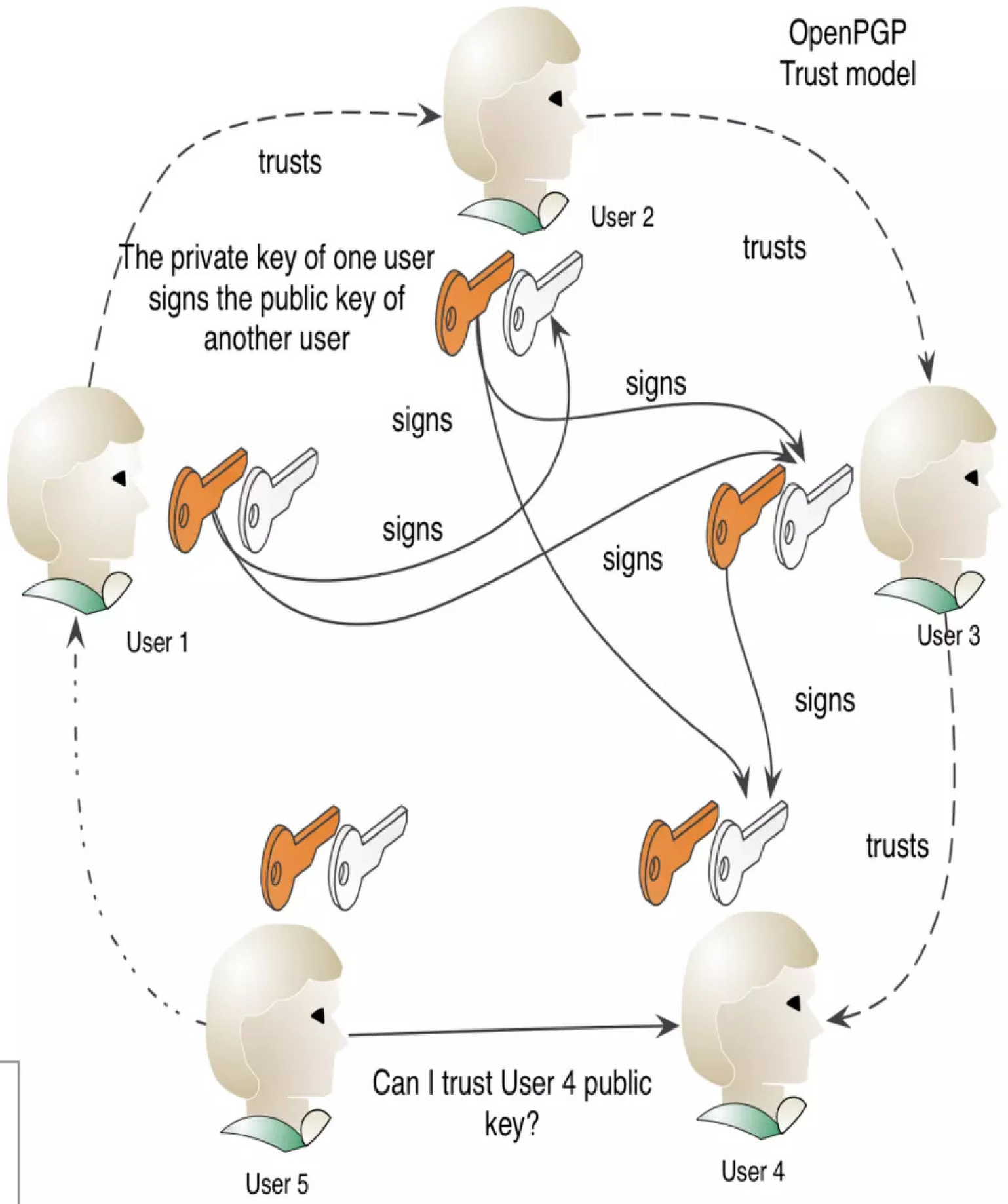
Asymmetric Cryptography

- Asymmetric key security depends on
  - Security of the private key
  - Strength of the password of the private key (just like any password)
  - Security of the cryptographic method (various types of attacks)

Asymmetric Cryptography

- Problems
  - How do you grant everyone access to all public keys? Will everyone send their keys to everyone else? And in case of revocation? How to replace the key pair?
  - How do you know if a key pair is trustworthy, or belongs to the person you think it does?

Web of Trust



Simplified explanation

Asymmetric Cryptography

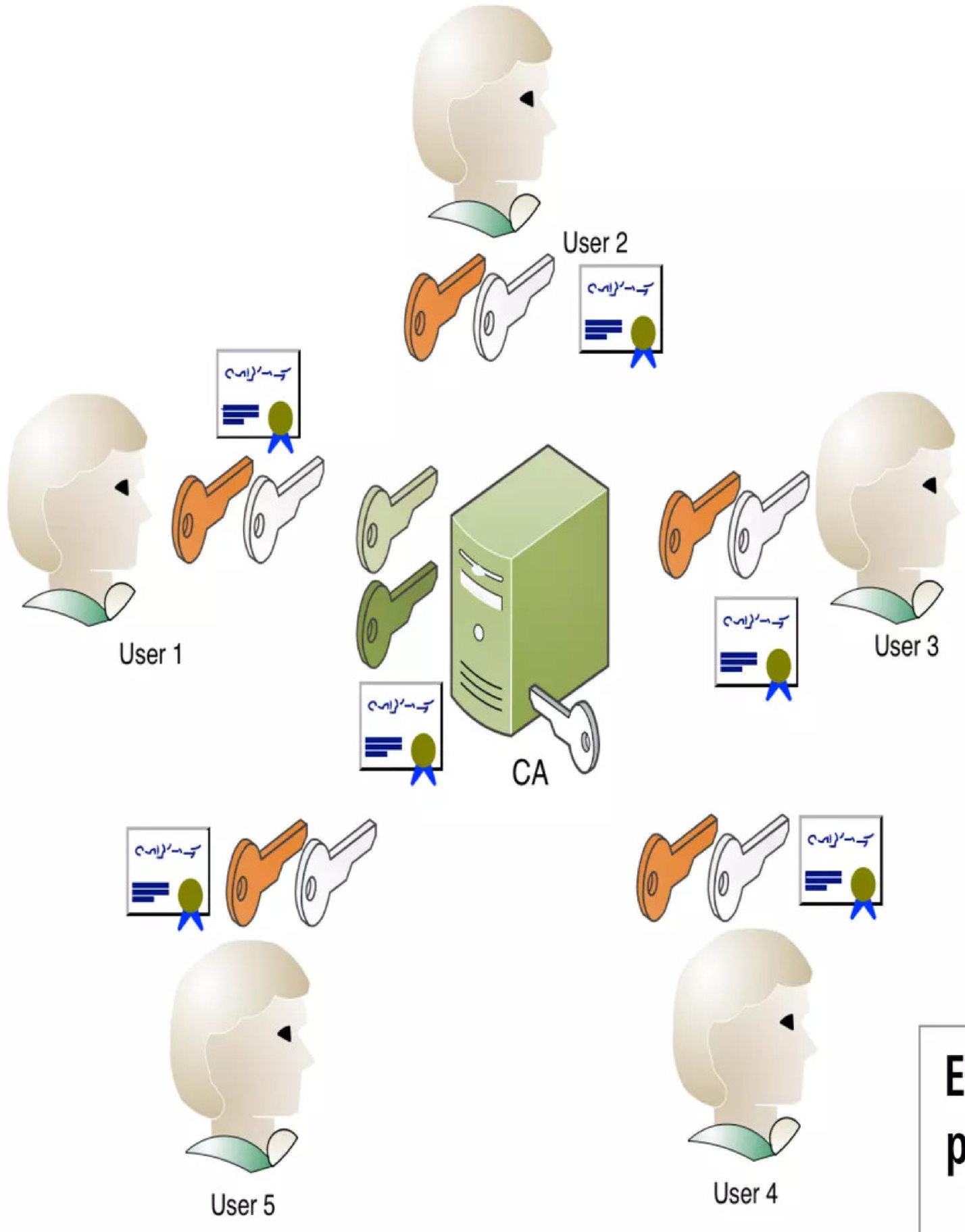
- Problems
  - How do you grant everyone access to all public keys? Will everyone send their keys to everyone else? And in case of revocation? How to replace the key pair?
    - Web of Trust allows a designated revocator
  - How do you know if a key pair is trustworthy, or belongs to the person you think it does?
    - Web of Trust depends on one person (or various) to endorse that key and on a “trust path” between two people

But there are other problems

Agenda

- What is cryptography?
- Hash
- Symmetric cryptography
- Asymmetric cryptography
- Public Key Infrastructure (PKI)
- Attacks
- Steganography
- Best practices: domestic use
- References

Public Key Infrastructure (PKI)



Every user has a pair of keys and a certificate

Public Key Infrastructure (PKI)

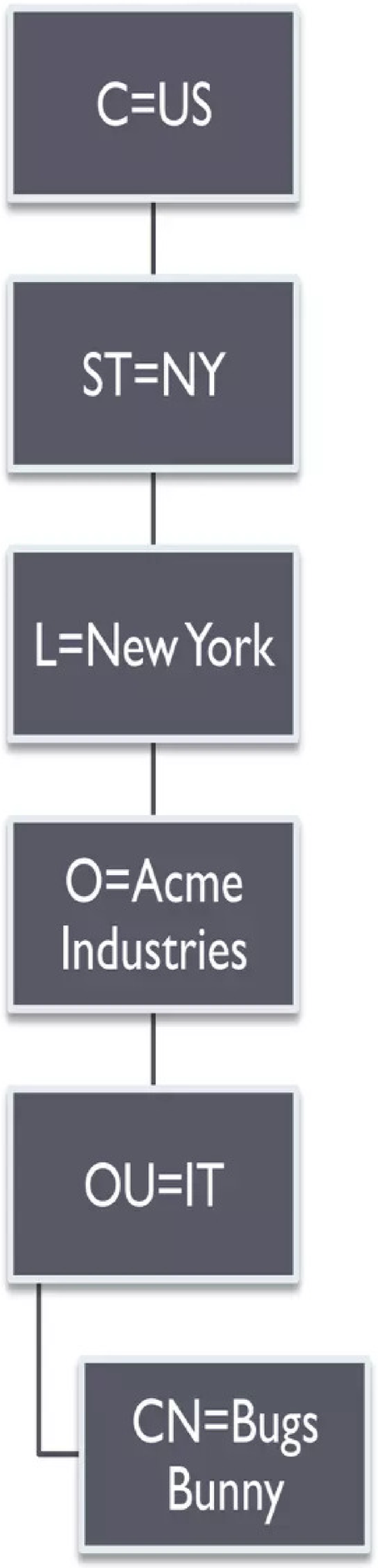
- X.509 v3 digital certificate
- Trust relationships
- Certificate import and export
- Certificate issuing, verification and revocation

Digital Certificate

- X.509 self-signed digital certificate (example)

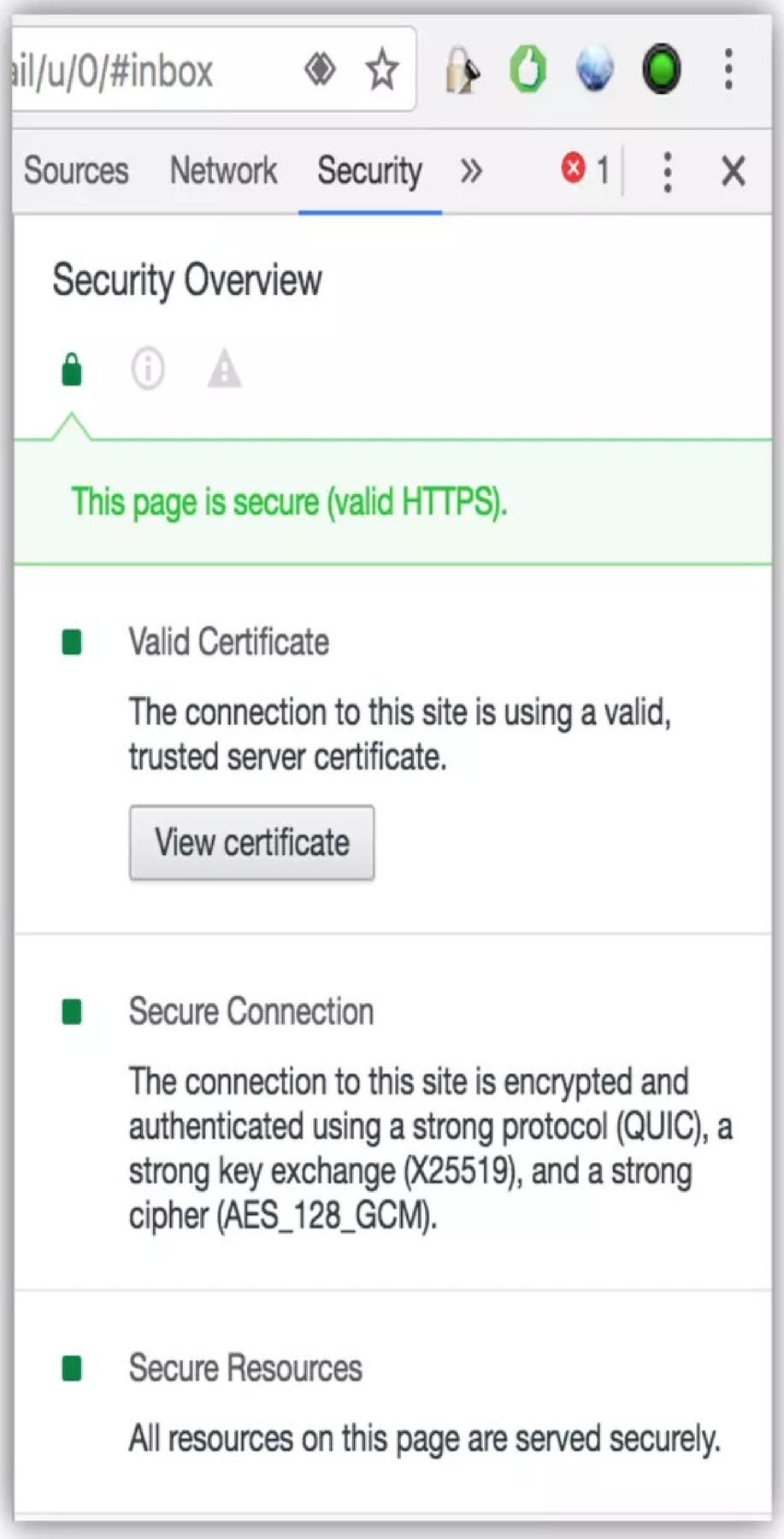
```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
            OU=Certification Services Division,
            CN=Thawte Server CA/emailAddress=server-certs@thawte.com
    Validity
        Not Before: Aug  1 00:00:00 1996 GMT
        Not After : Dec 31 23:59:59 2020 GMT
    Subject: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
            OU=Certification Services Division,
            CN=Thawte Server CA/emailAddress=server-certs@thawte.com
    Subject Public Key Info:
```

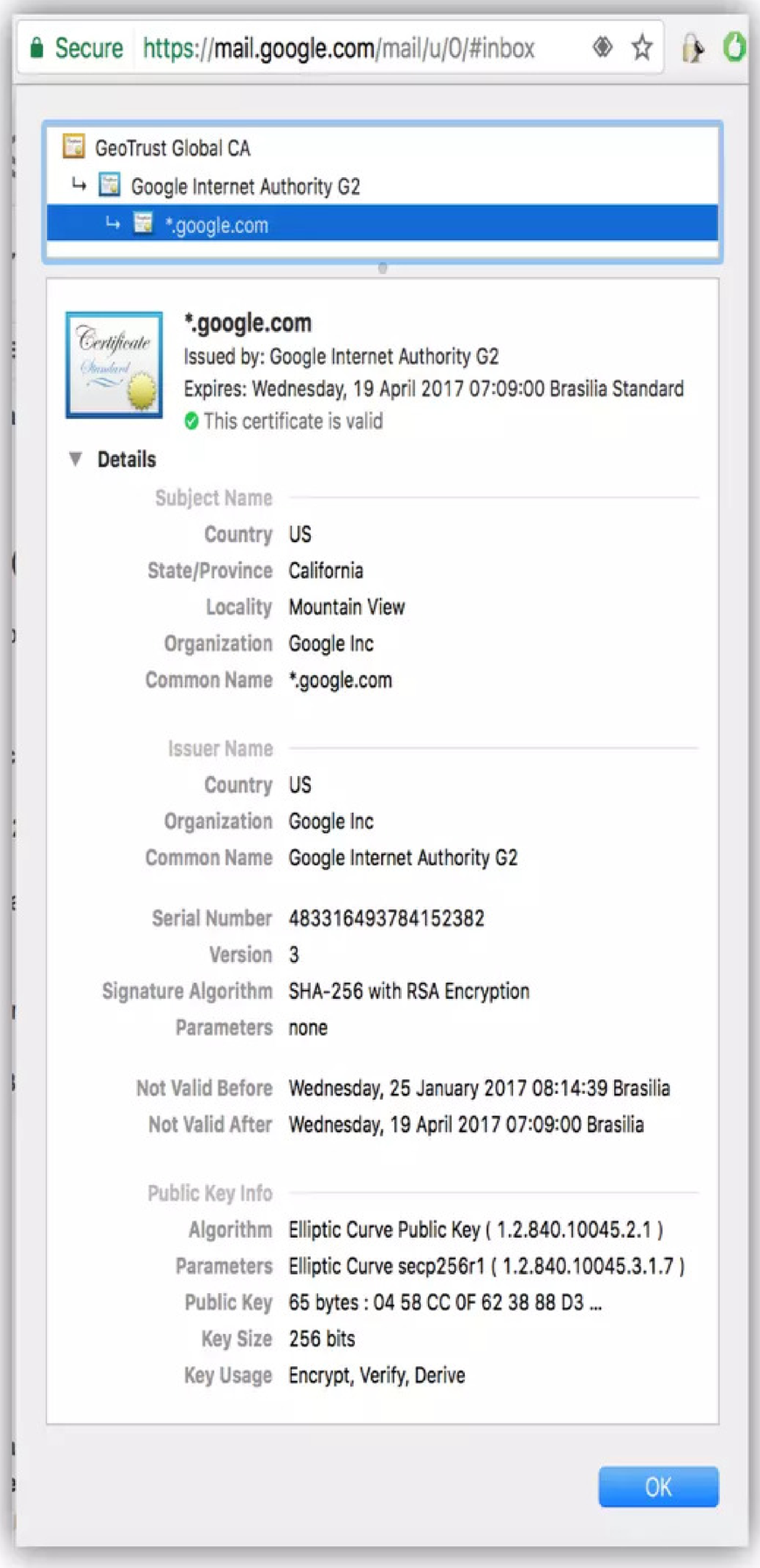
Digital Certificate



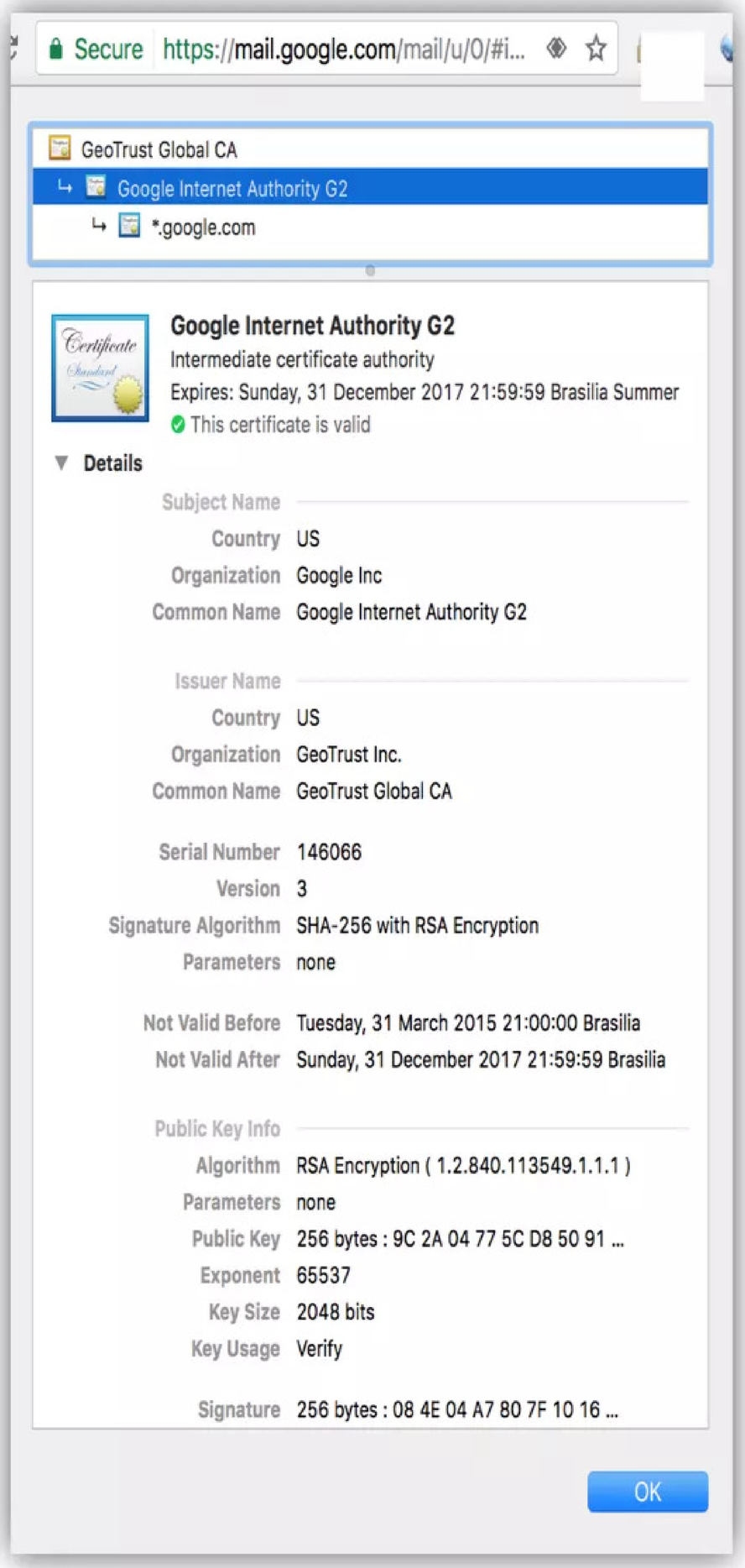
Directory structure

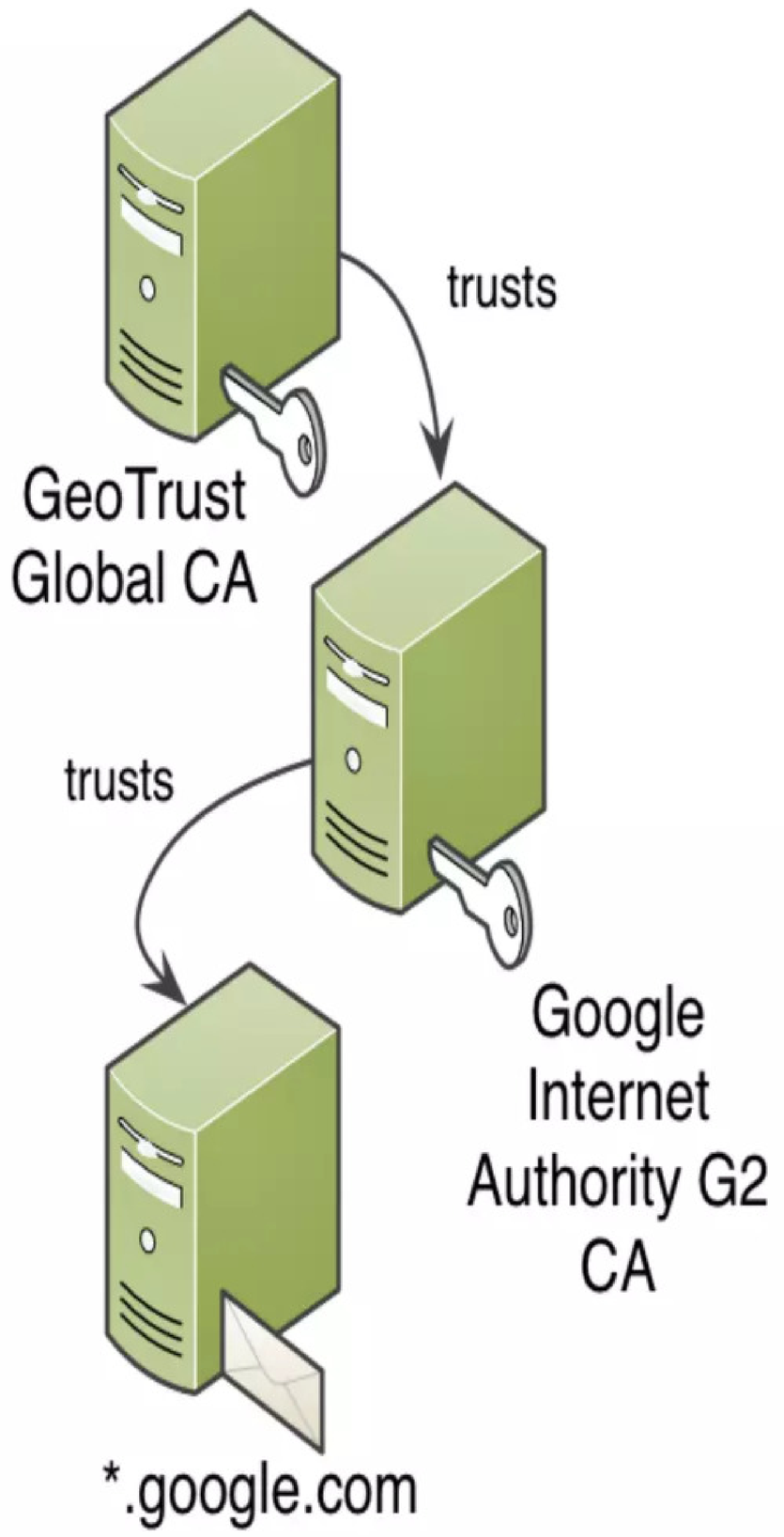
Digital Certificate





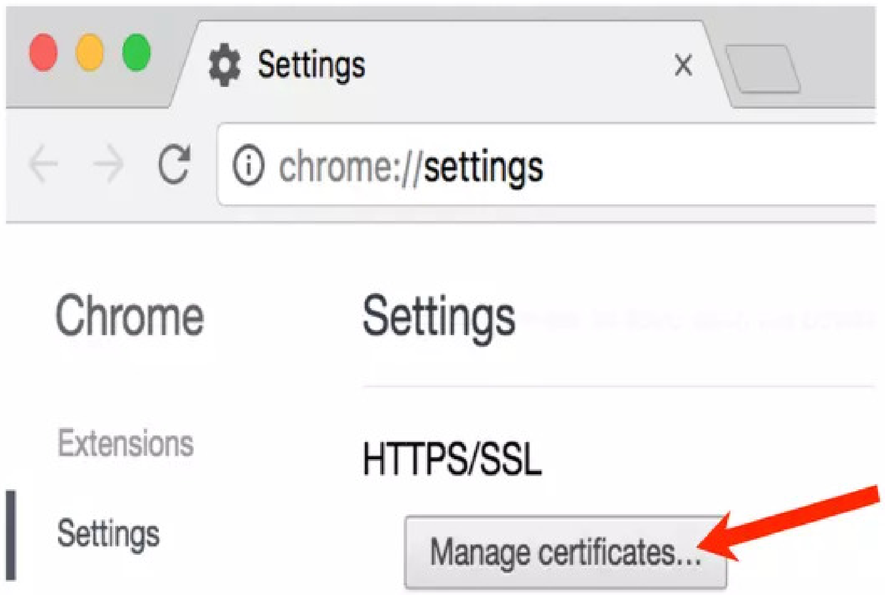
Digital Certificate



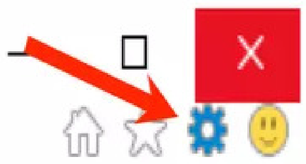


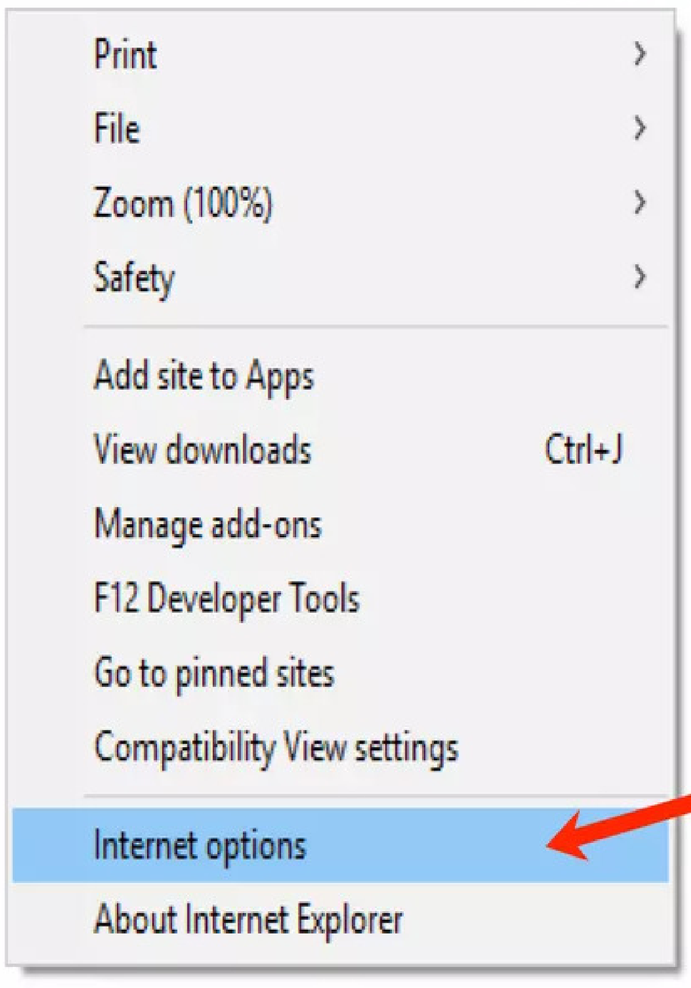
Certificate Manager



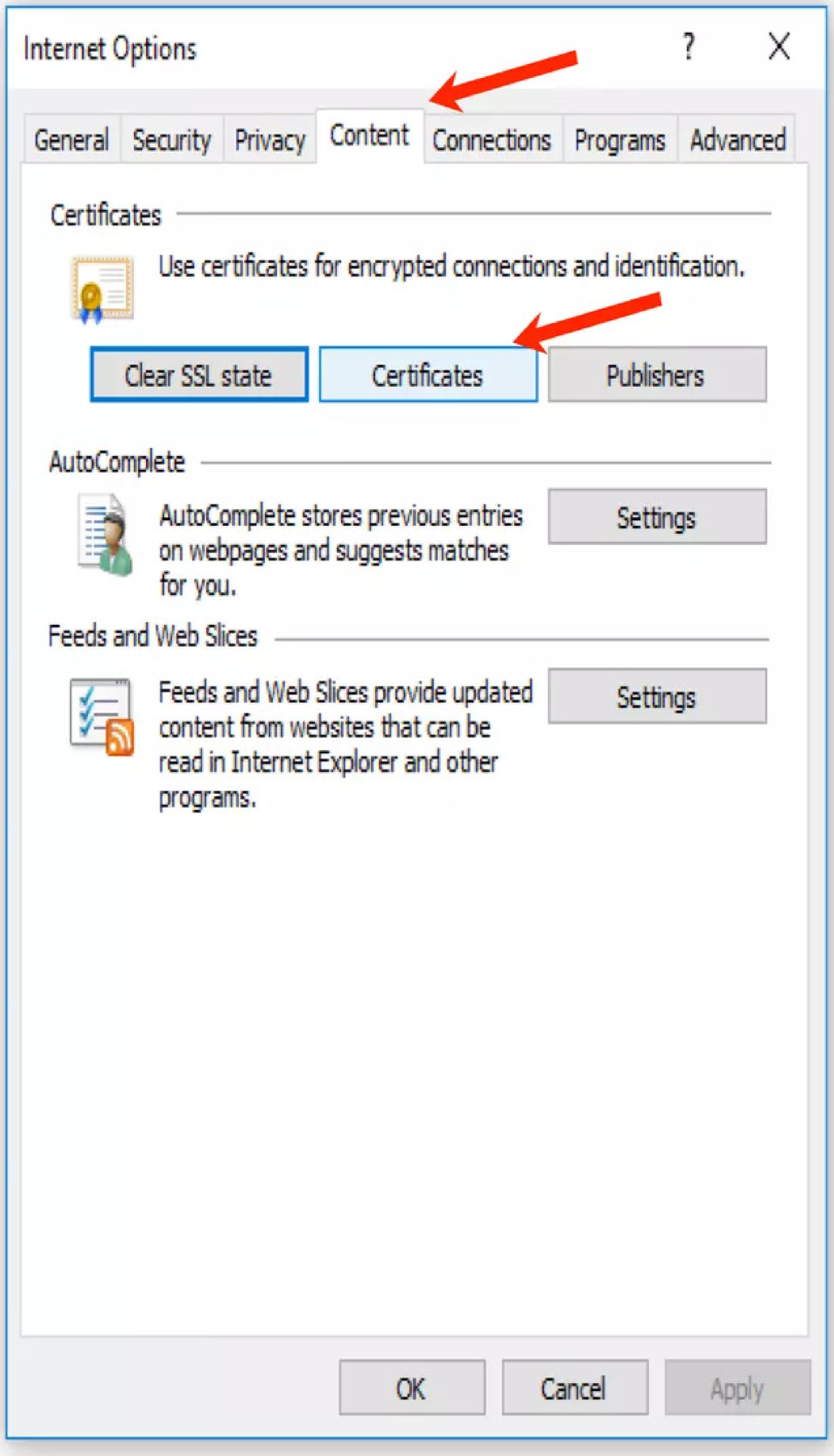


or

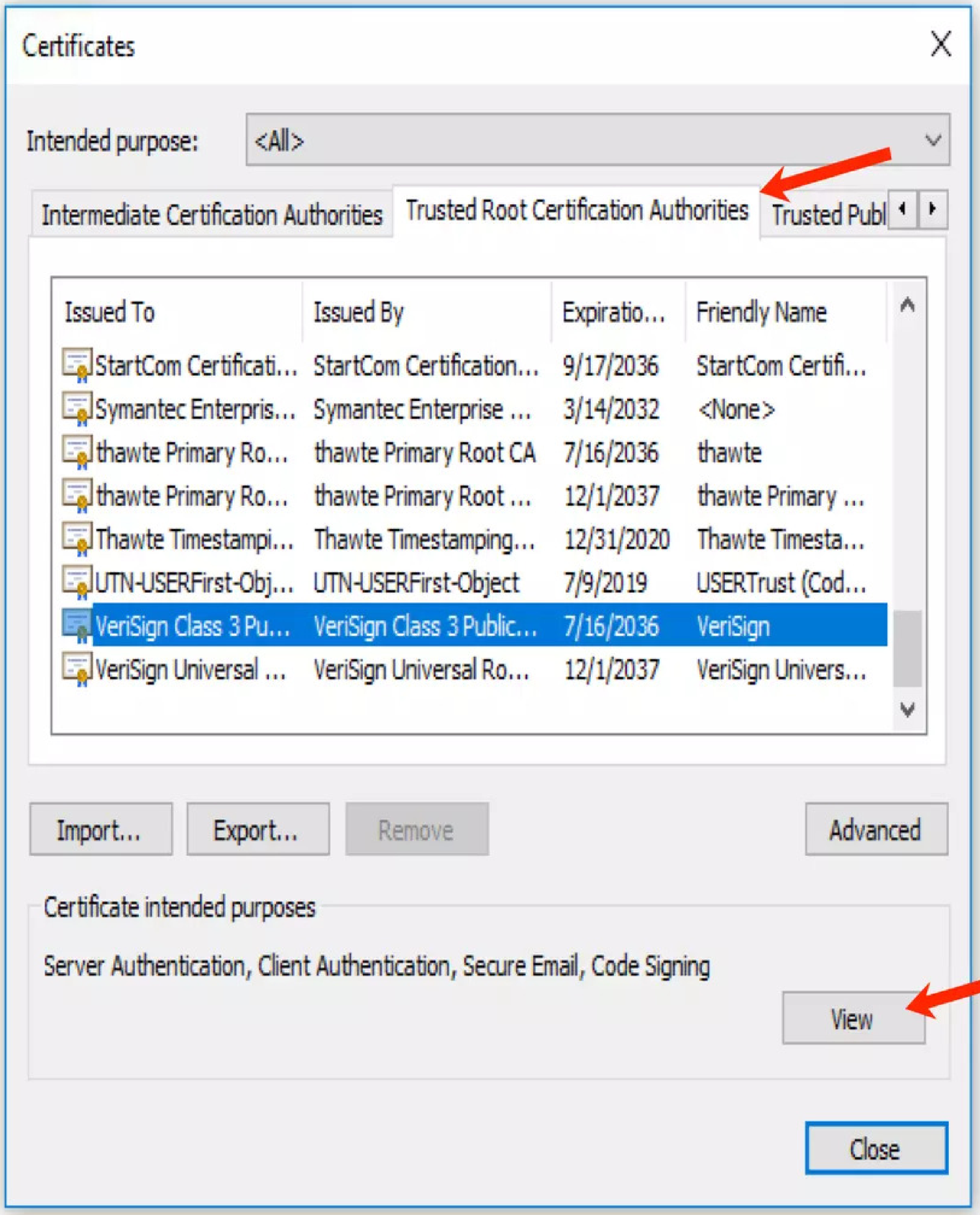




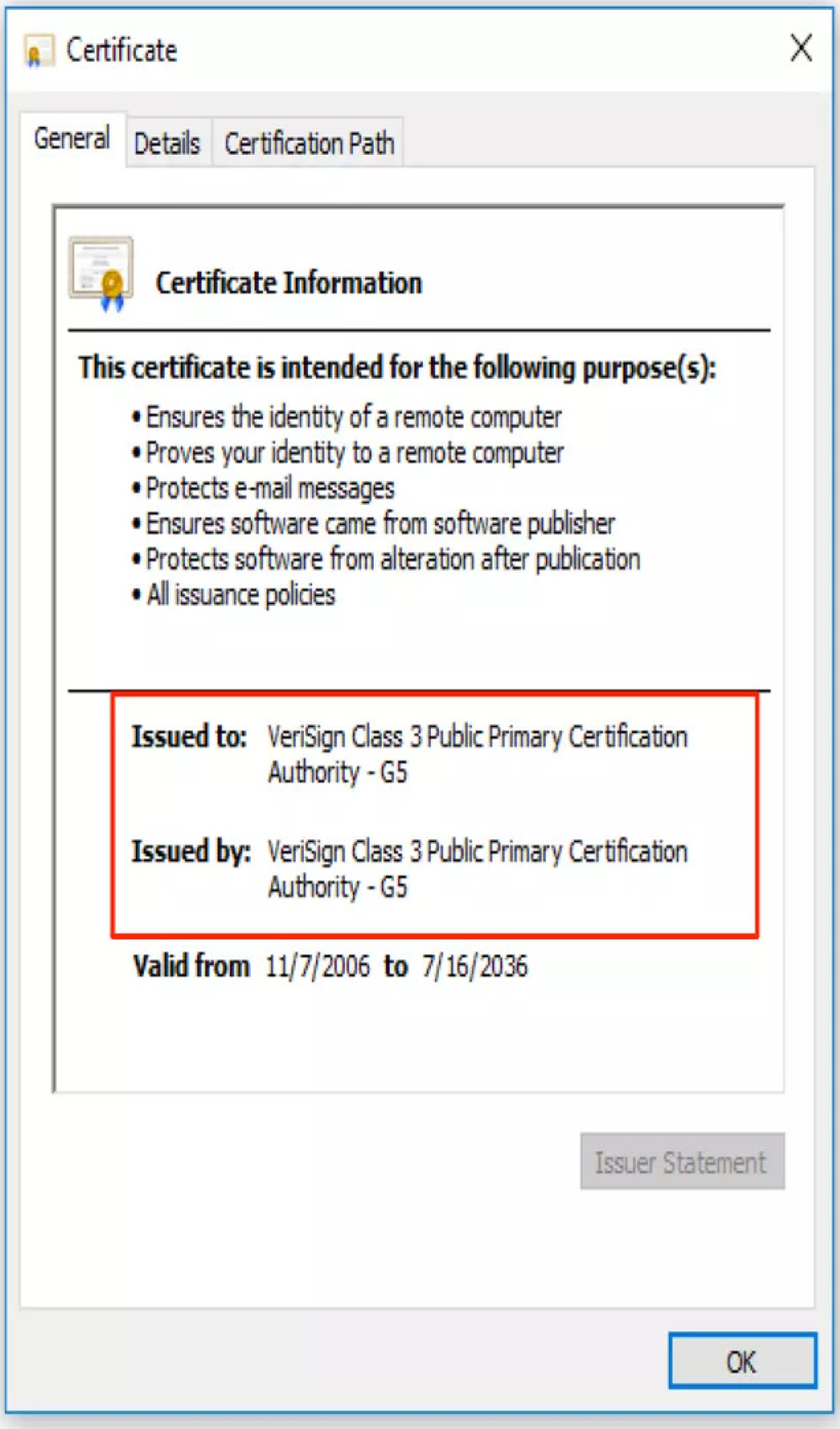




Root Certificates

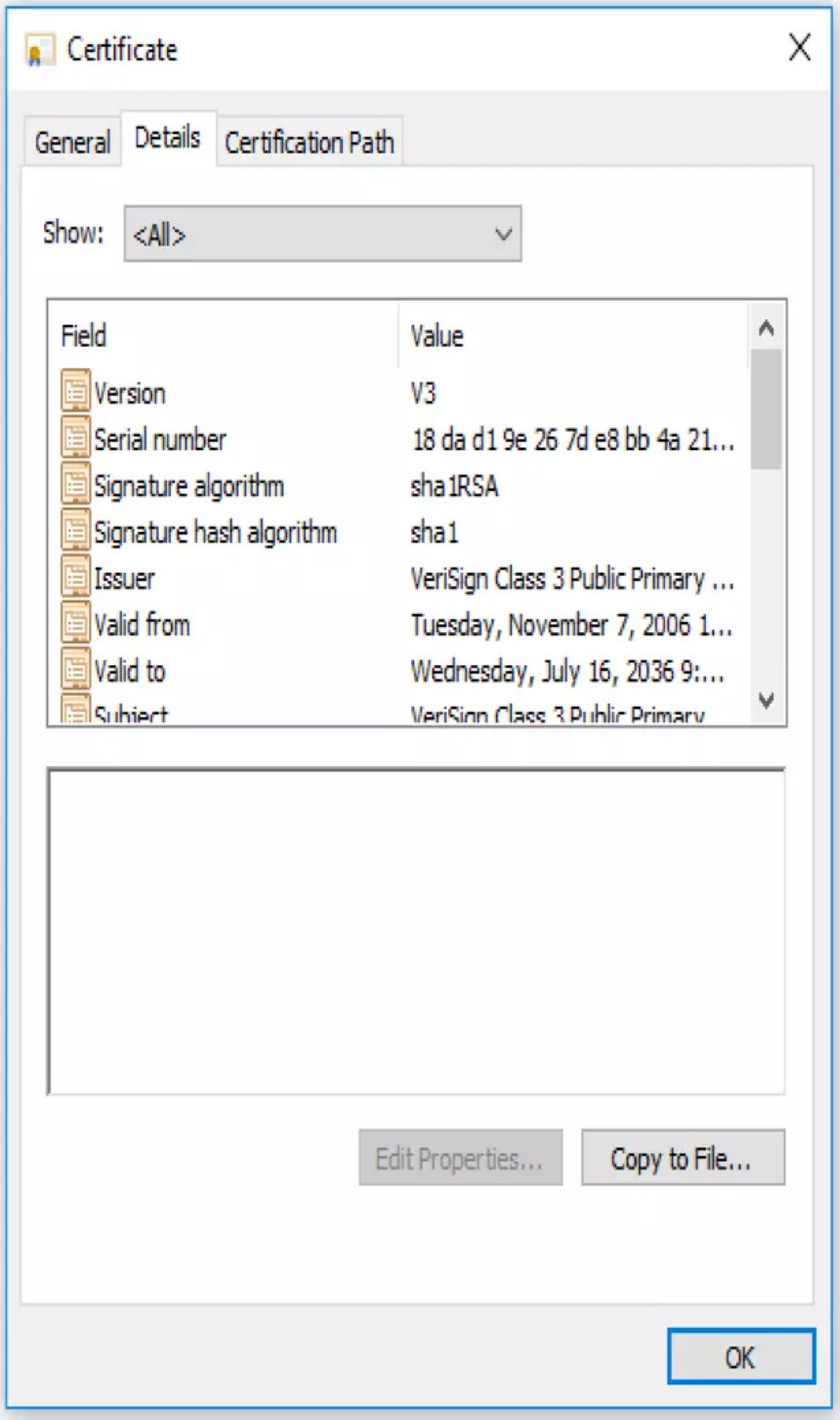


Root Certificates



Root CA self-signed certificate

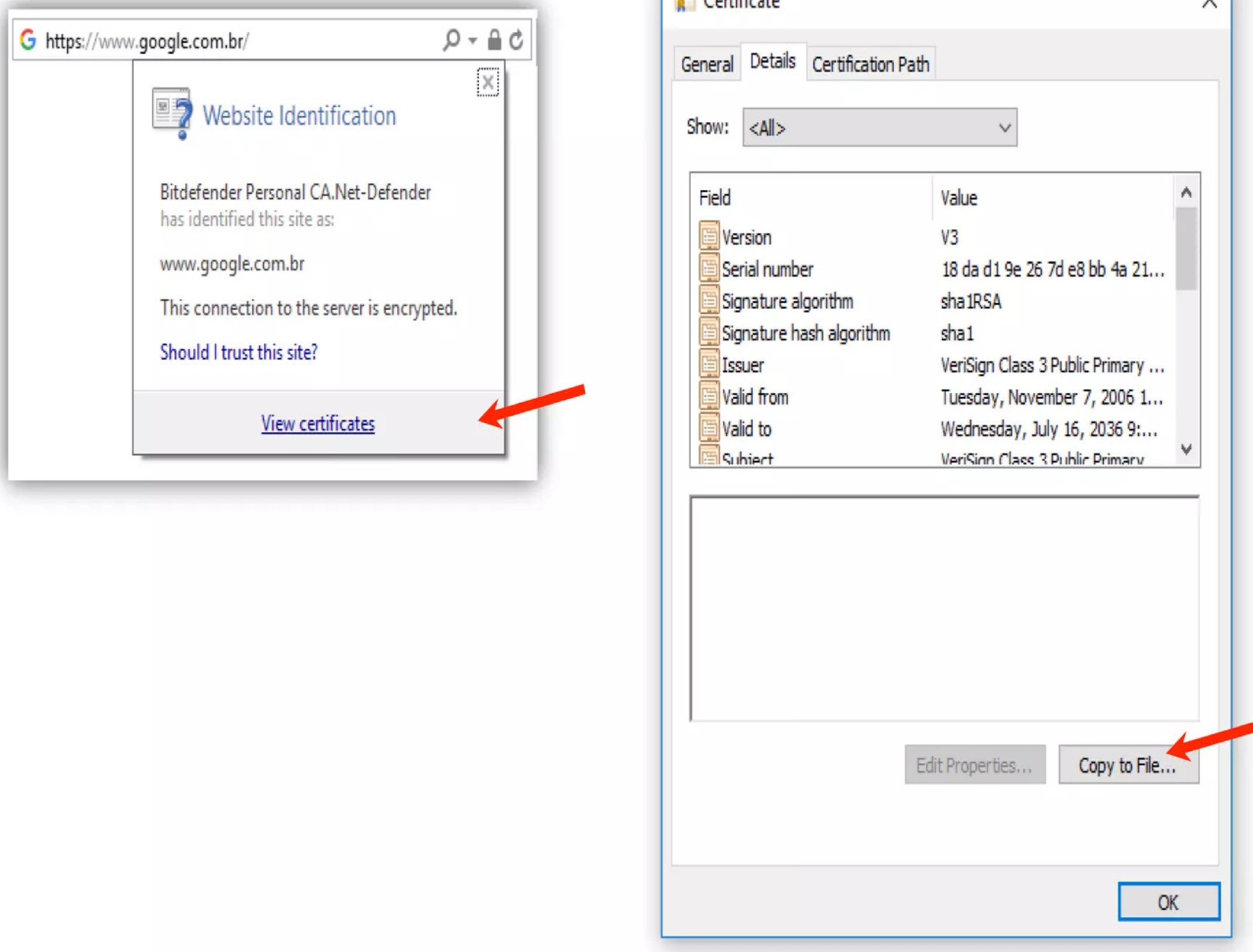
Root Certificates



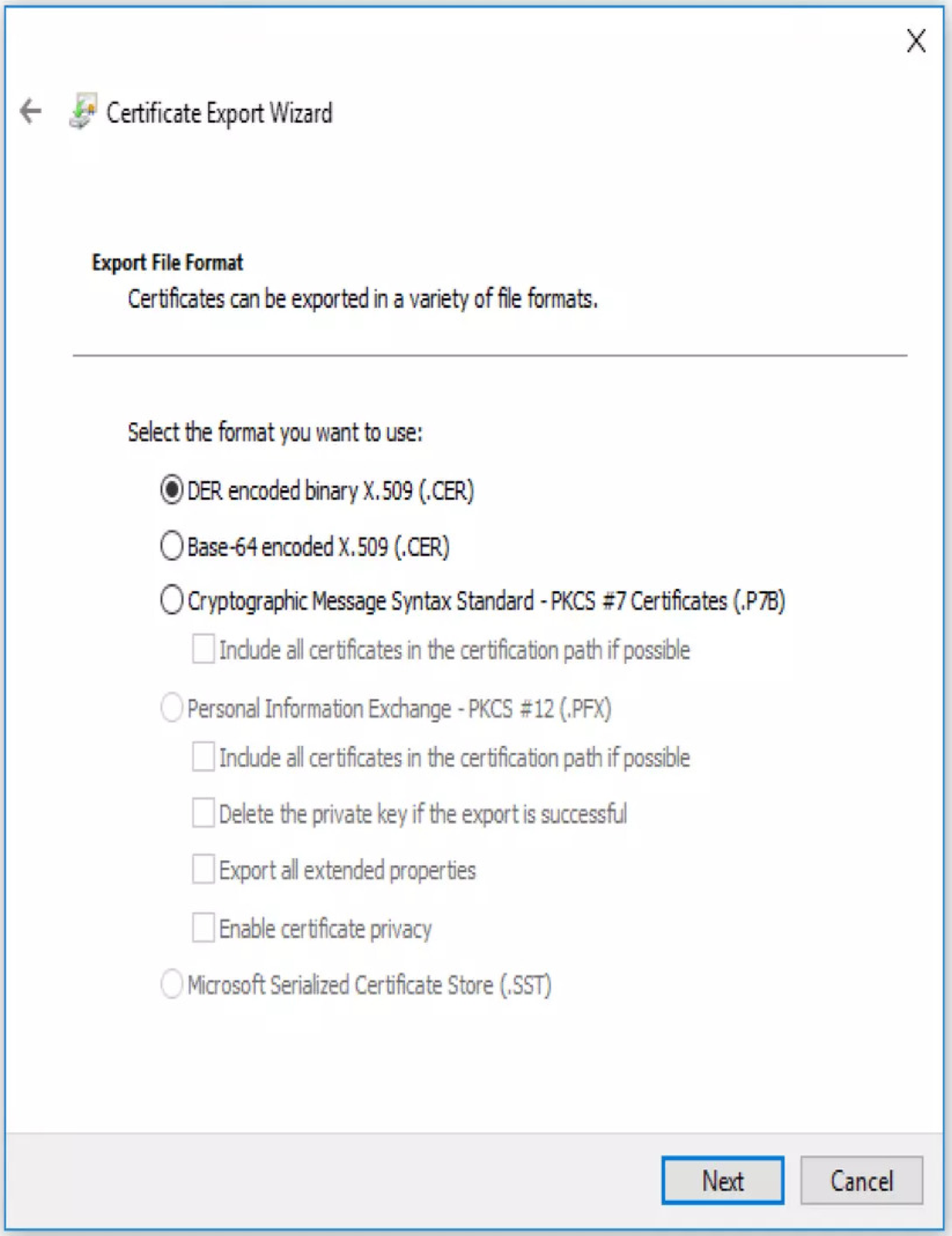
Viewing the details of the Root CA certificate

Digital Certificate

- Digital certificate export process



Digital Certificate

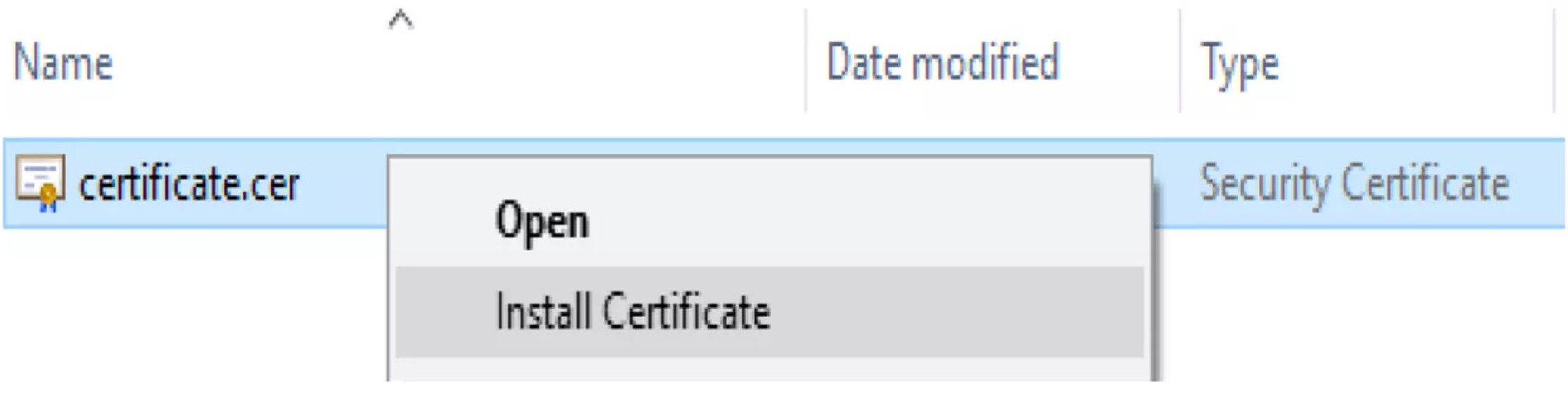


Select the format and click Next

Digital Certificate

- Digital certificate import process

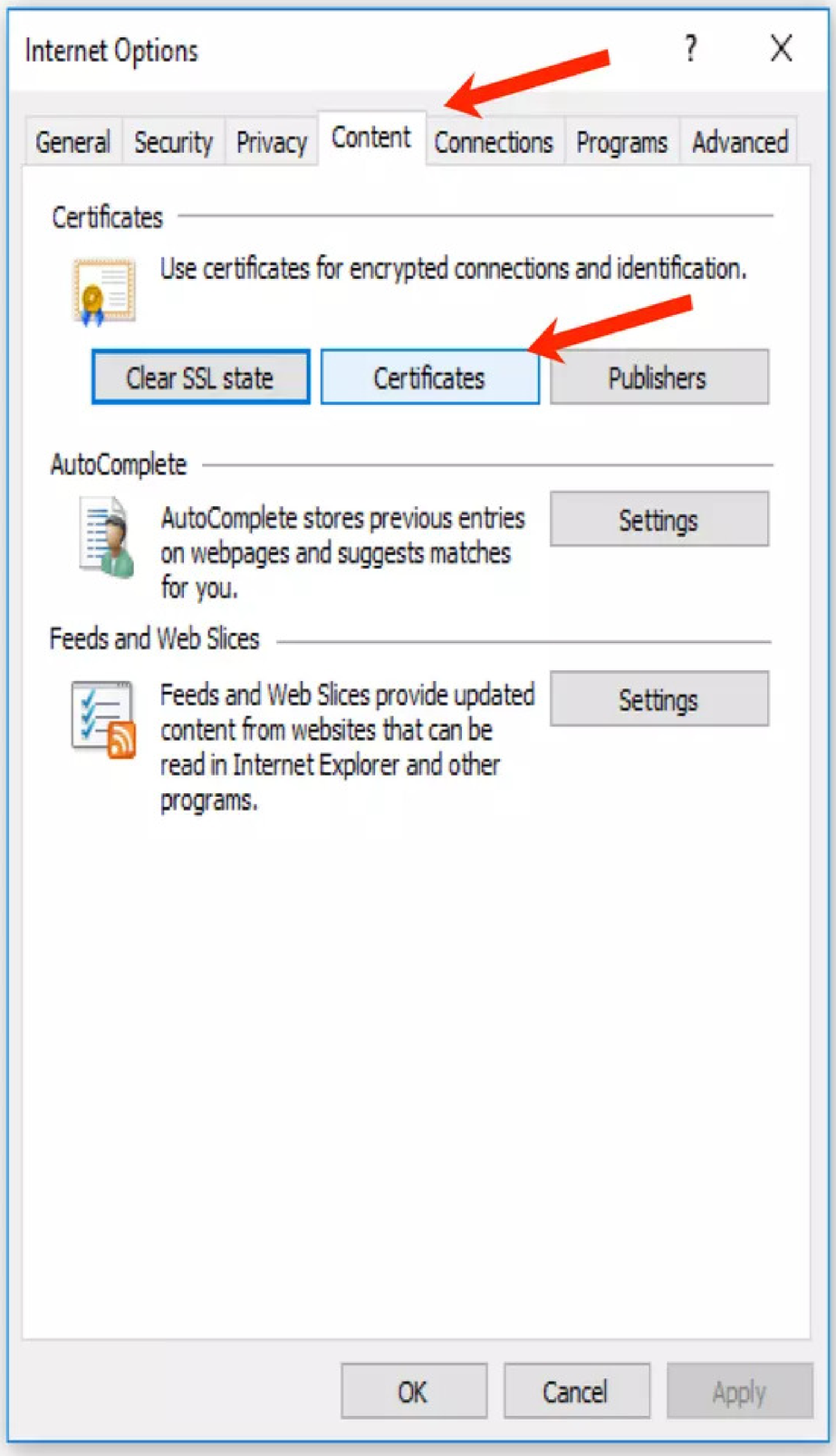


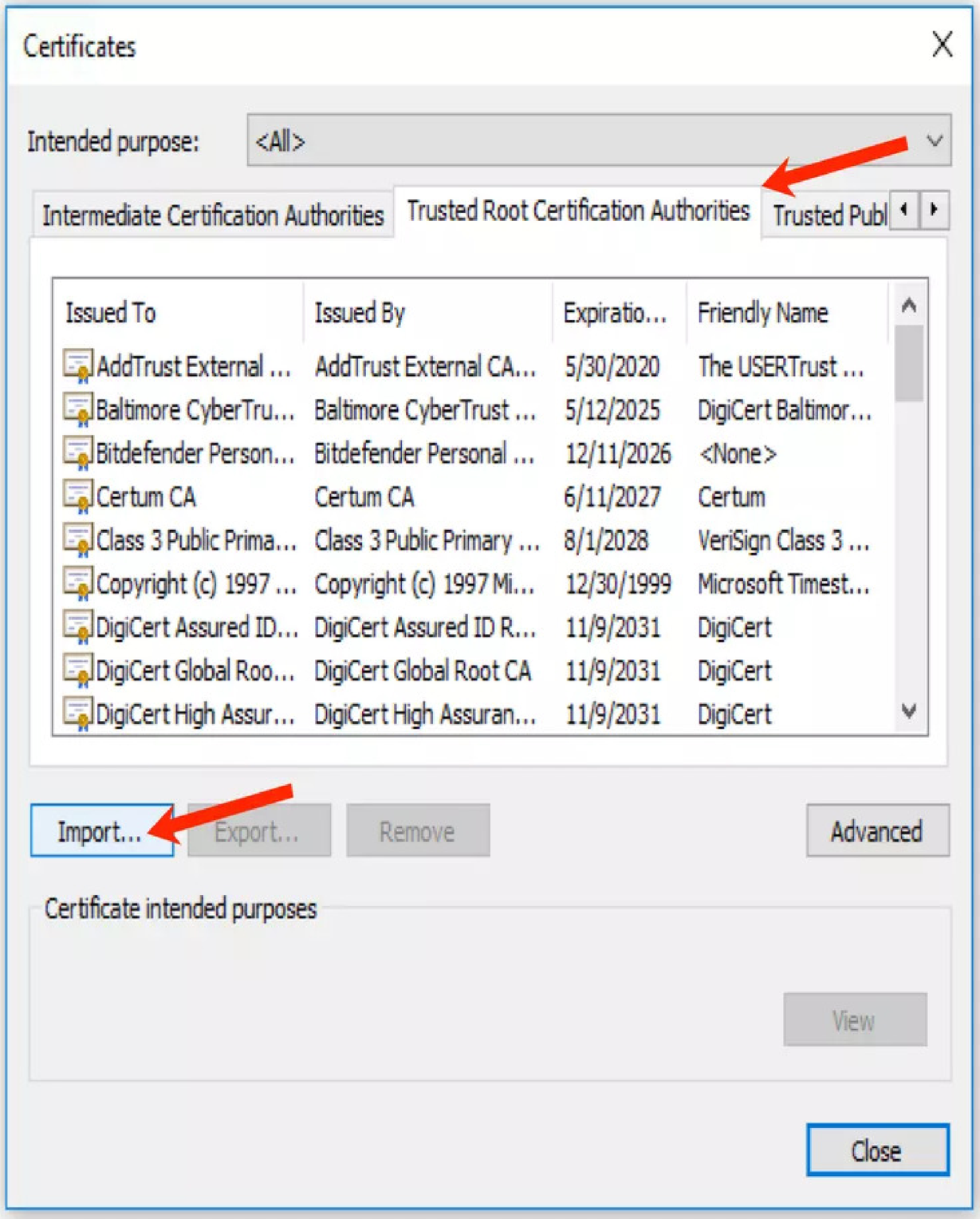


Be careful: When a certificate is installed (becomes trustworthy) the applications signed with that certificate also become trustworthy

Method #1

Digital Certificate

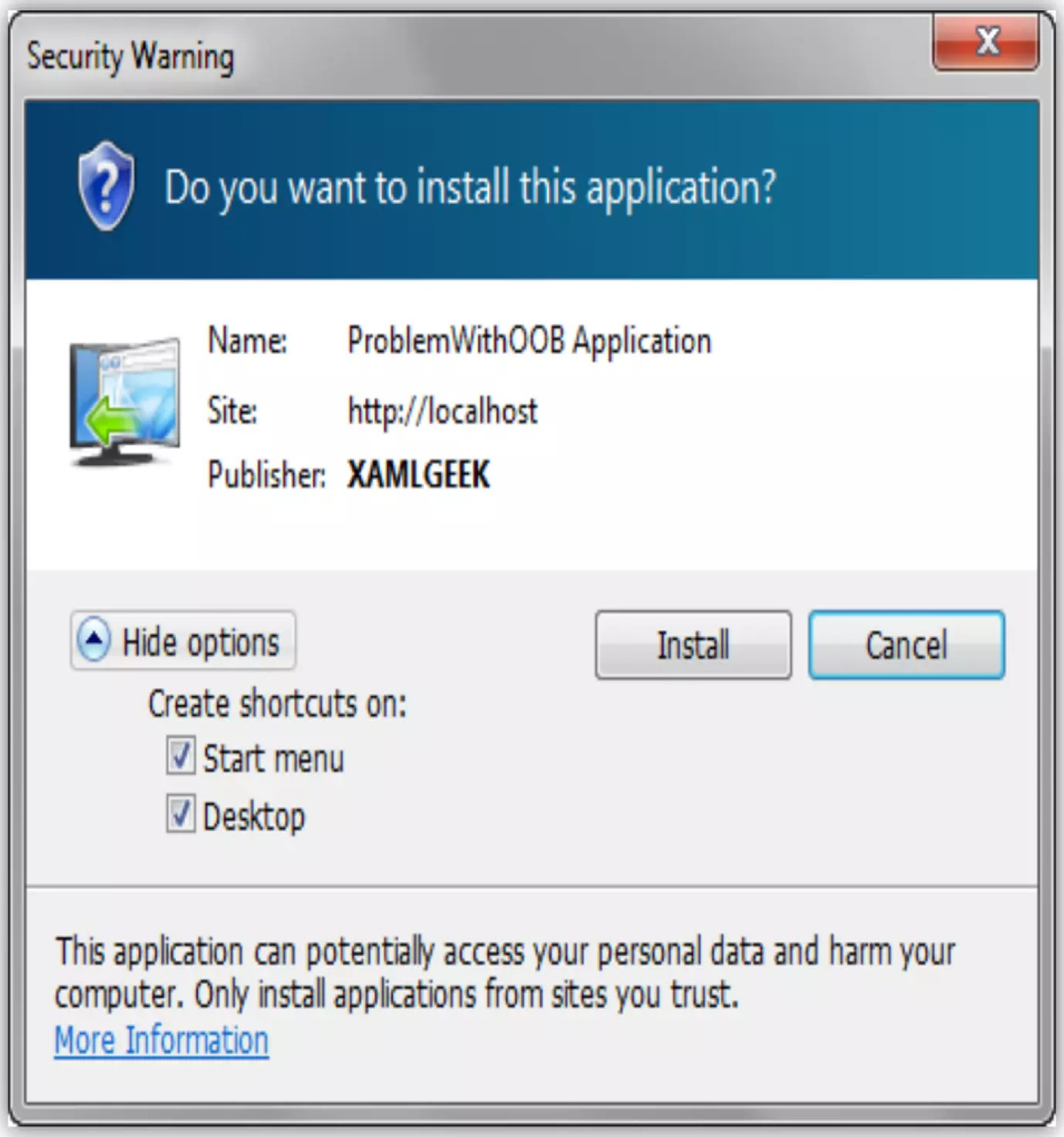




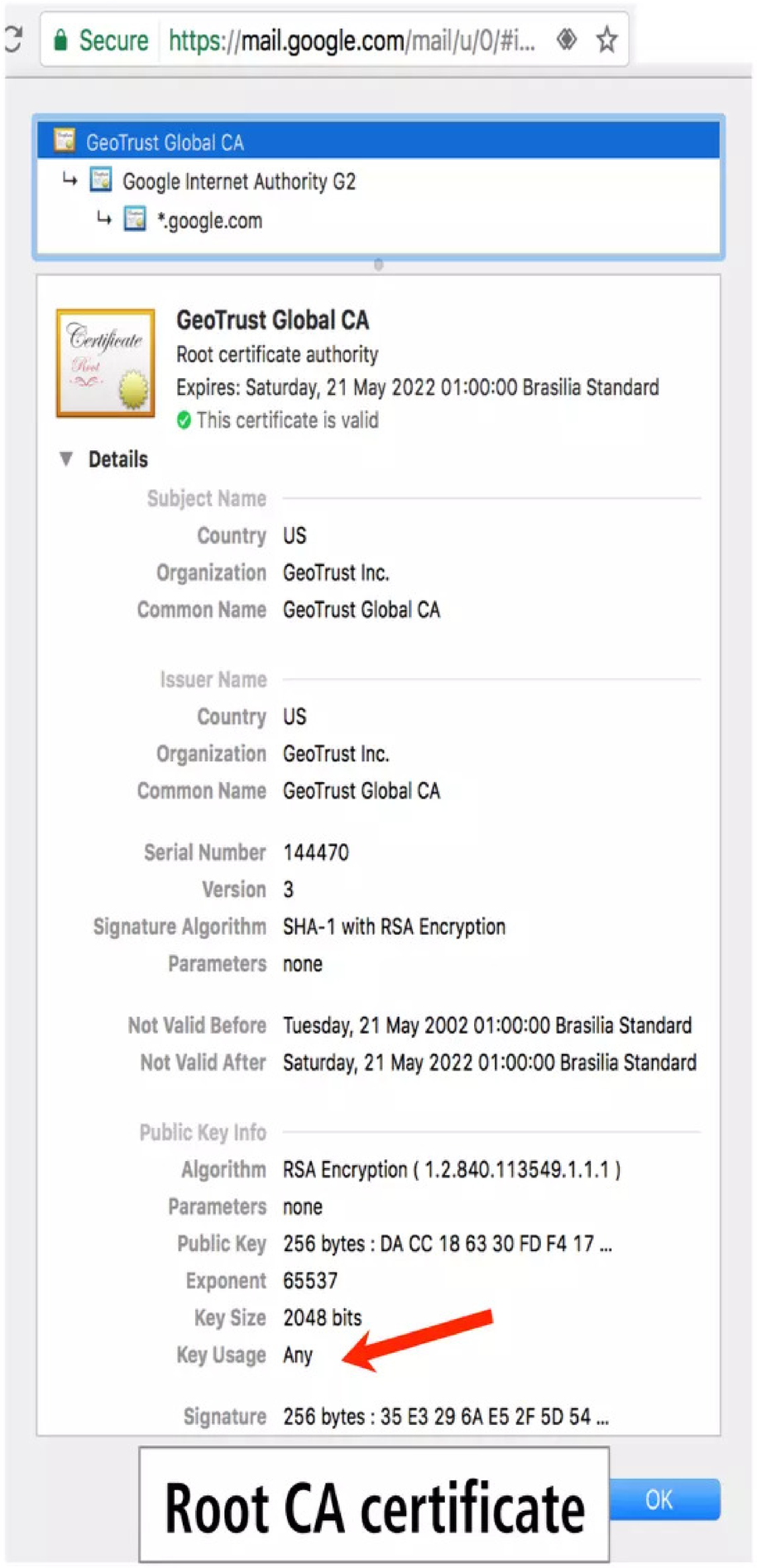


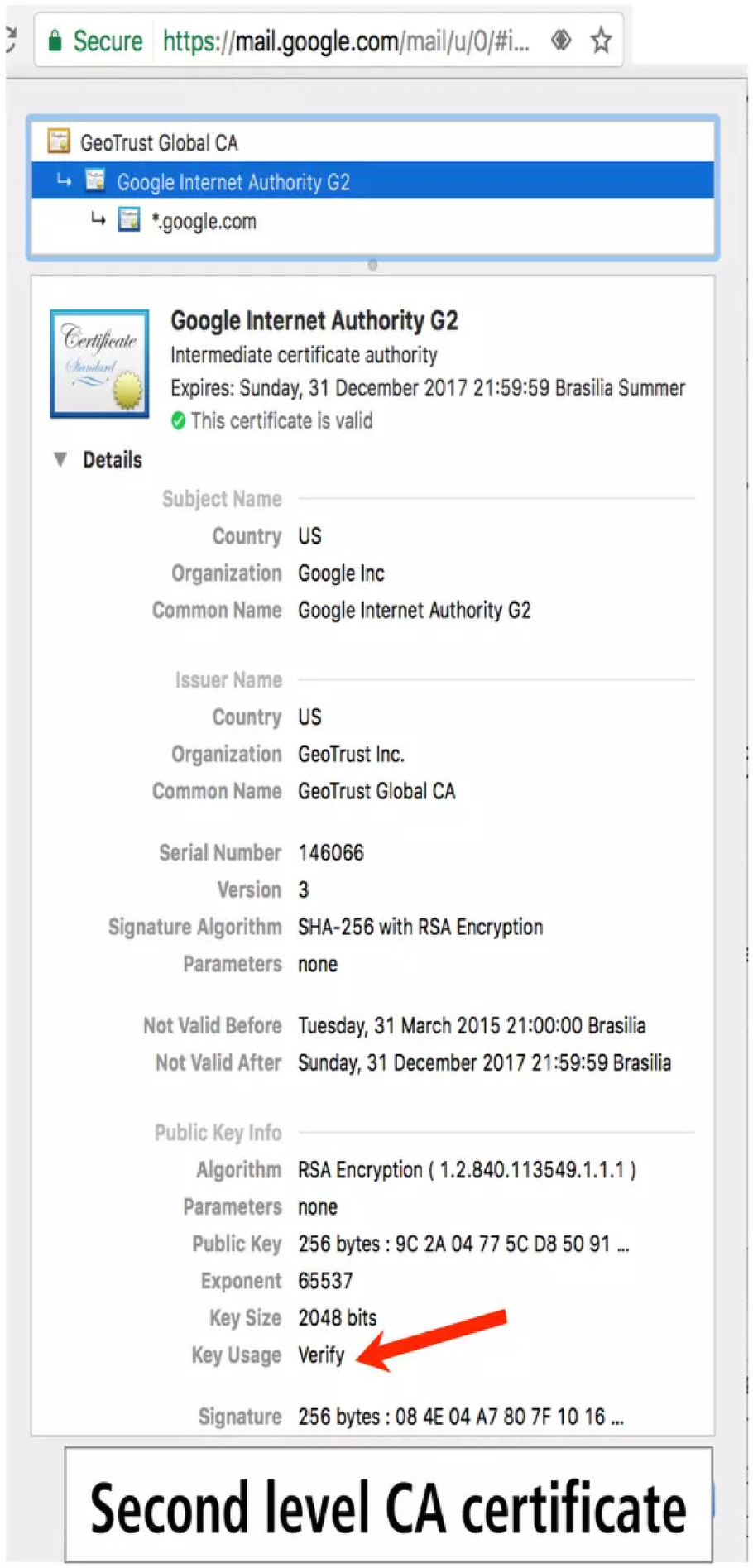
Method #2

Digital Certificate

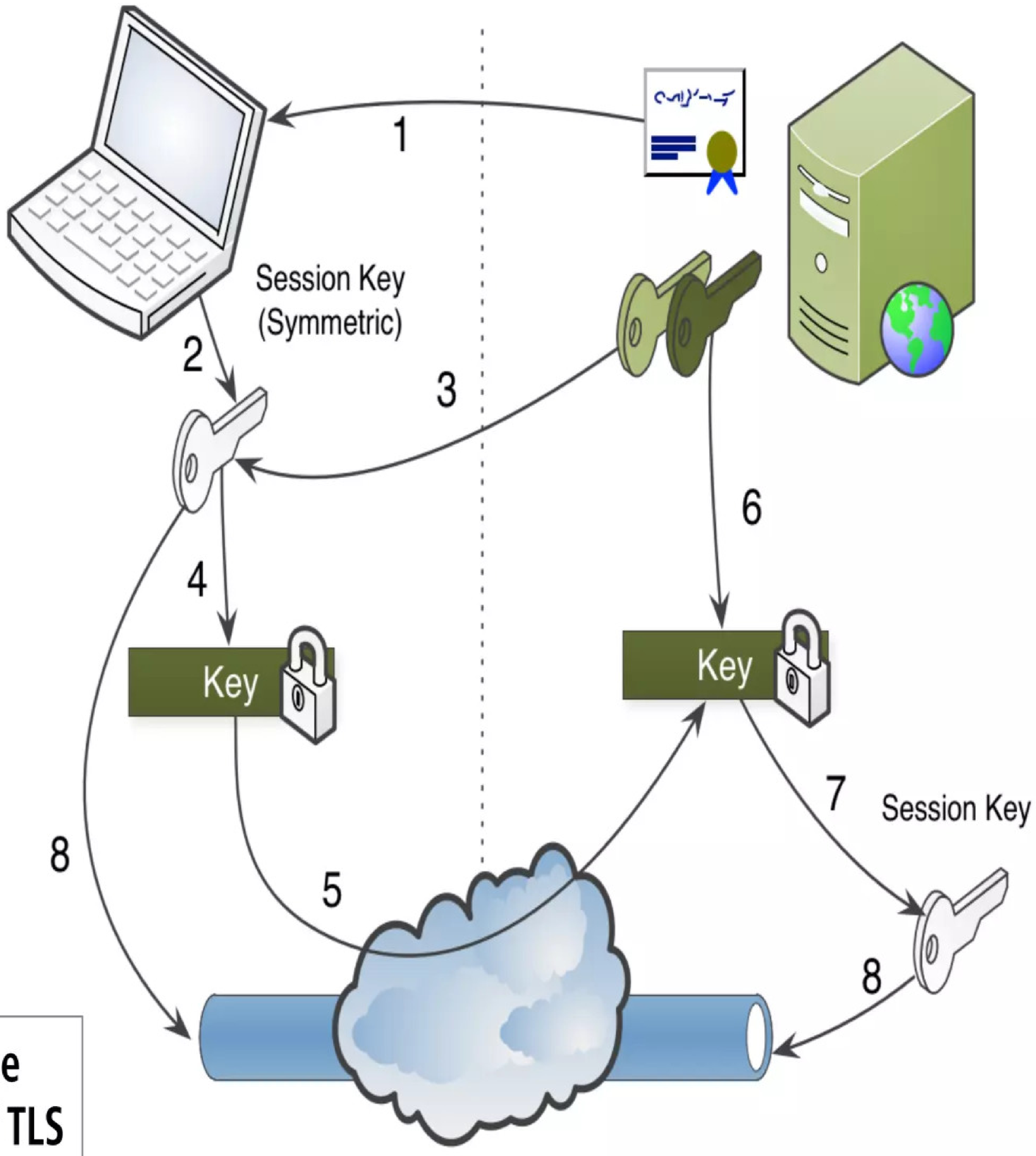


Digital Certificate





SSL/TLS



Always use the latest version of TLS

SSL/TLS

- Encryption and decryption
  1. Server's certificate is obtained
  2. Client generates a session key
  3. The session key is encrypted using server's public key (from the certificate)
  4. Encrypted session key is obtained
  5. Encrypted session key is sent over insecure media
  6. Server uses its private key to obtain the session key
  7. Session key is obtained
  8. Both parties use the session key to encrypt and decrypt the following messages

SSL/TLS

- Wireshark: www.[redacted]

No.	Time	Source	Destination	Protocol	Length	Info
17	1.415863	80.[redacted]	192.168.1.5	TCP	60	http > 7614 [FIN, ACK] Seq=1 Ack=1 Win=108 Len=0
18	1.415968	192.168.1.5	80.[redacted]	TCP	54	7614 > http [ACK] Seq=1 Ack=2 Win=4124 Len=0
51	4.382509	192.168.1.5	80.[redacted]	HTTP	767	GET /index.php HTTP/1.1
95	4.738884	80.[redacted]	192.168.1.5	TCP	60	http > 7616 [ACK] Seq=1 Ack=714 Win=453 Len=0
96	4.765802	80.[redacted]	192.168.1.5	TCP	1466	[TCP segment of a reassembled PDU]
97	4.768312	80.[redacted]	192.168.1.5	TCP	1466	[TCP segment of a reassembled PDU]
98	4.768383	192.168.1.5	80.[redacted]	TCP	54	7616 > http [ACK] Seq=714 Ack=2825 Win=37994 Len=0
99	4.770962	80.[redacted]	192.168.1.5	TCP	1466	[TCP segment of a reassembled PDU]
100	4.773051	80.[redacted]	192.168.1.5	TCP	1466	[TCP segment of a reassembled PDU]
101	4.773114	192.168.1.5	80.[redacted]	TCP	54	7616 > http [ACK] Seq=714 Ack=5649 Win=37288 Len=0
102	4.775402	80.[redacted]	192.168.1.5	TCP	1466	[TCP segment of a reassembled PDU]
103	4.777952	80.[redacted]	192.168.1.5	TCP	1466	[TCP segment of a reassembled PDU]
104	4.778008	192.168.1.5	80.[redacted]	TCP	54	7616 > http [ACK] Seq=714 Ack=8473 Win=36582 Len=0

```
⊞ Frame 51: 767 bytes on wire (6136 bits), 767 bytes captured (6136 bits)
⊞ Ethernet II, Src: HonHaiPr_ba:f9:5f (78:dd:08:ba:f9:5f), Dst: Giga-Byt_d3:a3:fb (00:0f:ea:d3:a3:fb)
⊞ Internet Protocol Version 4, Src: 192.168.1.5 (192.168.1.5), Dst: 80.[redacted](80.[redacted])
⊟ Transmission Control Protocol, Src Port: 7616 (7616), Dst Port: http (80), Seq: 1, Ack: 1, Len: 713
    Source port: 7616 (7616)
    Destination port: http (80)
    [Stream index: 21]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 714    (relative sequence number)]
    Acknowledgement number: 1    (relative ack number)
    Header length: 20 bytes
  ⊞ Flags: 0x18 (PSH, ACK)
    Window size value: 38700
    [Calculated window size: 38700]
    [Window size scaling factor: -1 (unknown)]
  ⊞ Checksum: 0xaa45 [validation disabled]
```

SSL/TLS

- Wireshark: www.[redacted] (Follow TCP Stream)

```
HTTP/1.1 304 Not Modified
Server: nginx/0.6.32
Date: Fri, 30 Dec 2011 19:30:11 GMT
Last-Modified: Sun, 16 Mar 2008 05:15:11 GMT
Connection: keep-alive

POST /login_verificar.php HTTP/1.1
Host: [redacted]
Connection: keep-alive
Content-Length: 60
Cache-Control: max-age=0
Origin: http:/[redacted]
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.7 (KHTML, like Gecko) Chrome/16.
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http:/[redacted]/index.php
Accept-Encoding: gzip,deflate,sdch
Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.6,en;q=0.4
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
Cookie: PHPSESSID=7280cb8844a750fe4876a9f6644289c1694d75dd; __utma=128716561.333696567.132527335
__utmz=128716561.1325273355.1.1.utmccn=(direct)|utmcsr=(direct)|utmcmd=(none); __utmb=128716561

txtLogin=meulogin&txtSenha=minhasenha&entrar.x=22&entrar.y=7HTTP/1.1 200 OK
Server: nginx/0.6.32
Date: Fri, 30 Dec 2011 19:30:14 GMT
Content-Type: text/html; charset=ISO-8859-1
Connection: keep-alive
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 510
```

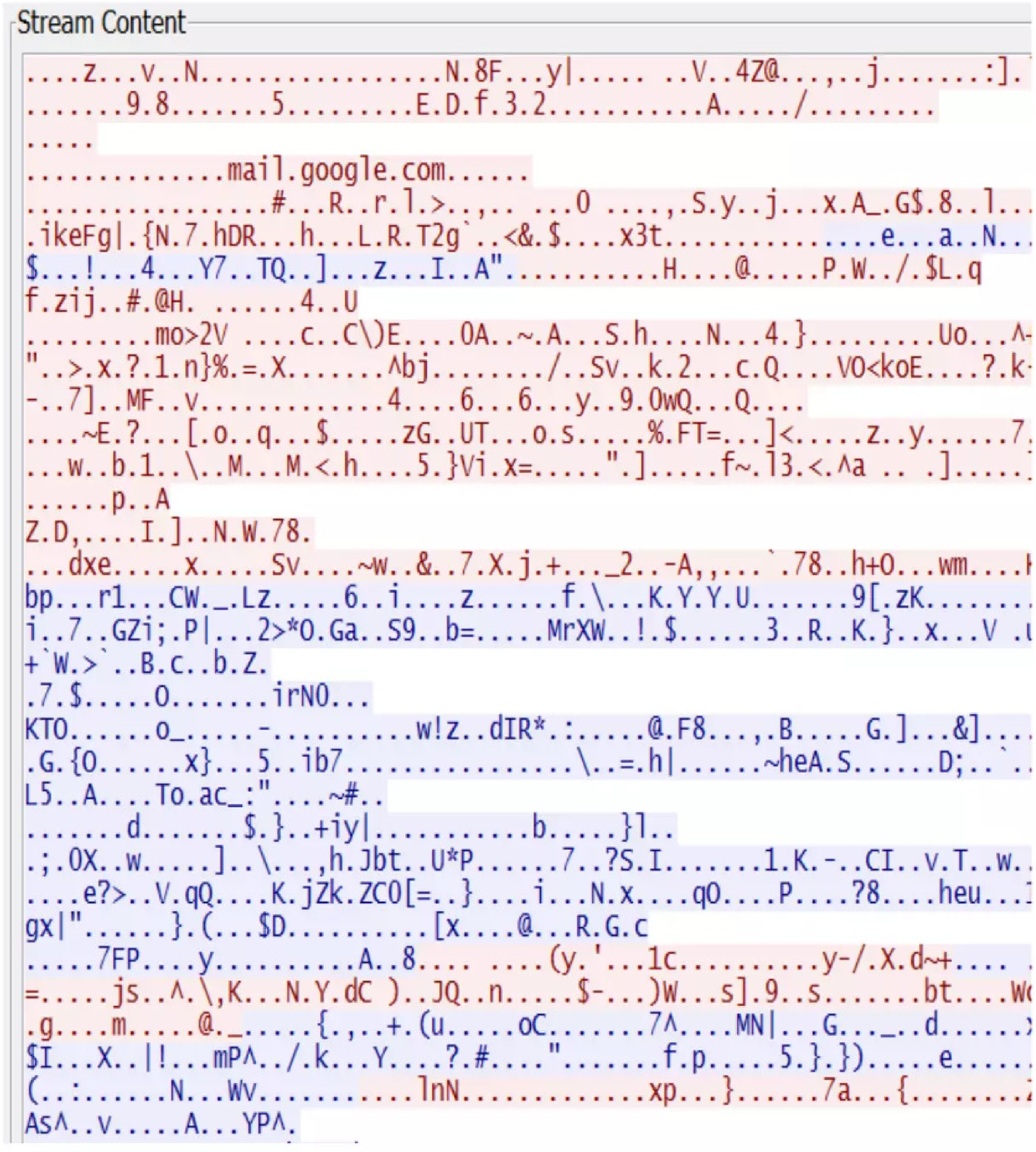
SSL/TLS

- Wireshark: mail.google.com

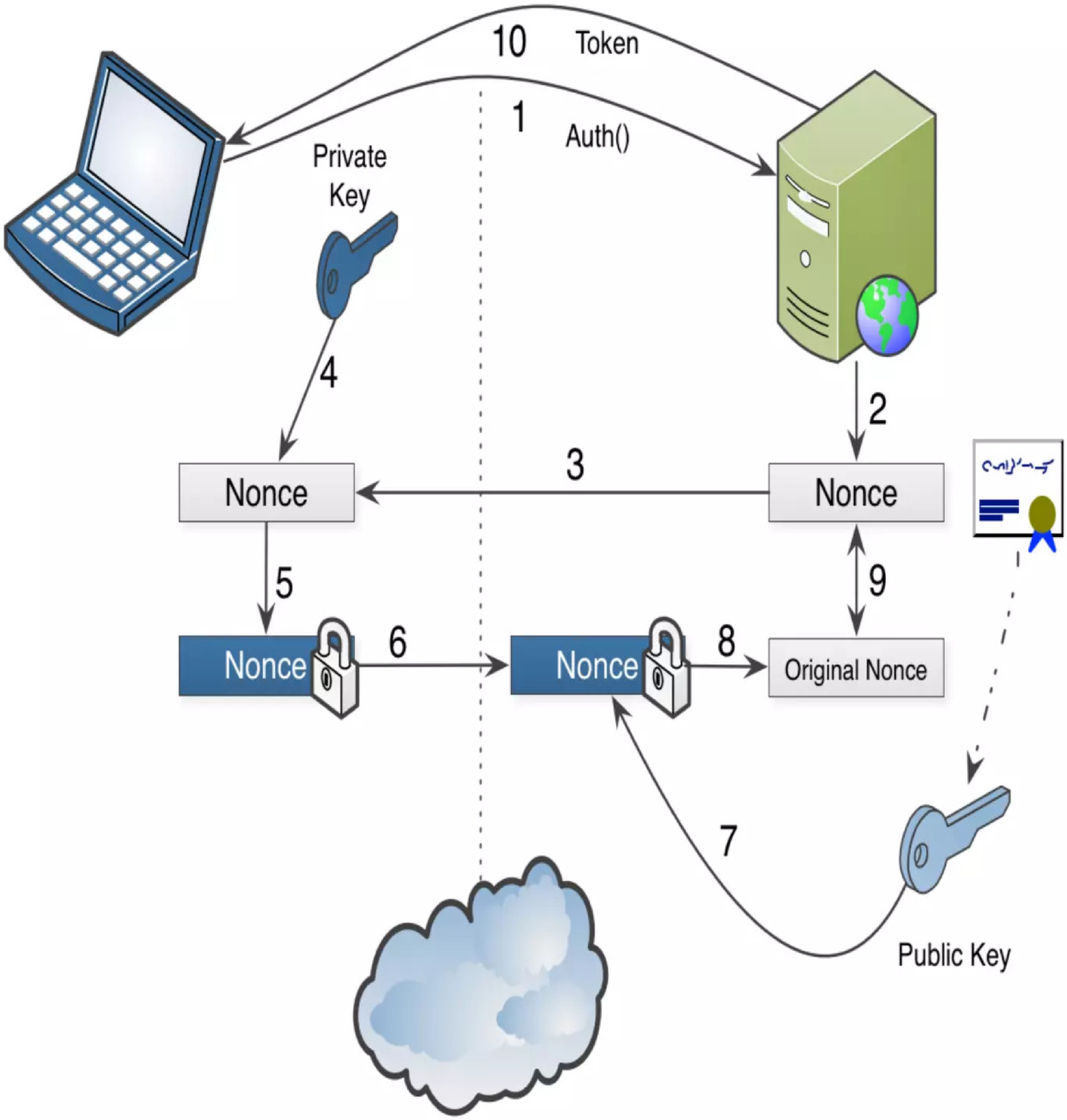
Source	Destination	Protocol	Length	Info
74.125.234.42	192.168.1.5	TLSv1	91	Application Data
192.168.1.5	74.125.234.42	TCP	54	57996 > https [ACK] Seq=1 Ack=38 Win=3984 Len=0
192.168.1.5	192.168.1.1	DNS	75	Standard query A mail.google.com
192.168.1.1	192.168.1.5	DNS	222	Standard query response CNAME googlemail.l.google.com A 74.125
192.168.1.5	74.125.234.54	TCP	66	57997 > https [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PE
192.168.1.5	192.168.1.1	DNS	84	Standard query A www.google-analytics.com
74.125.234.54	192.168.1.5	TCP	66	https > 57997 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1412 S
192.168.1.5	74.125.234.54	TCP	54	57997 > https [ACK] Seq=1 Ack=1 Win=16944 Len=0
192.168.1.5	74.125.234.54	TLSv1	437	Client Hello
192.168.1.1	192.168.1.5	DNS	456	Standard query response CNAME www-google-analytics.l.google.co
192.168.1.5	74.125.234.40	TCP	66	57998 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PE
74.125.234.54	192.168.1.5	TCP	60	https > 57997 [ACK] Seq=1 Ack=384 Win=6848 Len=0
74.125.234.54	192.168.1.5	TLSv1	207	Server Hello, Change Cipher Spec, Encrypted Handshake Message
192.168.1.5	74.125.234.54	TLSv1	137	Change Cipher Spec, Encrypted Handshake Message
192.168.1.5	74.125.234.54	TLSv1	107	Application Data
192.168.1.5	74.125.234.54	TLSv1	708	Application Data
74.125.234.40	192.168.1.5	TCP	66	http > 57998 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1412 SA
192.168.1.5	74.125.234.40	TCP	54	57998 > http [ACK] Seq=1 Ack=1 Win=16944 Len=0
74.125.234.54	192.168.1.5	TLSv1	99	Application Data

SSL/TLS

- Wireshark: mail.google.com (Follow TCP Stream)



Authentication (Digital certificate)



Authentication (Digital certificate)

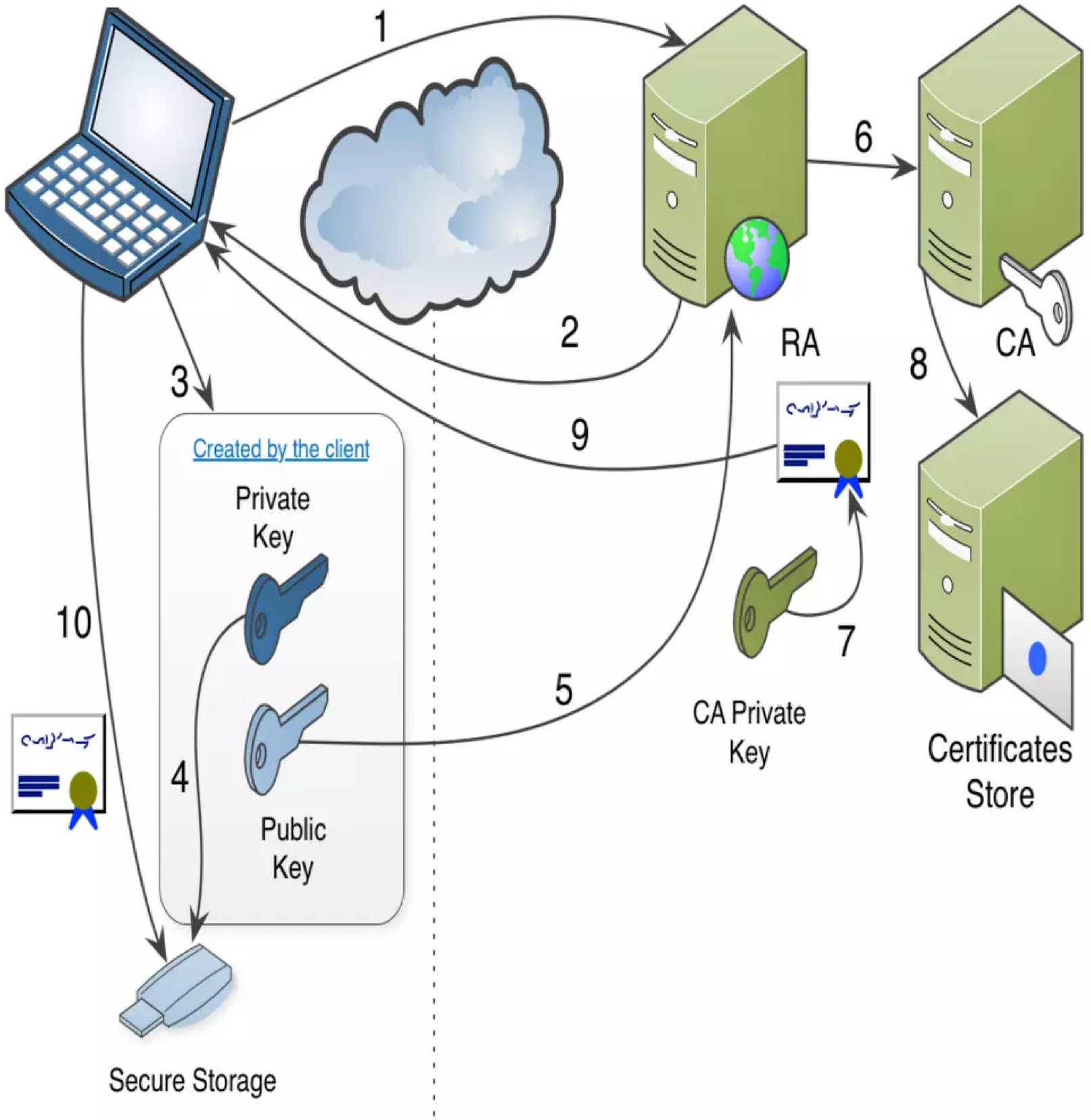
- Encryption and decryption
  1. Client tries to authenticate
  2. Server generates a Nonce
  3. Nonce is sent to the client
  4. Client uses its private key to encrypt Nonce
  5. Encrypted Nonce is obtained
  6. Encrypted Nonce is sent to the server
  7. Server uses client's public key (from client's certificate) to decrypt the Nonce
  8. Original Nonce is obtained
  9. Comparison between both Nonces

Public Key Infrastructure (PKI)

- Digital certificate real world use
  - E-mail
  - VPN
  - SSL/HTTPS
  - Authentication
  - File encryption (e.g. EFS)
  - Digital signature of files

Public Key Infrastructure (PKI)

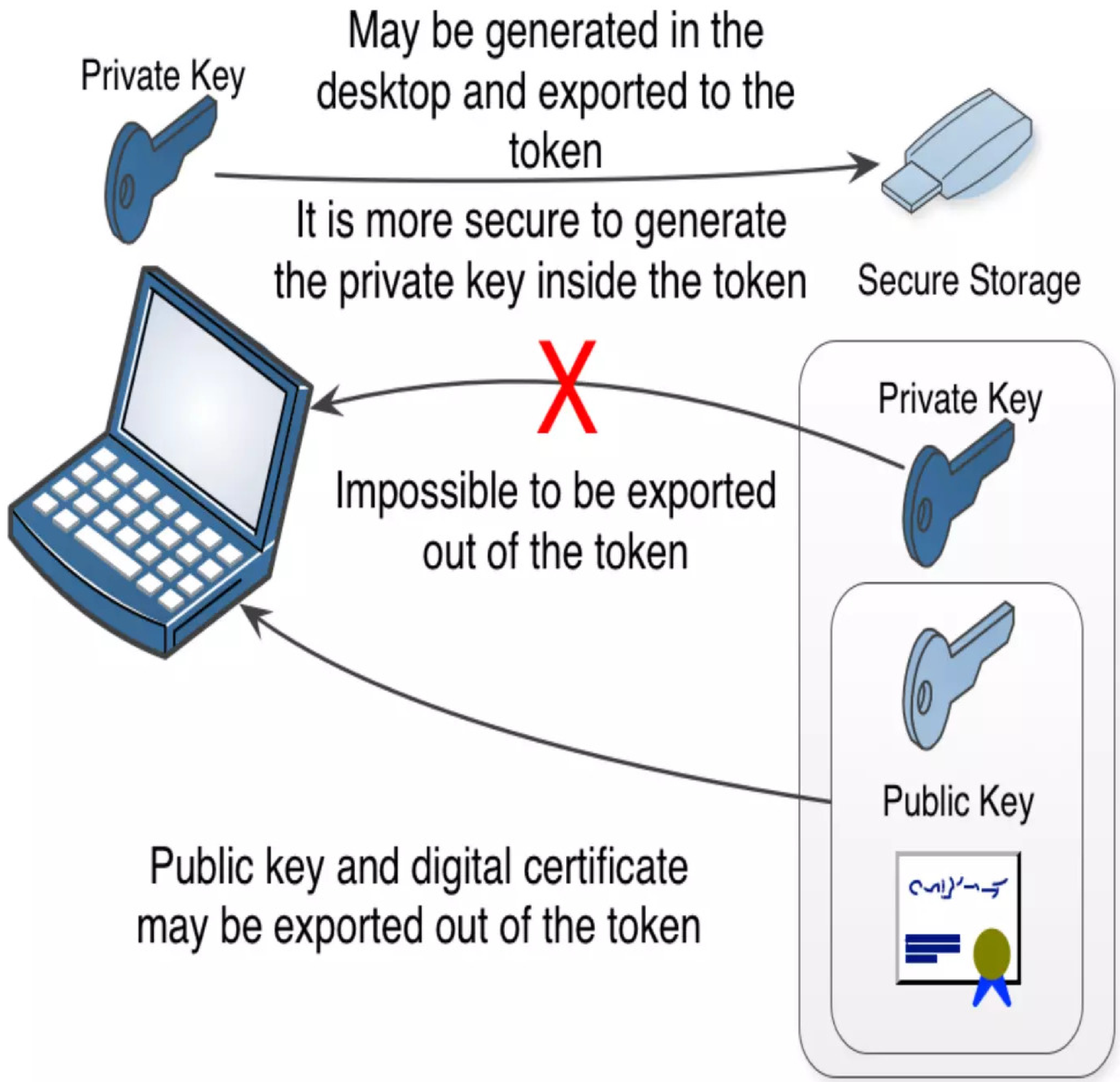
- Digital certificate issuing process



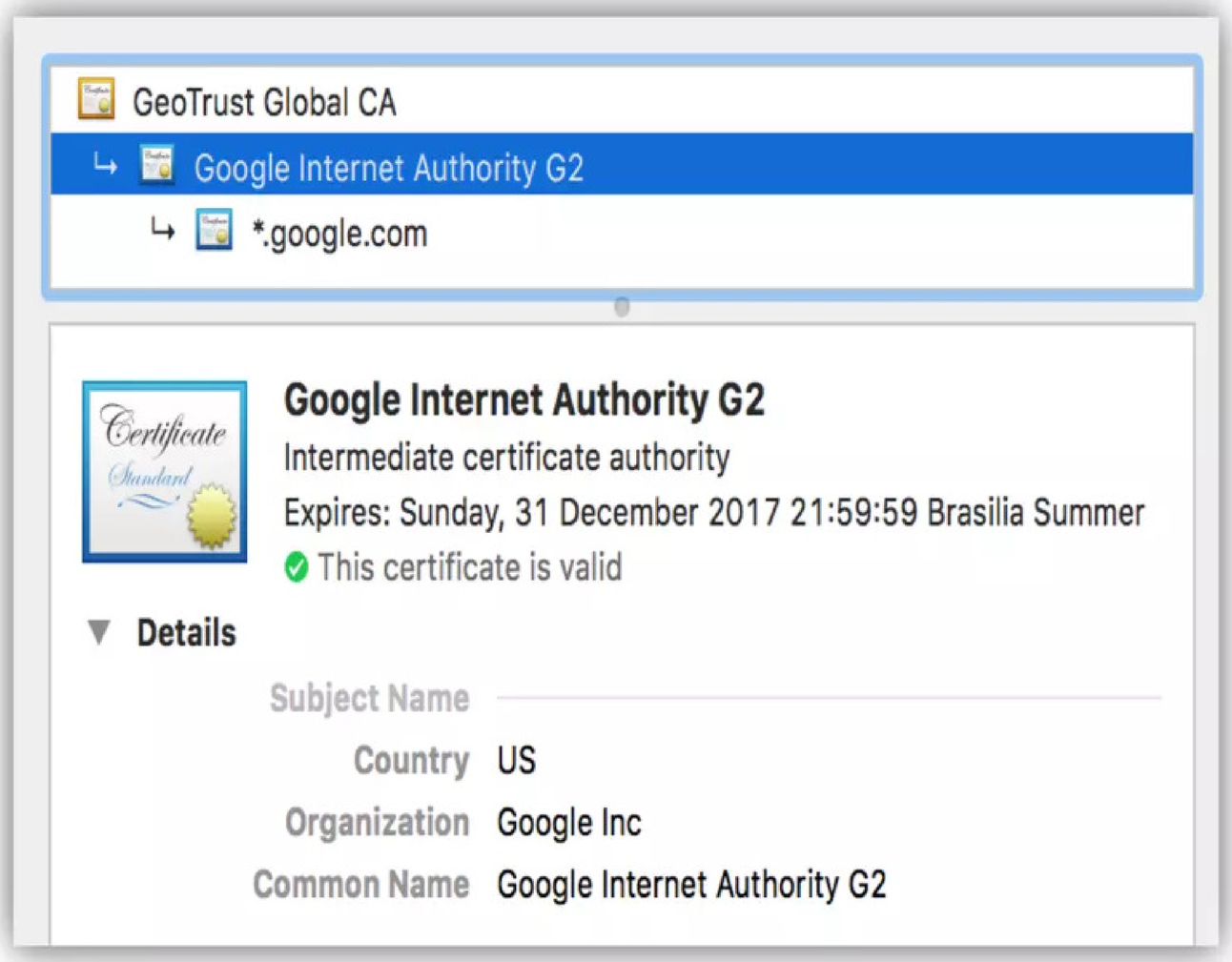
Public Key Infrastructure (PKI)

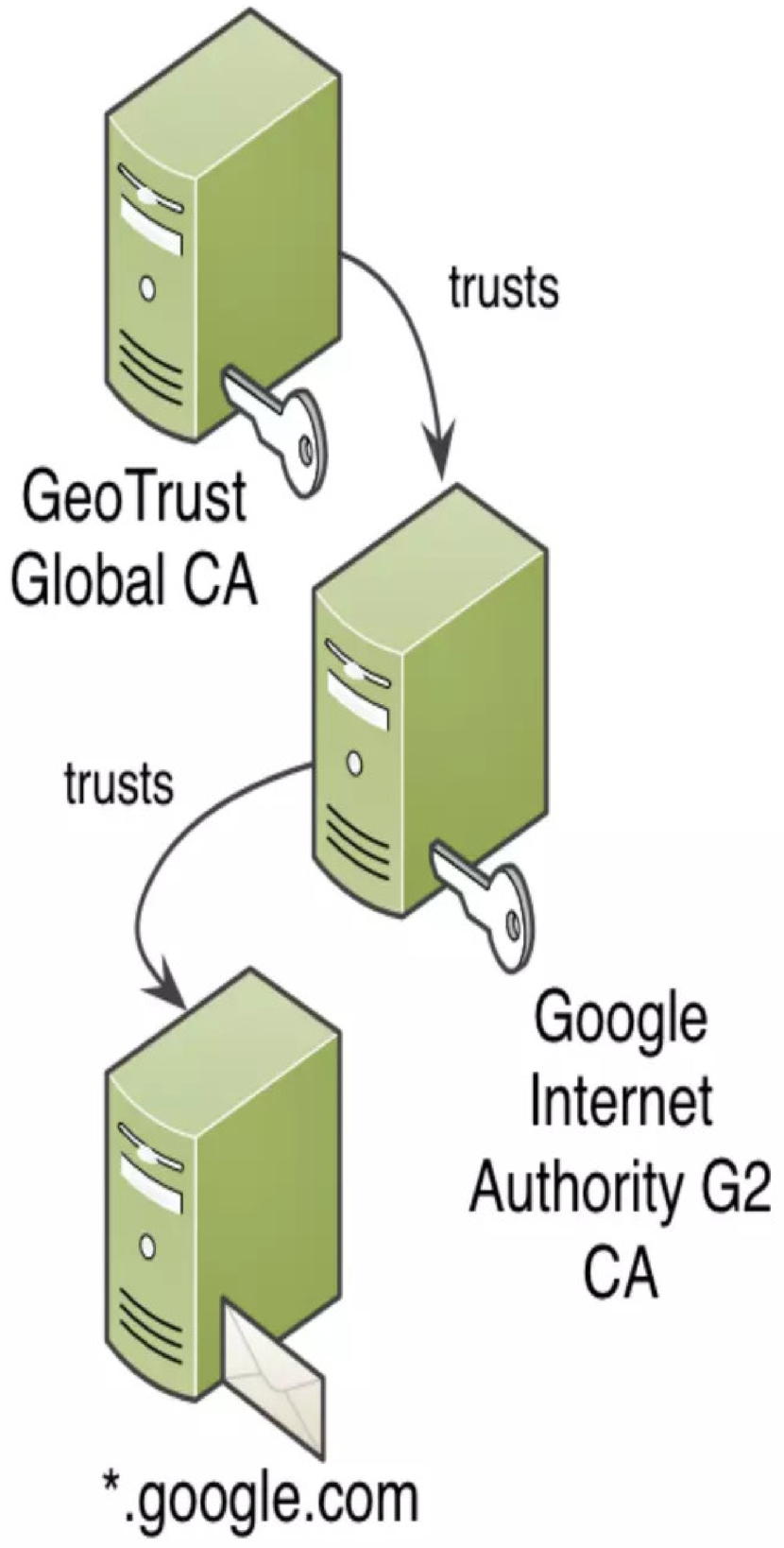
- Issuing process
  1. Client asks Registration Authority for a certificate
  2. RA tells client to generate the pair of keys
  3. Keys are generated by the client
  4. The private key is generated inside the token or exported to the token
  5. The public key is sent to the RA
  6. RA sends the public key and client's information to CA
  7. CA generates a digital certificate and signs it with its own private key
  8. The certificate is stored in Certificates Store
  9. The certificate is sent to the client
  10. The client stores the certificate inside the token

Public Key Infrastructure (PKI)

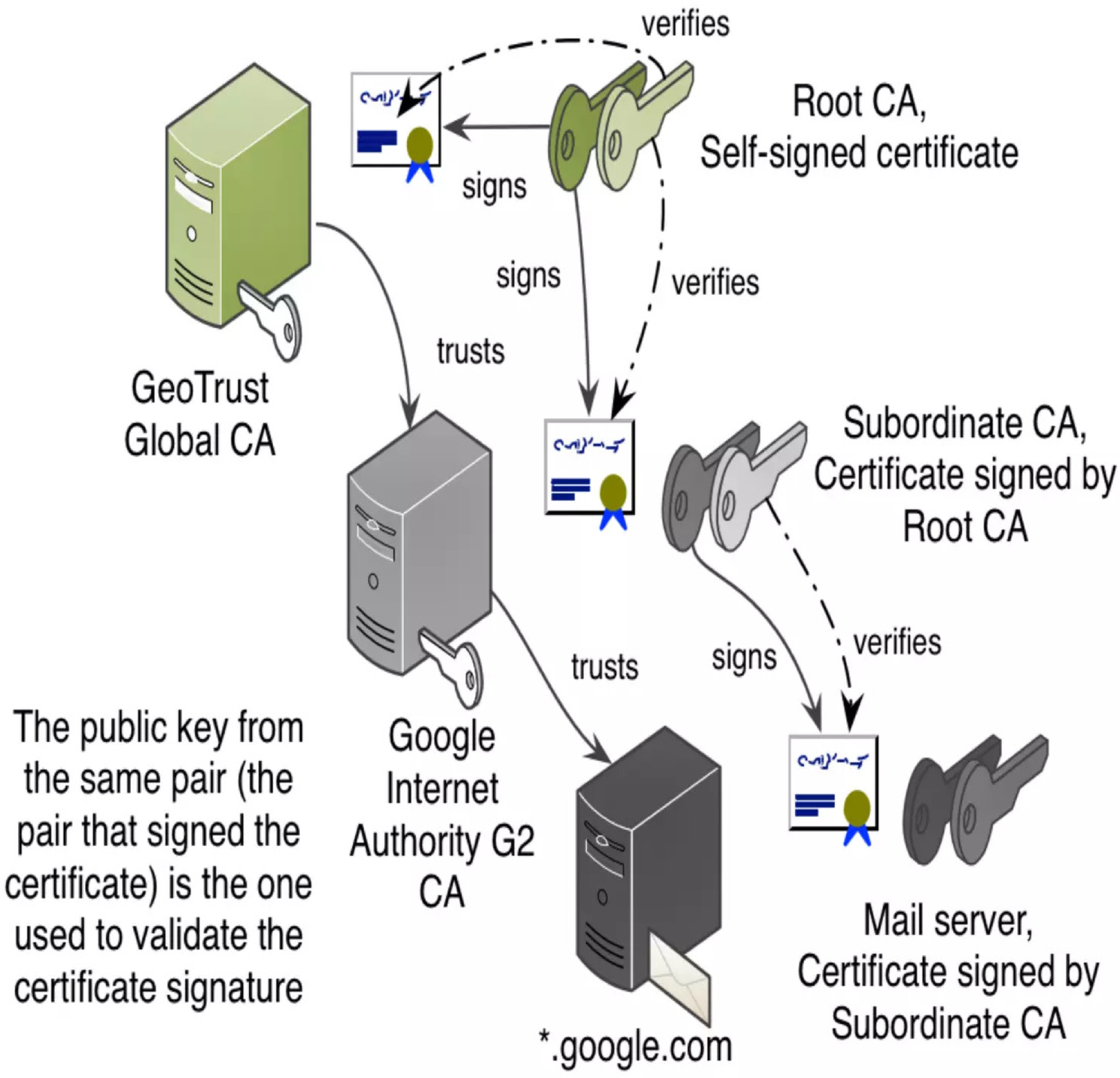


Public Key Infrastructure (PKI)

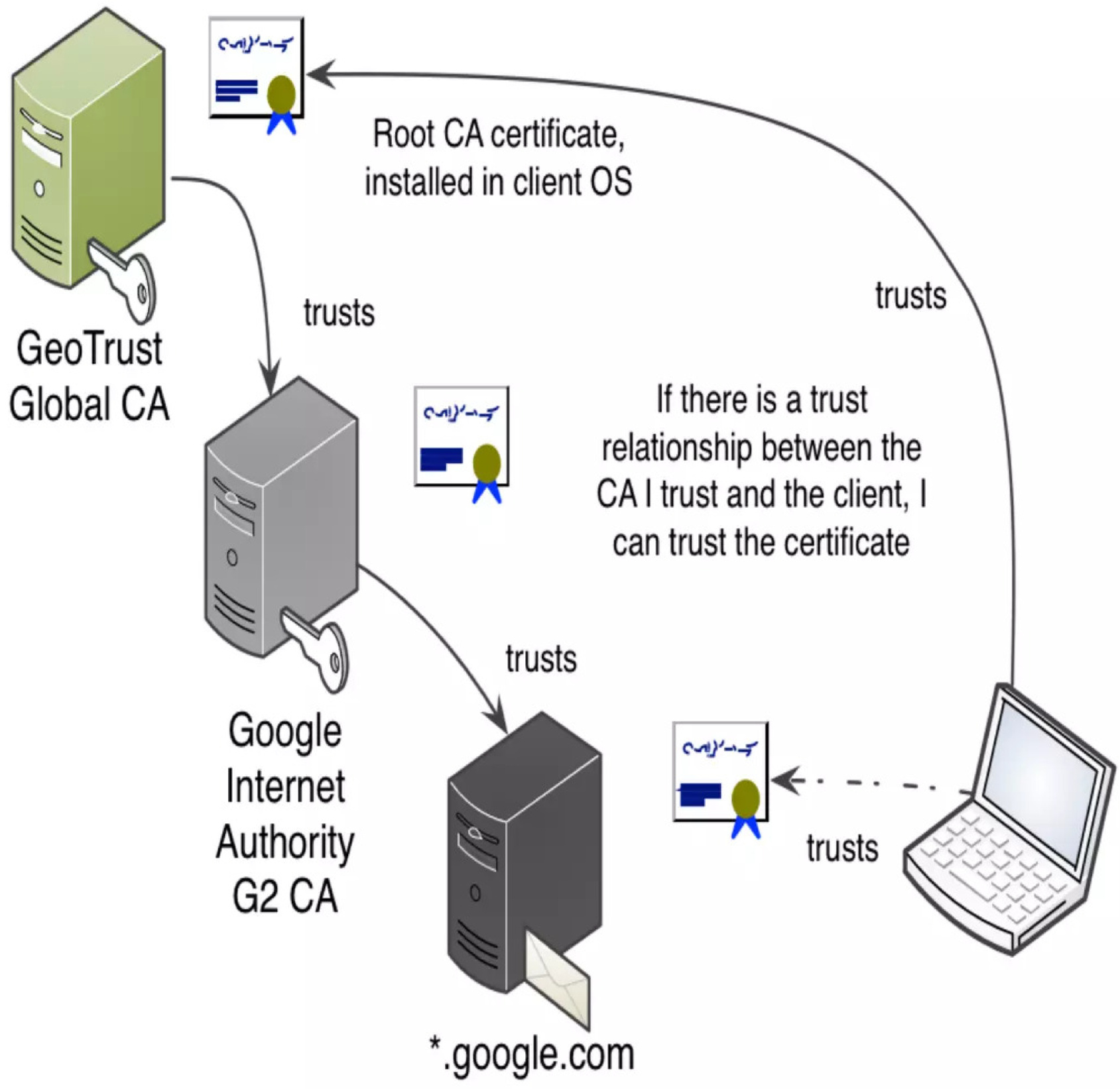




Public Key Infrastructure (PKI)



Public Key Infrastructure (PKI)



Public Key Infrastructure (PKI)

- Server-based Certification Validation Protocol (SCVP)
  - The certification path creation process is called *discovery path*
  - The application receiving the message is responsible for *discovery path*
  - Many applications use MS CAPI (Microsoft Crypto API)
  - It is not possible to create the chain of certificates (discovery path) if one of the certificates is not found

Public Key Infrastructure (PKI)

- Root certificate trust program
  - Microsoft Trusted Root Certificate Program
    - http://technet.microsoft.com/en-us/library/cc751157.aspx
  - Apple Root Certificate Program
    - http://www.apple.com/certificateauthority/ca_program.html
    - Included in iOS 10: https://support.apple.com/en-us/HT207177
  - Google Chromium
    - https://www.chromium.org/Home/chromium-security/root-ca-policy
  - OpenSSL: unavailable
  - Mozilla CA Certificate Store
    - http://www.mozilla.org/projects/security/certs/
  - Opera
    - http://www.opera.com/docs/ca/
  - Mono (open source .NET framework): unavailable
    - Reason: http://www.mono-project.com/FAQ:_Security

Public Key Infrastructure (PKI)

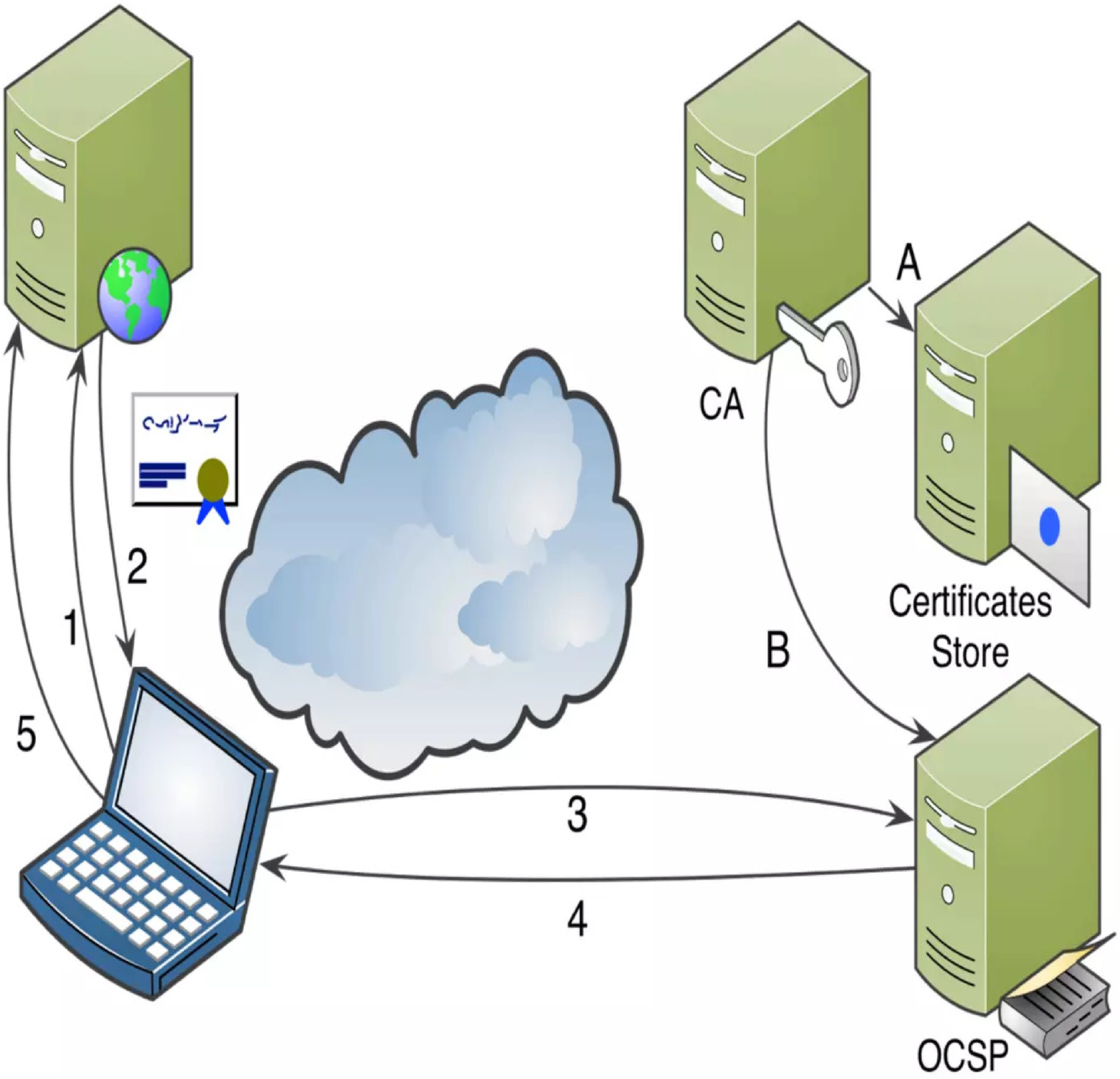
- Root certificate inclusion program
  - General rules
    - No fee
    - ISO 21188:2006 - Public key infrastructure for financial services -- Practices and policy framework
    - NIST SP 800-57 - Recommendation for Key Management – Part 1: General
    - Compliance with WebTrust Program for Certification Authorities
      - http://www.webtrust.org/homepage-documents/item27839.aspx
    - RFC 3280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

Public Key Infrastructure (PKI)

- Digital certificate verification process
  - OCSP (Online Certificate Status Protocol)
    - Consults OCSP Responder through protocol
    - Sends only the serial number to be checked
    - The request is forwarded in real time (or close to)
    - Do not expose the serial number of all certificates
    - Send less information, sparing server, network and client
    - Works well if the client caches the response
  - CRL (Certificate Revocation List)
    - Downloads the entire CRL (automatic process)
    - A list may contain 500KB
    - Usually updated every 24 hours or less
    - When the list changes it has to be downloaded again
    - Is digitally signed by CA to avoid forfeitures

Public Key Infrastructure (PKI)

- Digital certificate verification process
  - OCSP



Public Key Infrastructure (PKI)

- OCSP verification process
  1. Client tries to access web server
  2. Web server sends him its digital certificate
  3. Client (operating system) looks inside the certificate for OCSP server and contacts it asking for information on that certificate serial number
  4. OCSP responds
  5. If it's all good the client continues the connection

  A. The certificate is stored in Certificates Store
  B. The information about certificate serial number is sent to OCSP server

Public Key Infrastructure (PKI)

- Digital certificate verification process
  - OCSP support



IE 7+ (except on XP), automatic



Opera 8.0+, automatic



All versions, automatic



macOS. Supported by all versions, but require manual activation until 10.7



All versions,
automatic starting on 3+

Public Key Infrastructure (PKI)

- Digital certificate verification process
  - OCSP support



MS Outlook 2016:Yes



Apple Mail:Yes



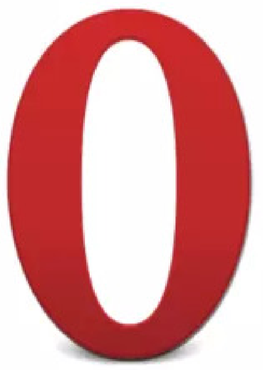
Mozilla Thunderbird:Yes



IBM Lotus Notes:Yes



Seamonkey:Yes



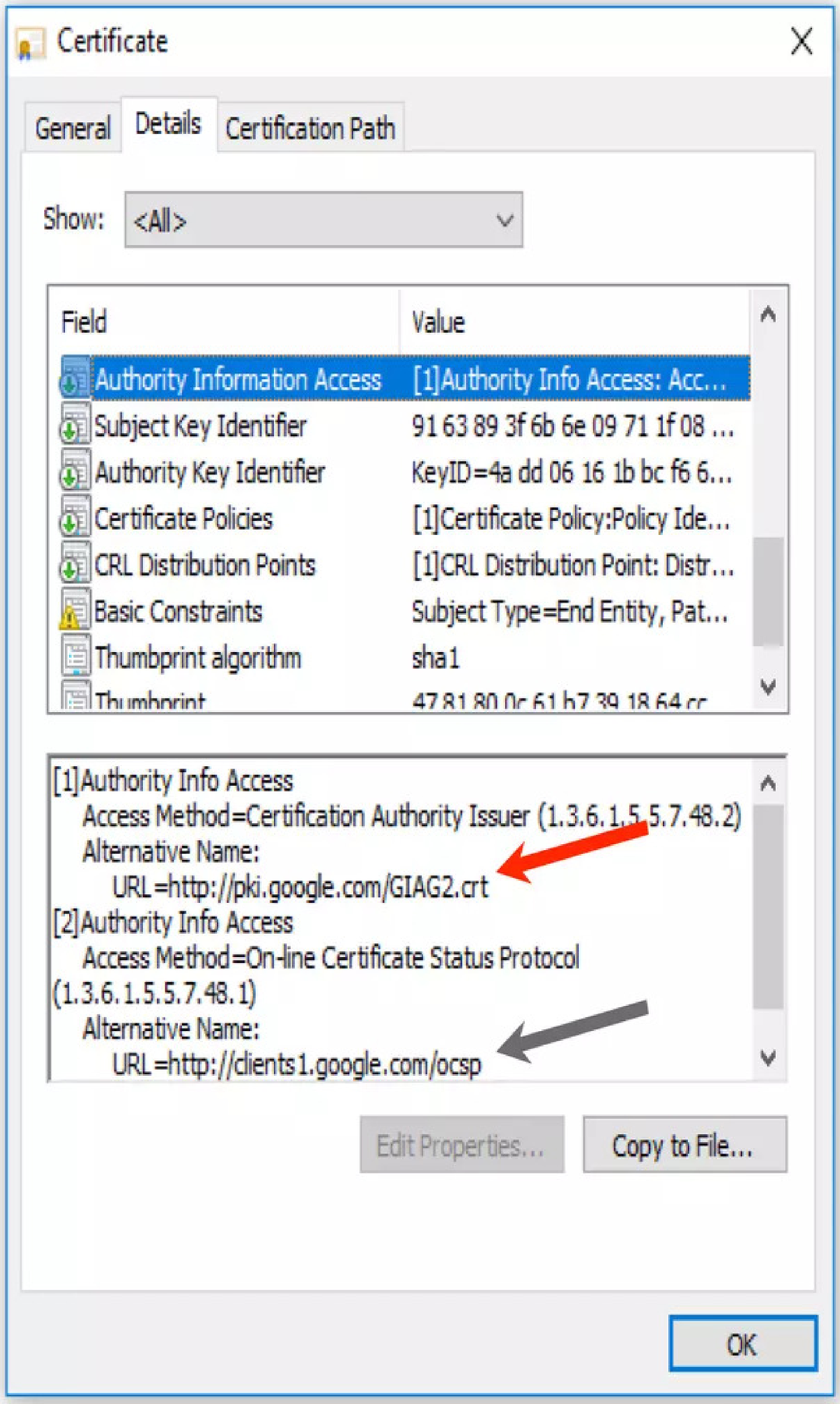
Opera Mail:Yes



The Bat:Yes

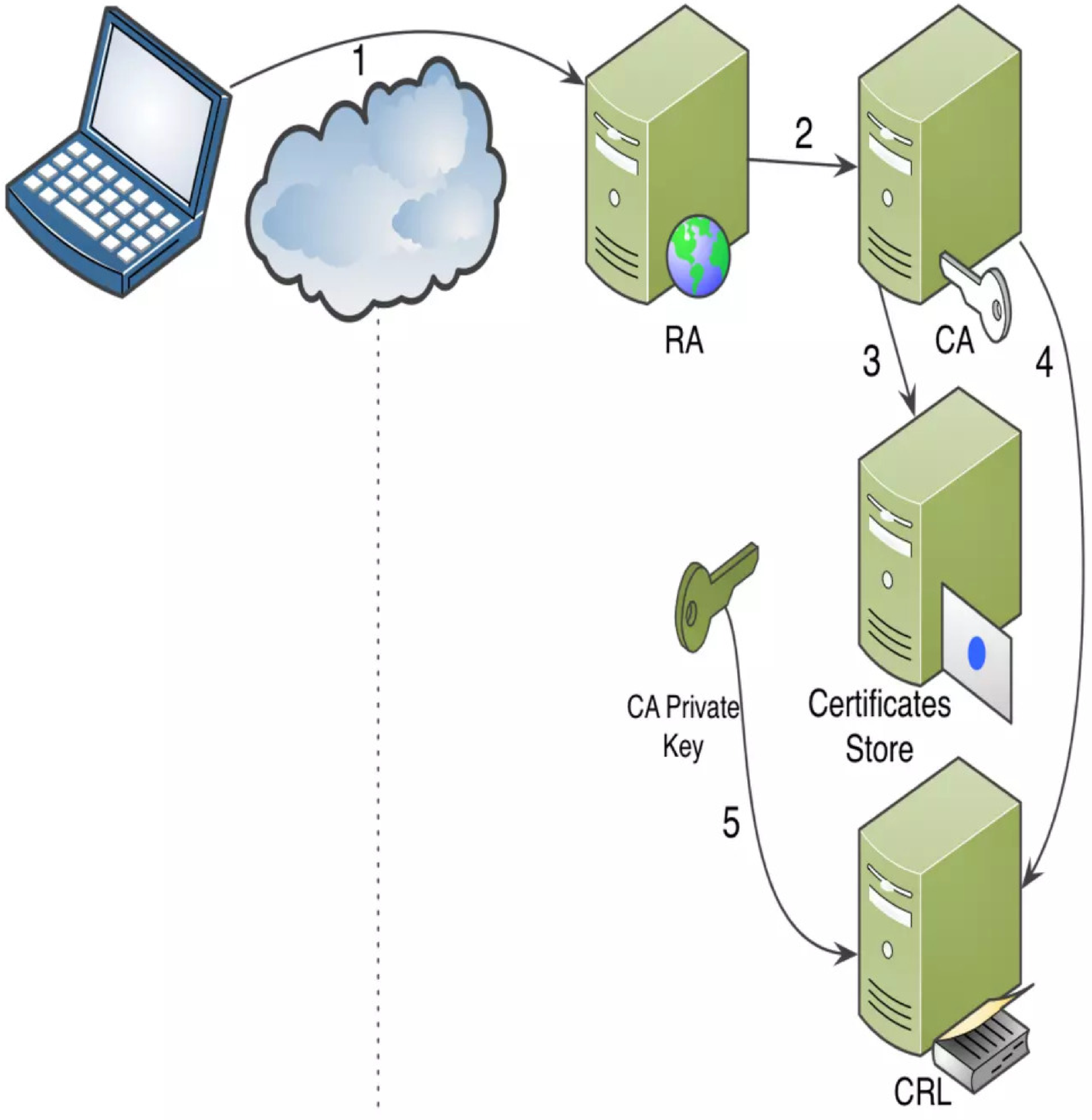
Public Key Infrastructure (PKI)

- Digital certificate verification process (OCSP)



Public Key Infrastructure (PKI)

- Digital certificate revocation process (CRL)

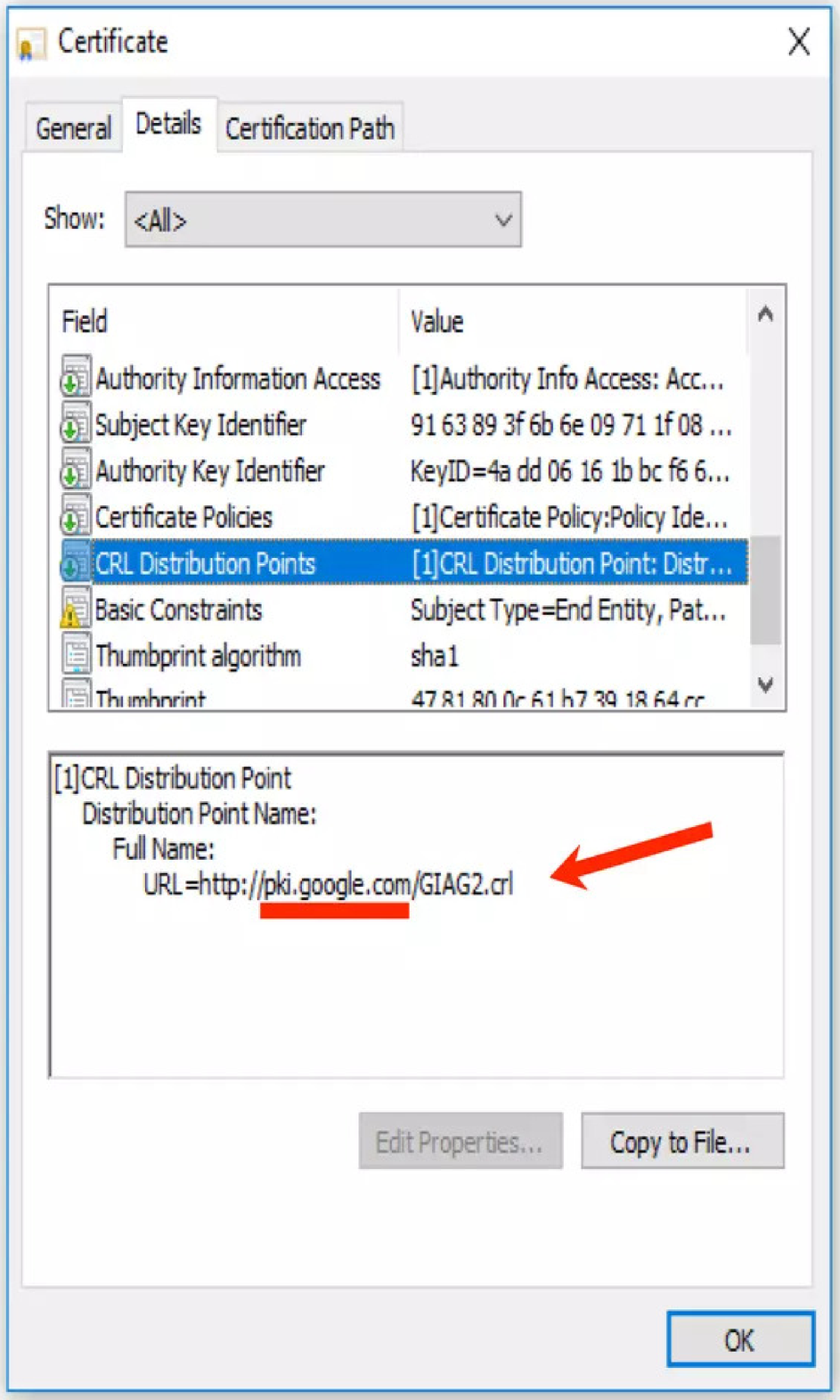


Public Key Infrastructure (PKI)

- CRL, revocation process
  1. Client asks RA to revoke certificate
  2. RA, after its verification process, sends the information to the CA
  3. CA accepts the revocation and stores this information in the Certificates Store
  4. CA includes the serial number from revoked certificate in the CRL
  5. CRL is digitally signed using CA's private key

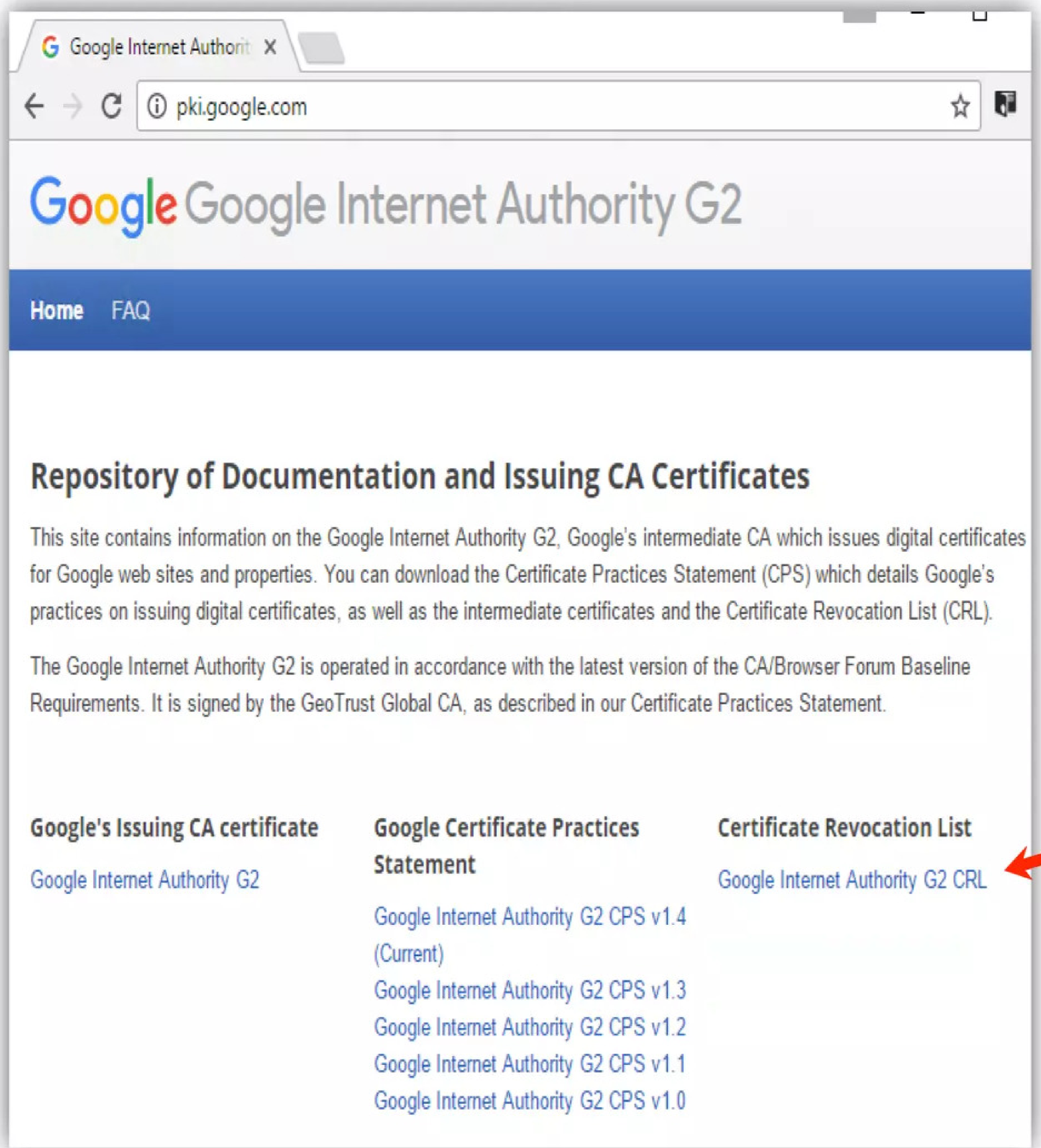
Public Key Infrastructure (PKI)

- Digital certificate verification process (CRL)



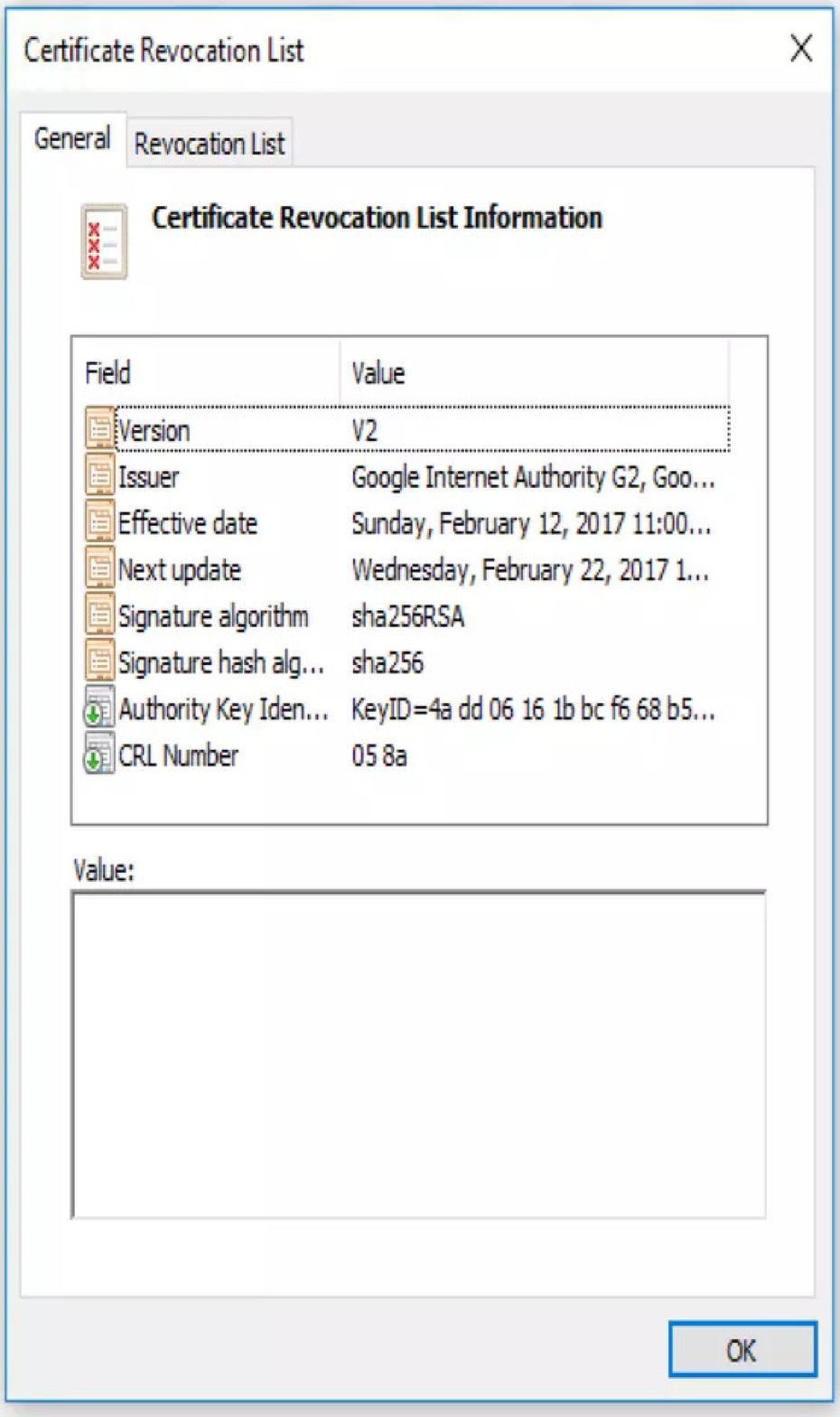
Public Key Infrastructure (PKI)

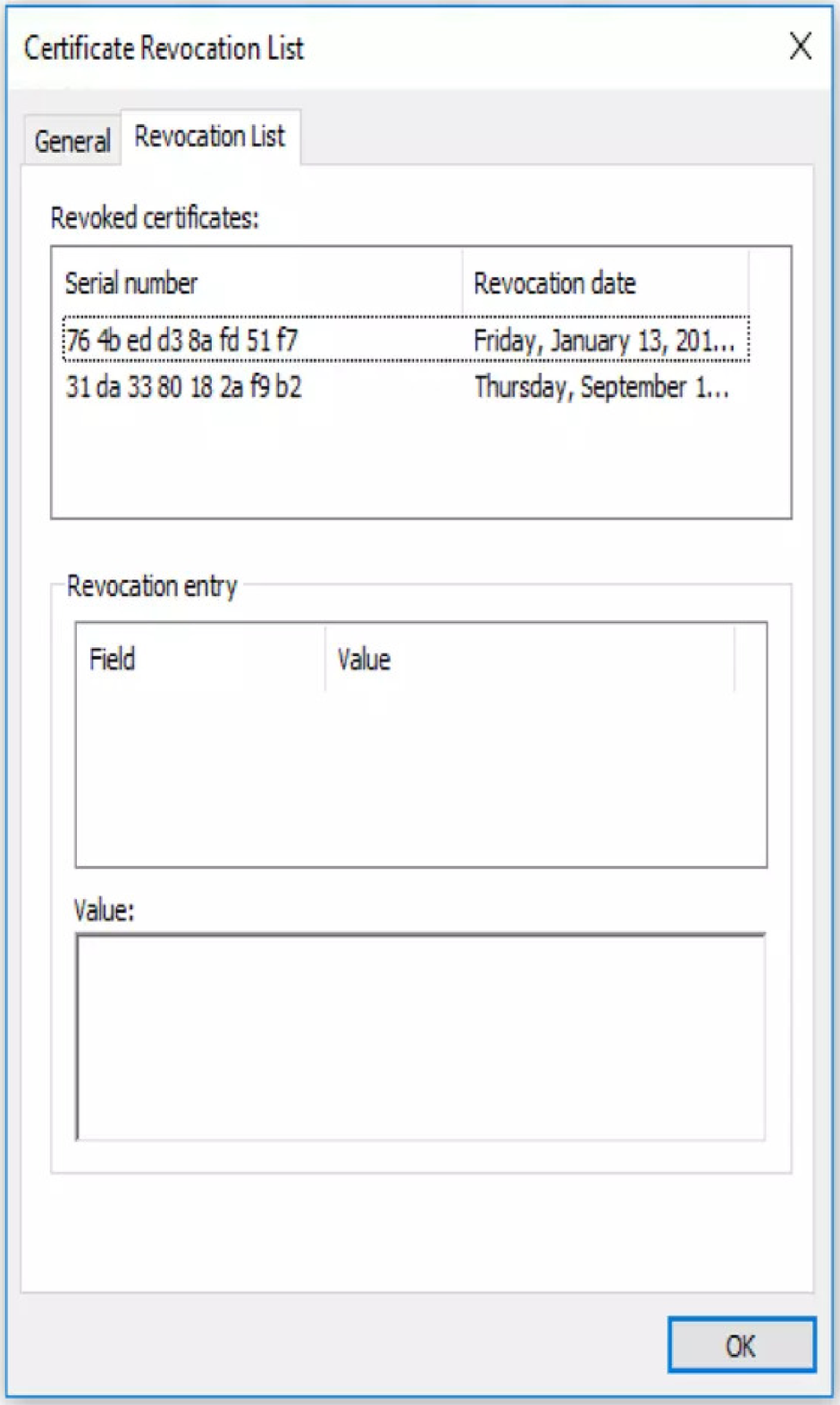
- Digital certificate verification process (CRL)



Public Key Infrastructure (PKI)

- Digital certificate verification process (CRL)





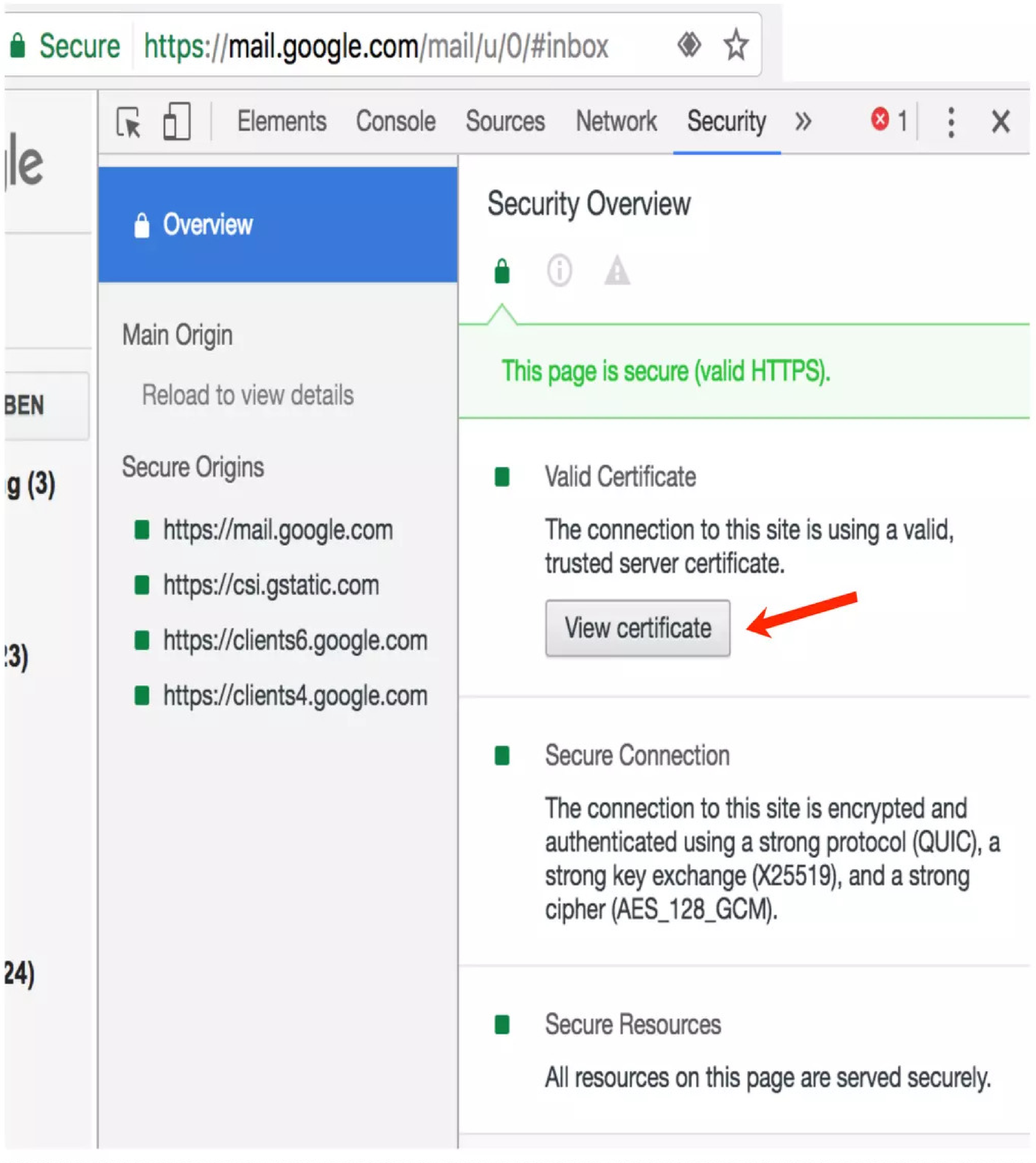
Public Key Infrastructure (PKI)

- Wildcard certificate
  - Certificate issued to protect several domains or servers
  - More practical and economical
  - Example: Certificate issued for *.example.com
    - Works for: payment.example.com, contact.example.com, intranet.example.com, etc.

Public Key Infrastructure (PKI)

- EV-SSL (Extended Validation)
  - 'Complete' validation of requester's ID
  - Reason: commercial pressure made CAs offer "domain validation only" certificate
  - Criteria for issuing EV-SSL
    - Verification of documents and physical presence from someone of the enterprise
    - Control over domain is exclusive to the owner
    - Confirm ID and authority of owners of the website
  - Supported by Microsoft IE 7+ (except XP), Mozilla Firefox 3+, Opera 8+, Apple Safari 3.2+, Google Chrome

Public Key Infrastructure (PKI)



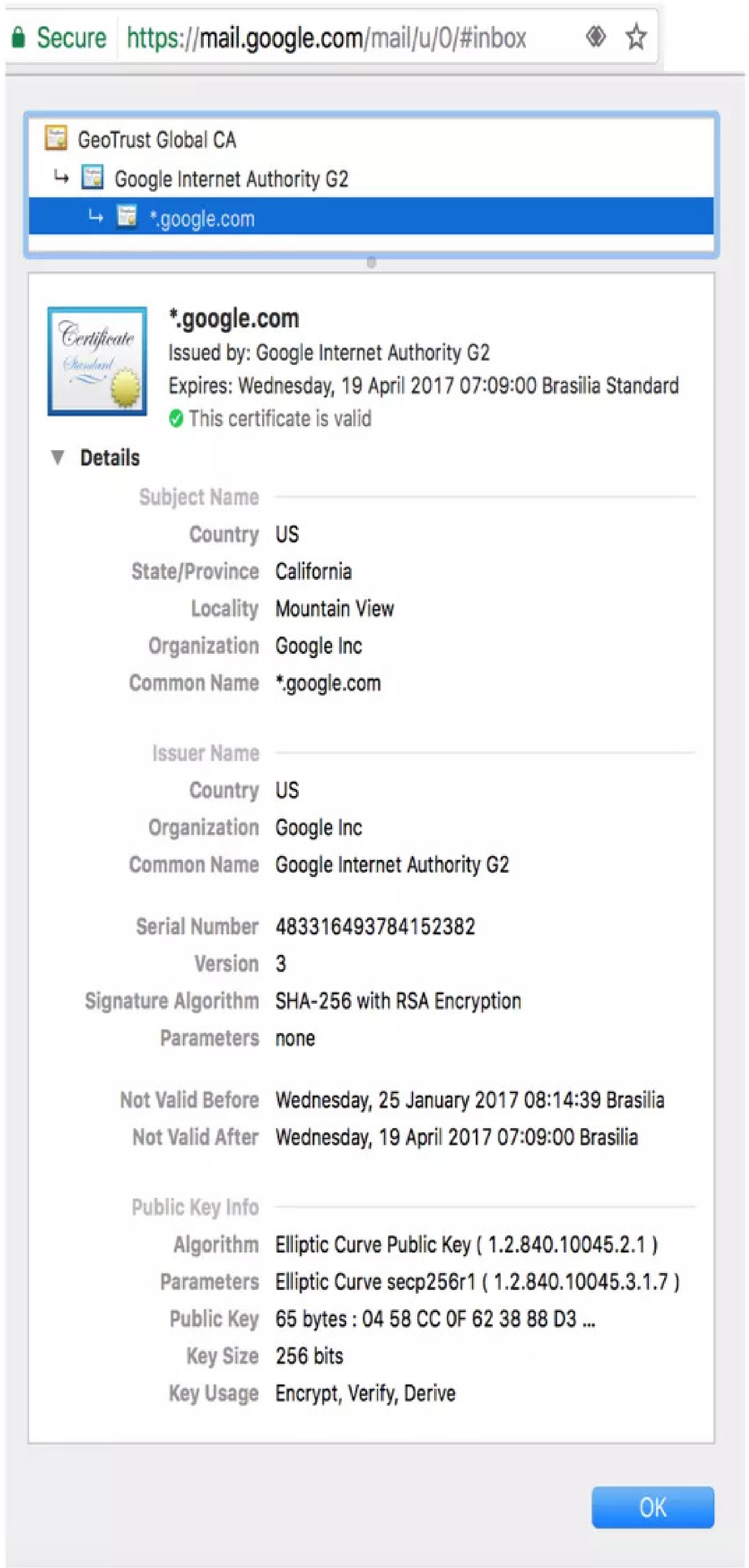


Google Chrome

Mac: Cmd+Opt+I
Win: Ctrl+Alt+I

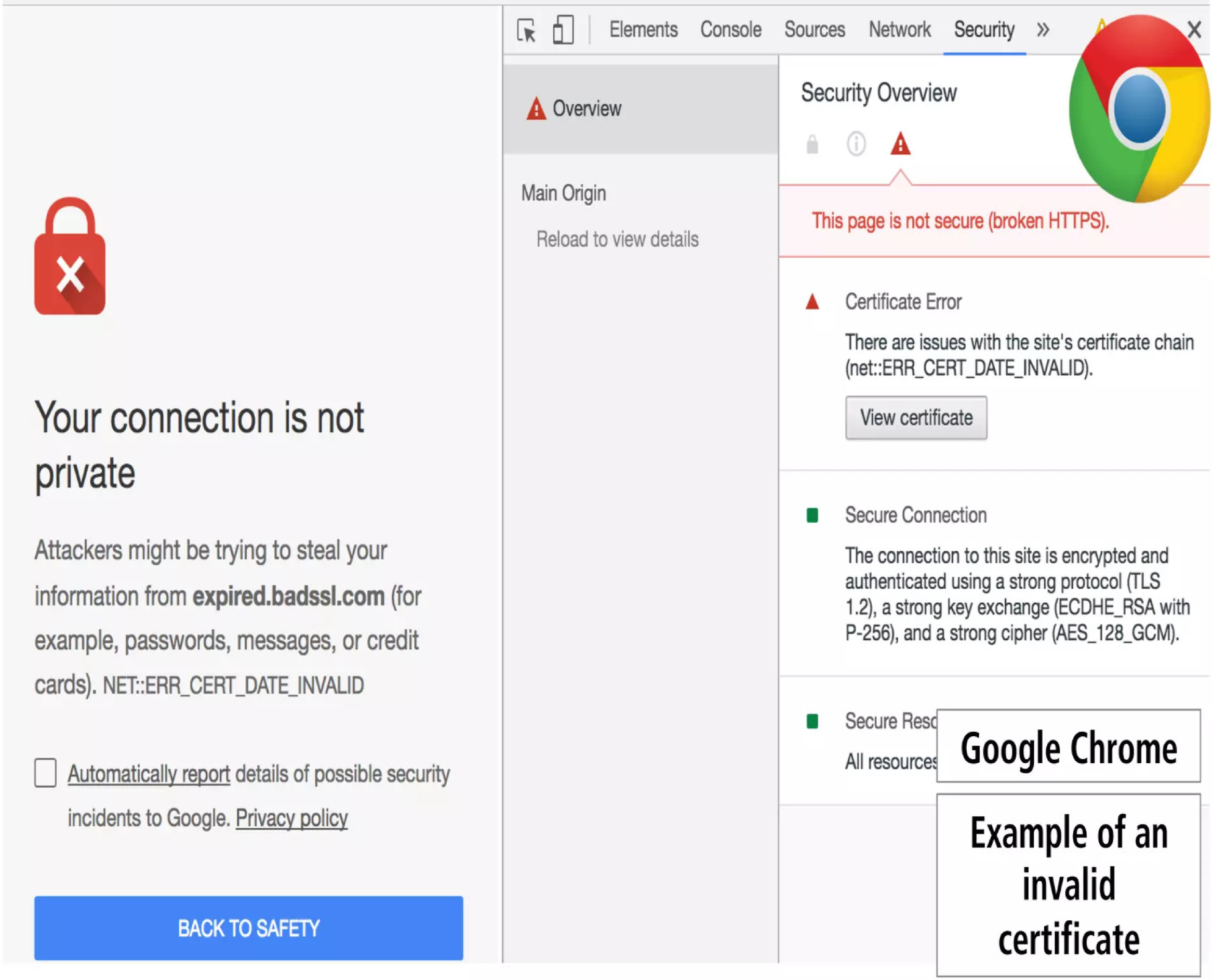
Click on View Certificate

Public Key Infrastructure (PKI)

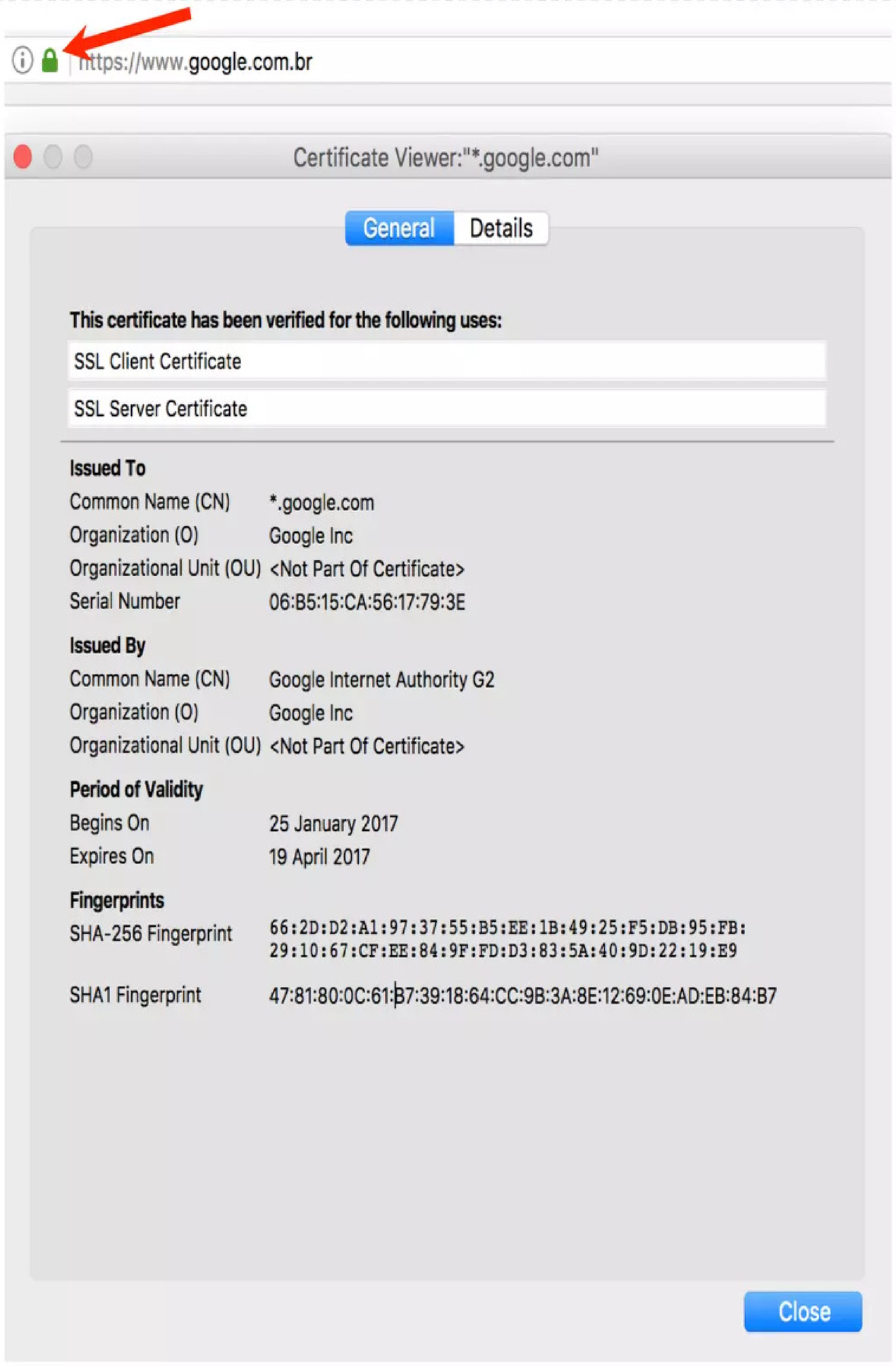


Google Chrome: View certificate

Public Key Infrastructure (PKI)

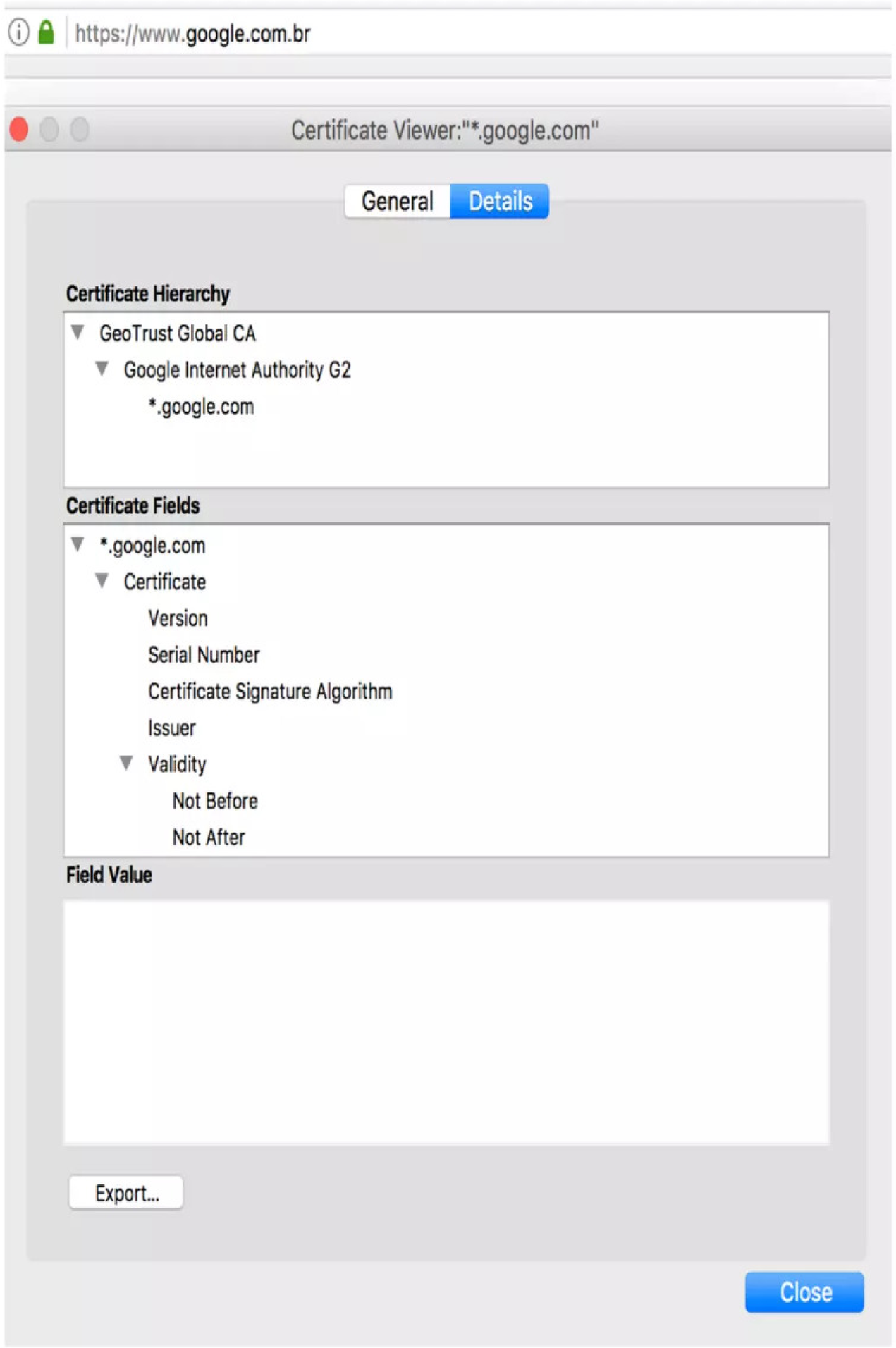


Public Key Infrastructure (PKI)



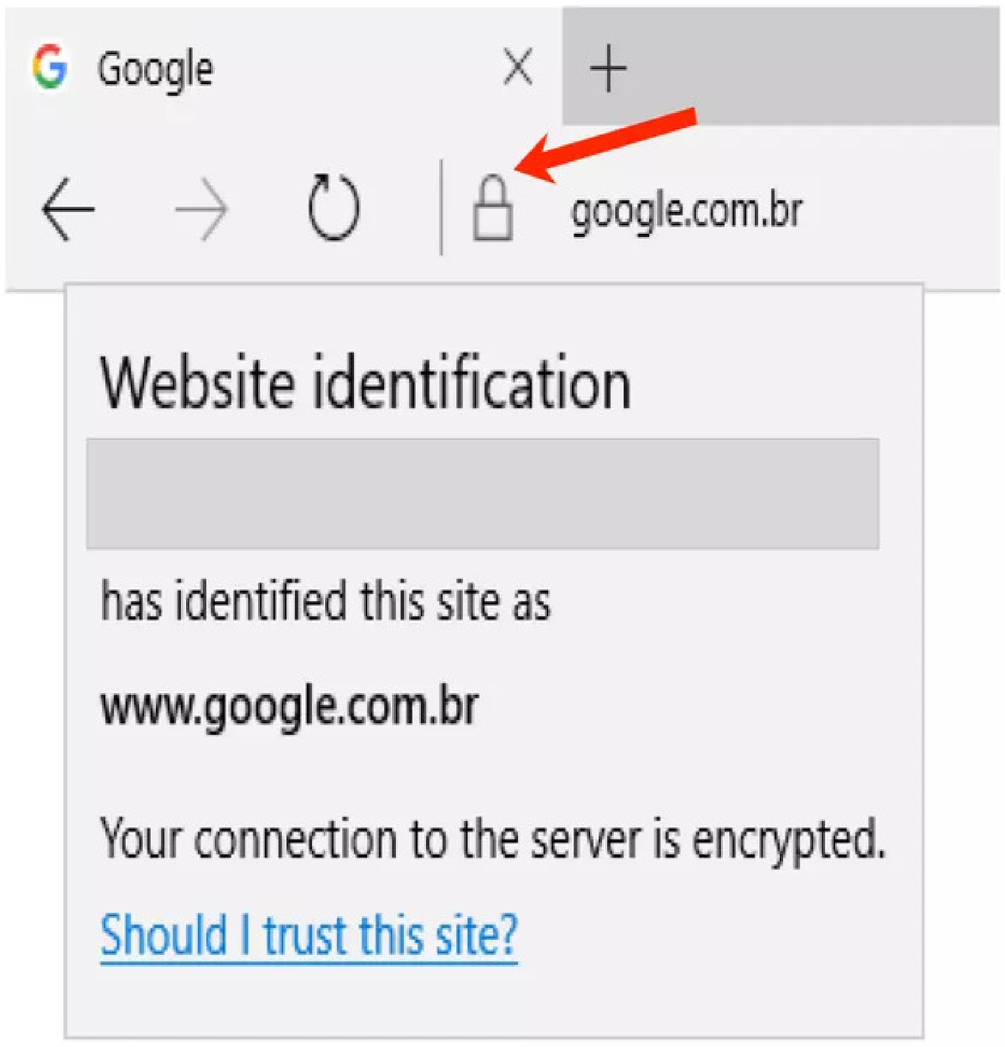
Mozilla Firefox: Certificate Viewer

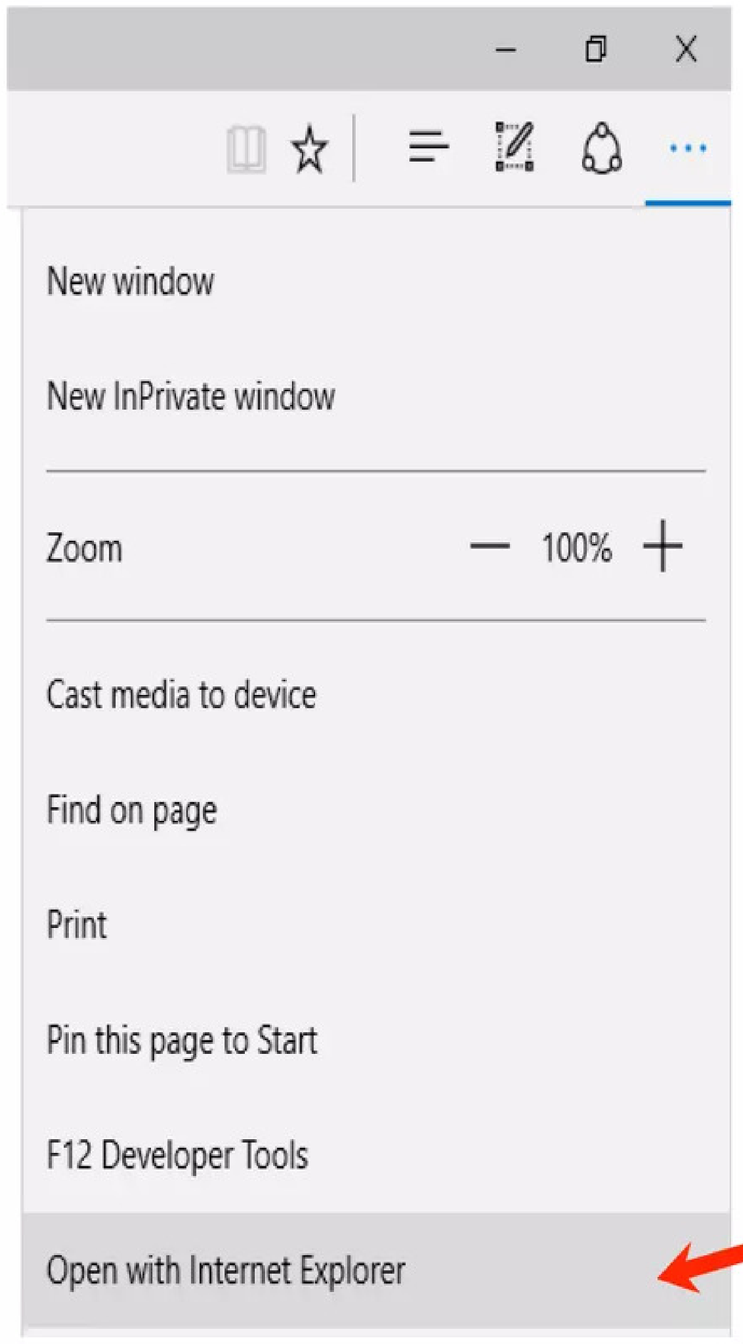
Public Key Infrastructure (PKI)



Mozilla Firefox: Certificate Viewer

Public Key Infrastructure (PKI)

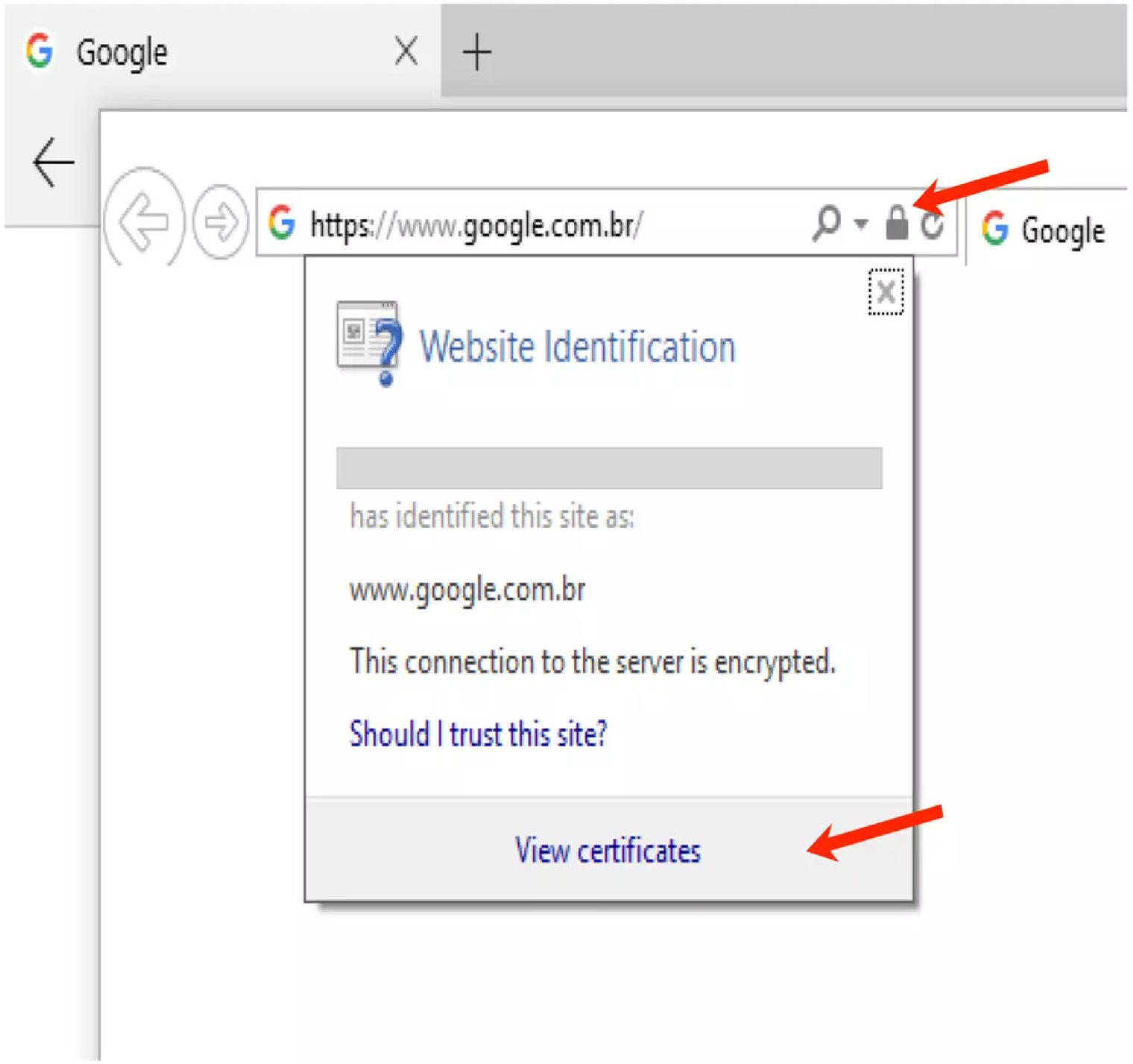




Microsoft Edge: You don't

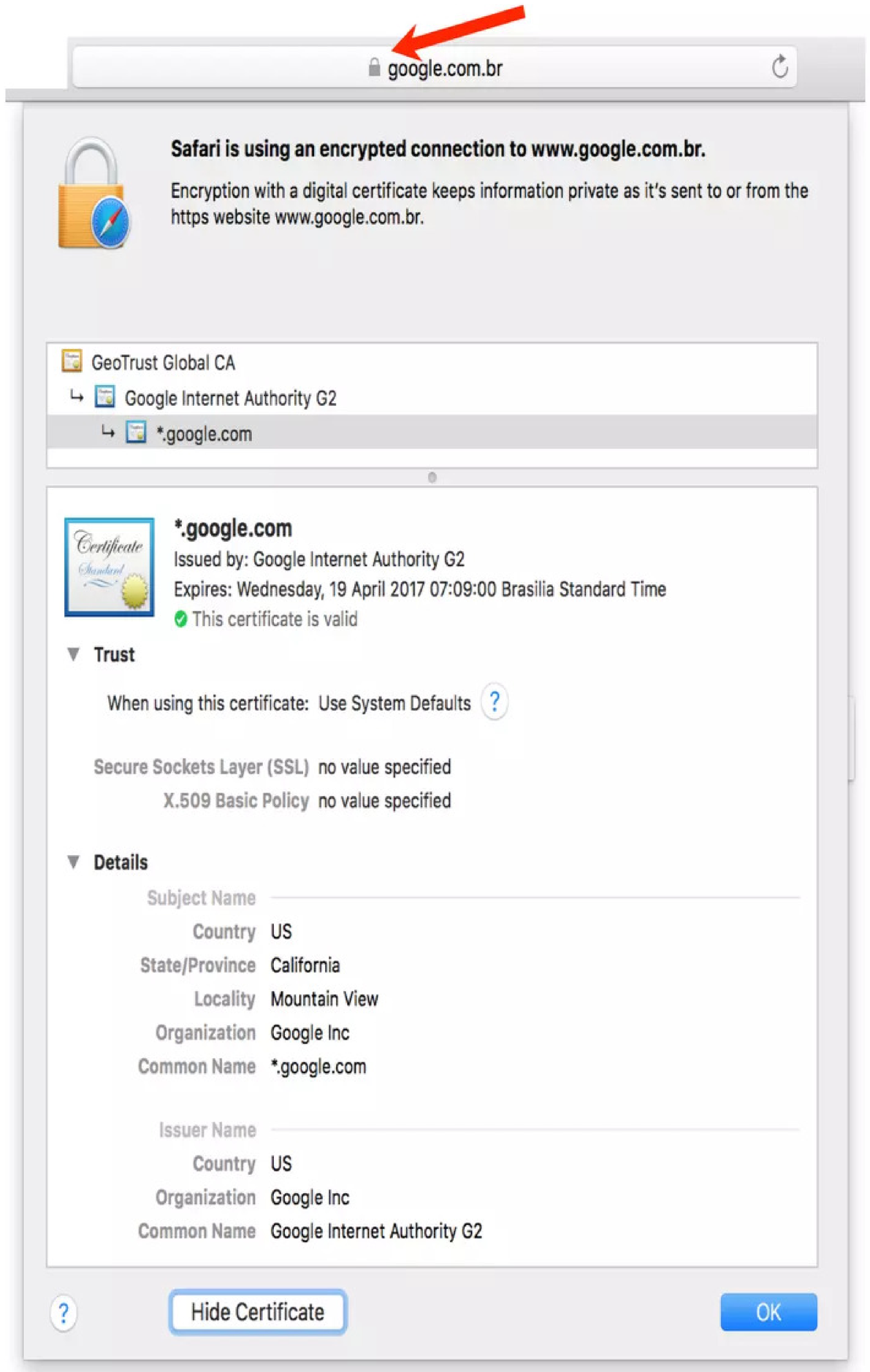
This is all the info you get

Public Key Infrastructure (PKI)



MS IE 11:
View certificates

Public Key Infrastructure (PKI)



Apple Safari:
View certificates

Public Key Infrastructure (PKI)

- Digital certificate revocation
  - Revocation
  - Suspension (temporary)

- Reasons for revocation
  - Suspicion or compromise of private key
  - Suspicion or compromise of CA
  - End of operation
  - End of affiliation

- Revocation and expiration are distinct events!

Public Key Infrastructure (PKI)

- Cripto tokens









Public Key Infrastructure (PKI)

- Hardware Security Modules (HSM)





Public Key Infrastructure (PKI)

- Microsoft Crypto API
  - Present since Windows NT 4.0
  - Also used to access tokens and HSMs
  - Present in .NET Framework and JDK (SunMSCAPI Provider, as a conduit to MS-CAPI)

- Oracle SunJSSE Provider
  - Present in JDK
  - Capacity for a large number of ciphersuites)

Agenda

- What is cryptography?
- Hash
- Symmetric cryptography
- Asymmetric cryptography
- Public Key Infrastructure (PKI)
- Attacks
- Steganography
- Best practices: domestic use
- References

Attacks

- Brute force, dictionaries, default passwords
- Rainbow tables
- Pass-the-hash
- Private key recovery
- Comparison of files
- Tunnel interception
- Flawed encryption method
- Symmetric key vs. Asymmetric key

Attacks

- Default passwords list
  - http://www.cirt.net/passwords
- Dictionaries
  - http://www.openwall.com/passwords/wordlists/
  - http://lastbit.com/dict.asp
- Dictionaries generator
  - L517
    - http://code.google.com/p/l517/

Attacks

- Brute force
  - The attacker tries all possible values from a range of possibilities
- Rainbow tables
  - Usually a group of attackers generates the files containing passwords and their hashes
  - When a password hash is generated it is not necessary to go through the whole brute force task
  - All it takes is a search in the hashes database

Attacks

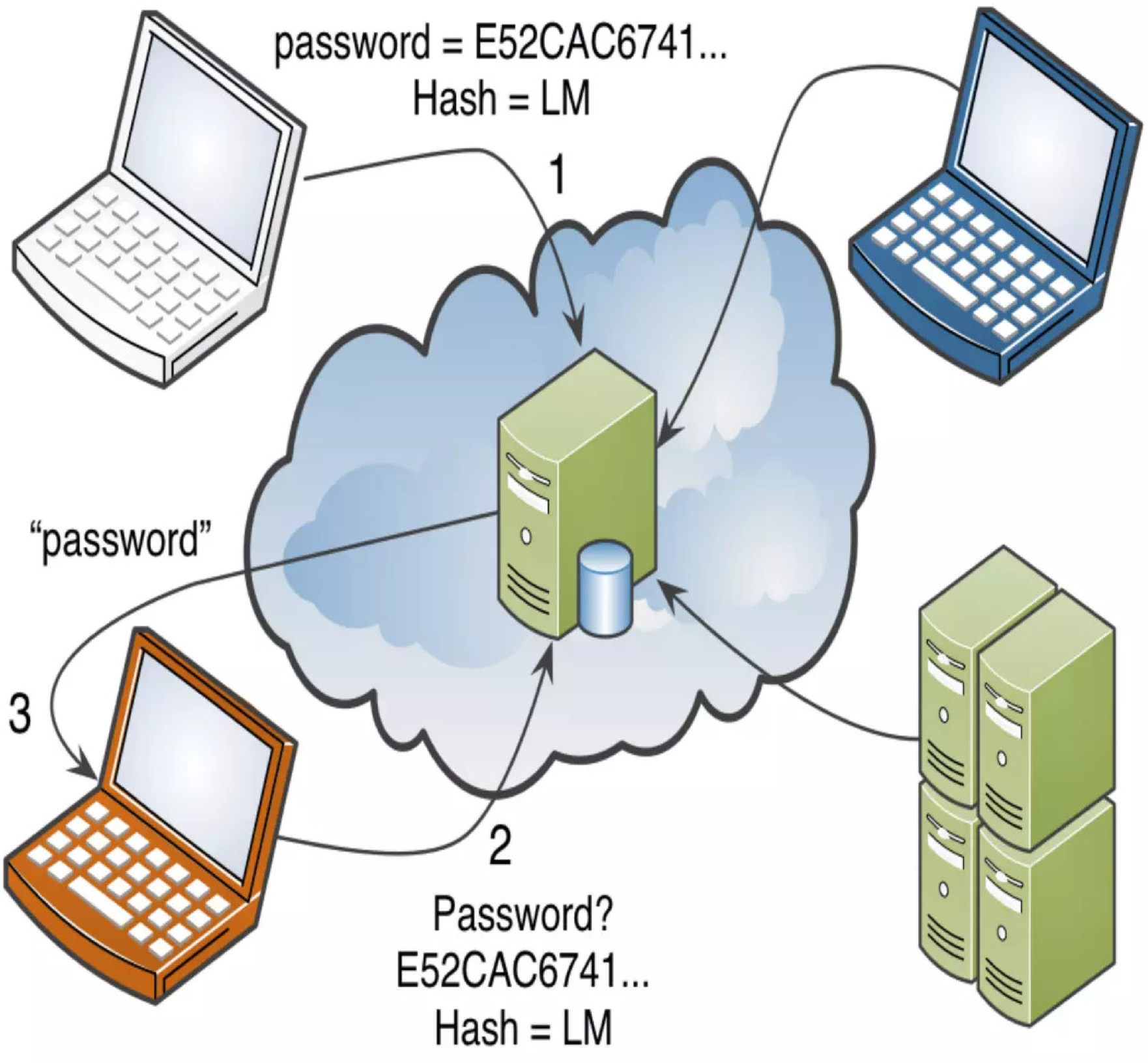
- Brute force

Key (Bits)	Permutations	Brute force time for a device checking 2^{56} permutations/second
8	2^{8}	0 milisseconds
40	2^{40}	0.015 milisseconds
56	2^{56}	1 second
64	2^{64}	4 minutes and 16 seconds
128	2^{128}	149.745.258.842.898 years
256	2^{256}	50.955.671.114.250.100.000.000.000.000.000. 000.000.000.000.000.000.000 years

Source: NIST SP 800-57 Part 1 (2007)

Attacks

- Rainbow tables (freerainbowtables.com / distributed.net)



Attacks

- Hash Capture
  - Offline NT Password and Registry Editor
    - http://pogostick.net/~pnh/ntpasswd/
    - Supports all Windows versions from NT 3.5 to Win 8.1, 32 or 64 bit, also Server versions (like 2003, 2008, 2012)
    - With Windows offline, accesses the files that contain account hashes
    - It can change passwords, unlock and enable accounts! (if the user is using EFS, will lose access to the files)
    - Doesn't need to know the current password



  - Ophcrack, pwdump7 and other capture softwares
  - Sniffers, like Wireshark

Attacks

- Hash Capture
  - pwdump7

```
C:\Software\AUDITORIA\Hacking\pwdump>pwdump7.exe
Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

Administrador:500:NO PASSWORD**********************:CF589E918774DBA4FC46770C18378486:::
nobody:501:NO PASSWORD**********************:31D6CFE0D16AE931B73C59D7E0C089C0:::
SUPPORT_388945a0:1001:NO PASSWORD**********************:ADD7CC58257D86C0ABDAAC3C34E54CEE:::
IUSR_REDBULL:1003:3E956118D15B9BA081A1EB5765371732:12BF0538D9DCE47F5BD2852F9F9C1396:::
IWAM_REDBULL:1004:2C8EA4143D341222618E2B1942AEE5C8:BEE0B2CEF8264F7F6A0140899D7BD085:::
ASPNET:1006:NO PASSWORD**********************:207853CECE9459A9A37DC2958135C182:::
__vmware_user__:1021:NO PASSWORD**********************:5137896A2C18A9A967A0E6A2EC92B3AC:::
```

Attacks

- Rainbow tables (example)

LM	#0
Characters	[ABCDEFGHIJKLMNOPQRSTUVWXYZ]
Combinations	8.353.082.582
Table size	610 MB
Probability of success	0.9904 (99.04%)

LM	#1
Characters	[ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]
Combinations	80.603.140.212
Table size	3 GB
Probability of success	0.991 (99.1%)

Attacks

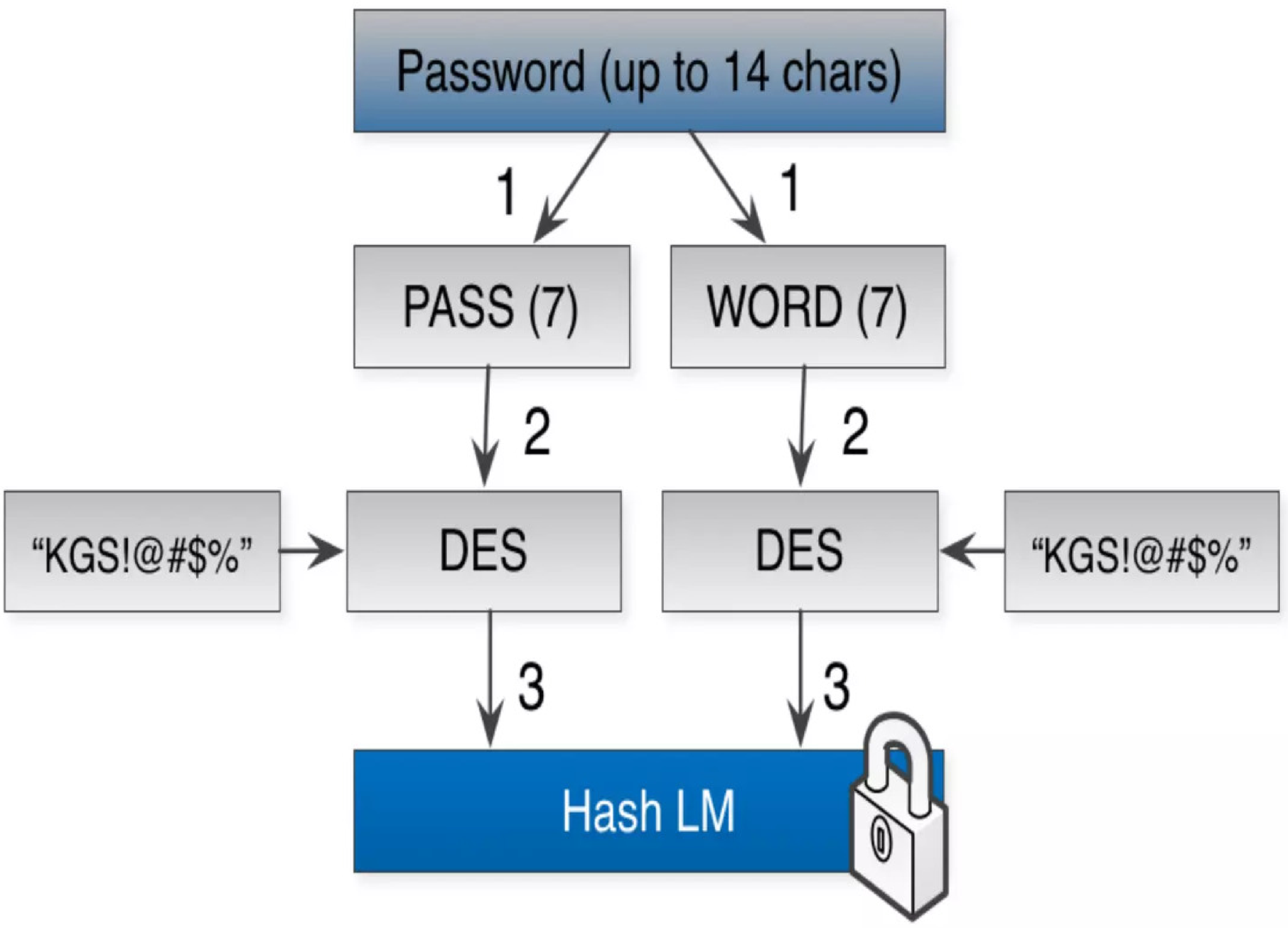
- Rainbow tables (example)

LM	#5
Characters	[ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$ %^&*()-_+=]
Combinations	915.358.891.407 (2 ^ 39.7)
Table size	24 GB
Probability of success	0.9990 (99,90%)

LM	#6
Characters	[ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$ %^&*()-_+=~`[]{}|\:;"'<>,.?/]
Combinations	7.555.858.447.479 (2 ^ 42.8)
Table size	64 GB
Probability of success	0.9999 (99,99%)

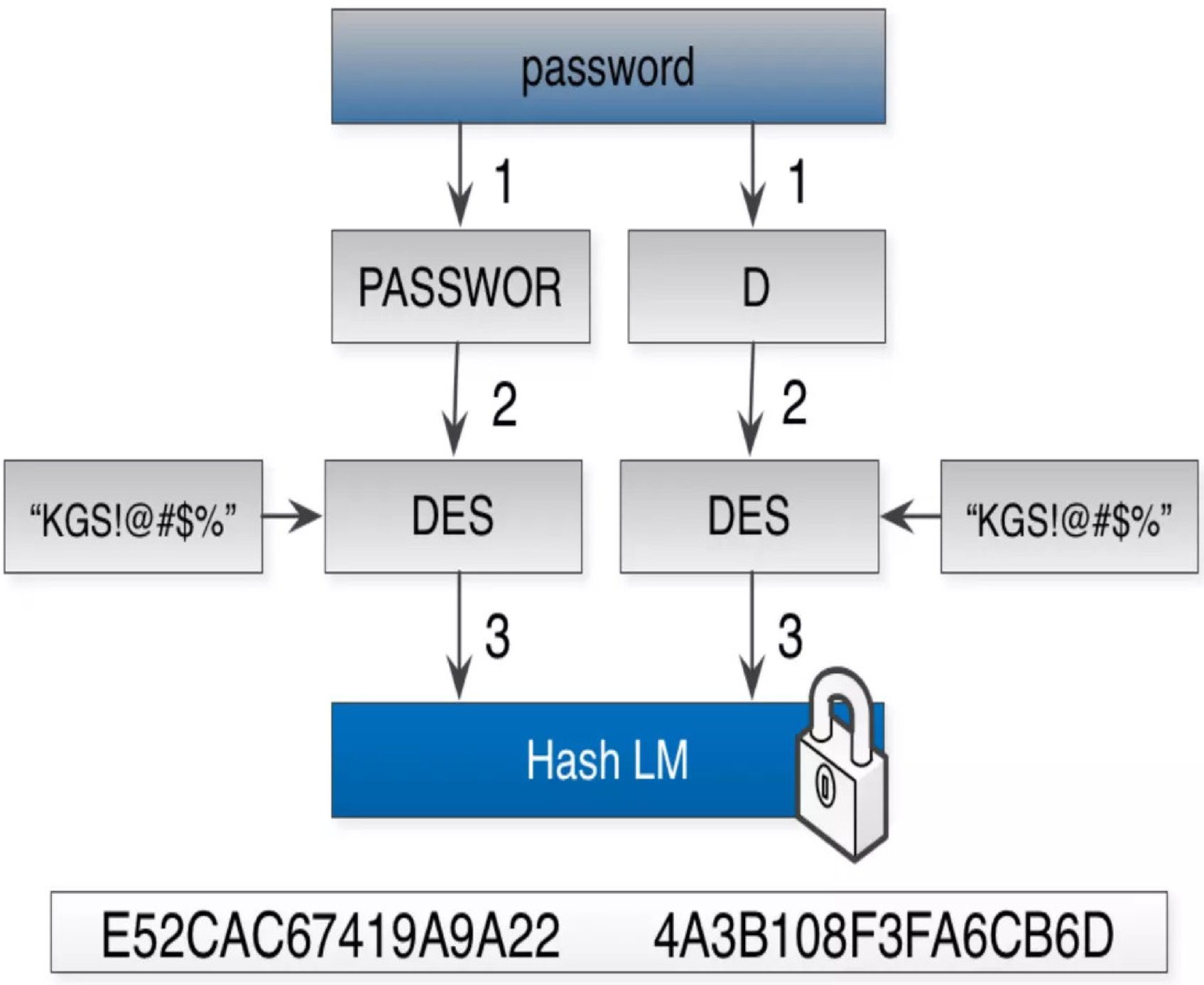
Attacks

- Rainbow tables
  - LM Hash cryptography problem



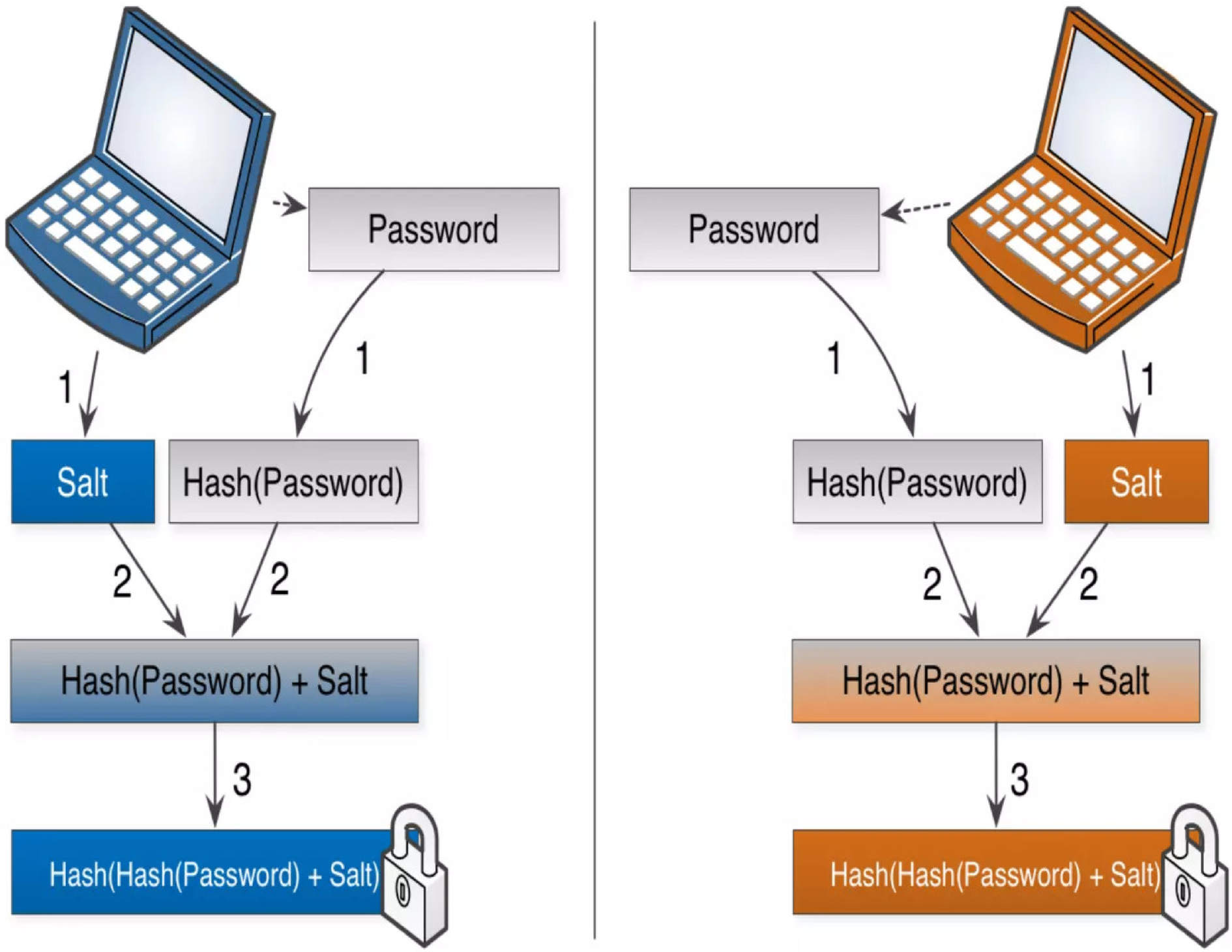
Attacks

- Rainbow tables
  - LM Hash cryptography problem (example)



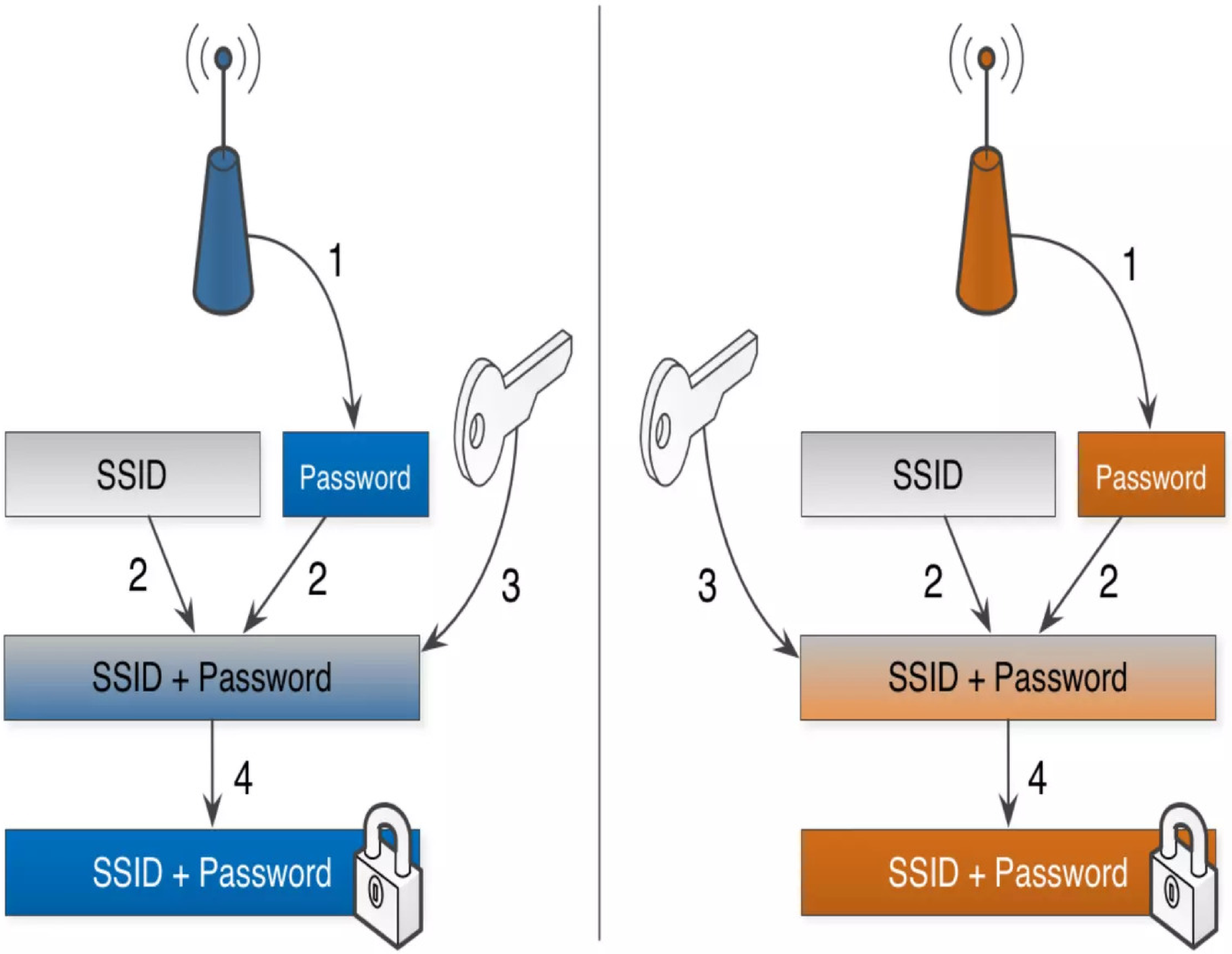
Attacks

- Defense against Rainbow tables
  - Salt: random bit sequence



Attacks

- Defense against Rainbow tables
  - WPA2



Attacks

- Defense against Brute force and Rainbow tables
  - Passwords should have a minimum of 12 characters
  - A password containing only 14 lowercase letters is as strong as a 10 char password containing lowercase, uppercase, numbers ans symbols
  - Include numbers and symbols if it is allowed by the system
  - If the system differeciates uppercase and lowercase, use both
  - Do not use the same password everywhere

  - Example: `4pRte!ai@3` – mixes uppercase, lowercase, numbers and symbols

Attacks

- Weak passwords
  - Default passwords: *password*, *default*, *admin*, *guest*, etc.
  - Dictionary: words in any language
  - Added number: *password1*, *deer2000*, *john1234*, etc.
  - With simple obfuscation: *p@ssw0rd*, *h4cker*, *h3ll0*, etc.
  - Doubled: *crabcrab, stopstop, treetree, passpass,* etc.
  - Common keyboard sequences: *qwerty*, *12345*, *asdfgh*, *fred*.
  - Known numeric sequences: 911, 3,14159... (π), 2,7182... (e).
  - Identifiers: *jsmith123*, *1/1/1970*, *555–1234*, “login”, etc.
  - Personal information: license plate, phone numbers, birth date, name of relatives or pets, nicknames, etc. that require a simple investigation

Attacks

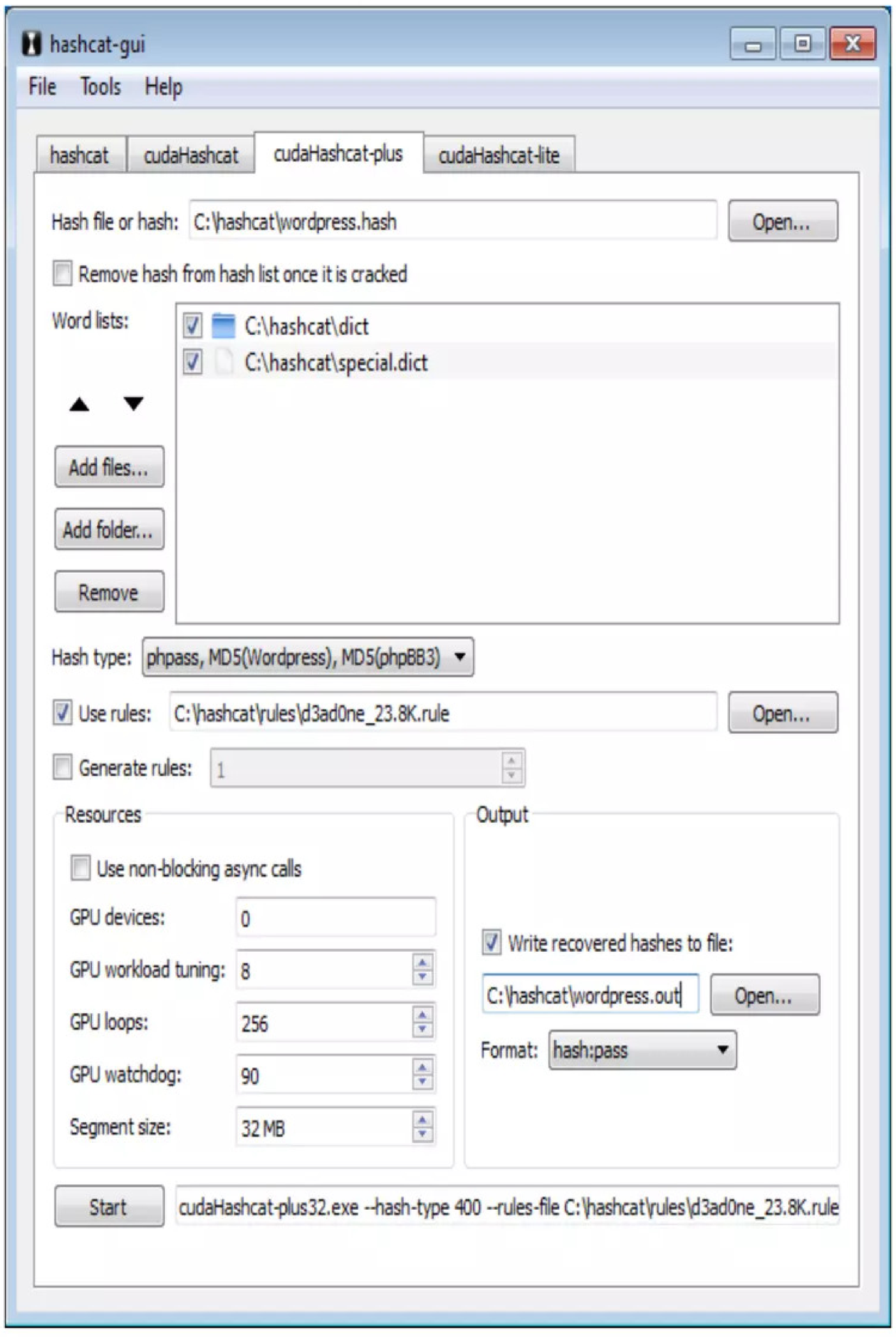
- Weak passwords
  - Dave Kleiman, American forensic investigator, detected through the analysis of 3 million passwords that
    - Letter "e" was used over 1.5 million times
    - Letter "f" was used only 250.000 times
    - In a normal distribution each letter should be used 900.000 times
  - Most used number is “1”
  - The most common letters are “a”, “e”, “o”, “r”

Attacks

- Brute force
  - Hashcat: http://hashcat.net/hashcat-gui/ (Windows / Linux)
  - Ophcrack (Windows, free and commercial)
  - LC6 (Windows, commercial)
  - John the Ripper (Unix, free)
  - Elcomsoft (Windows, commercial)
  - Hydra (Unix, free)
  - Hashkill (Linux, free)
  - PDF: FreewarePDFUnlocker (Windows, free)
  - RAR: cRARk (Windows, free)

- Rainbow tables
  - OnlineHashCrack: www.onlinehashcrack.com
  - Tobtu: www.tobtu.com/md5.php

Attacks



Hashcat

Attacks

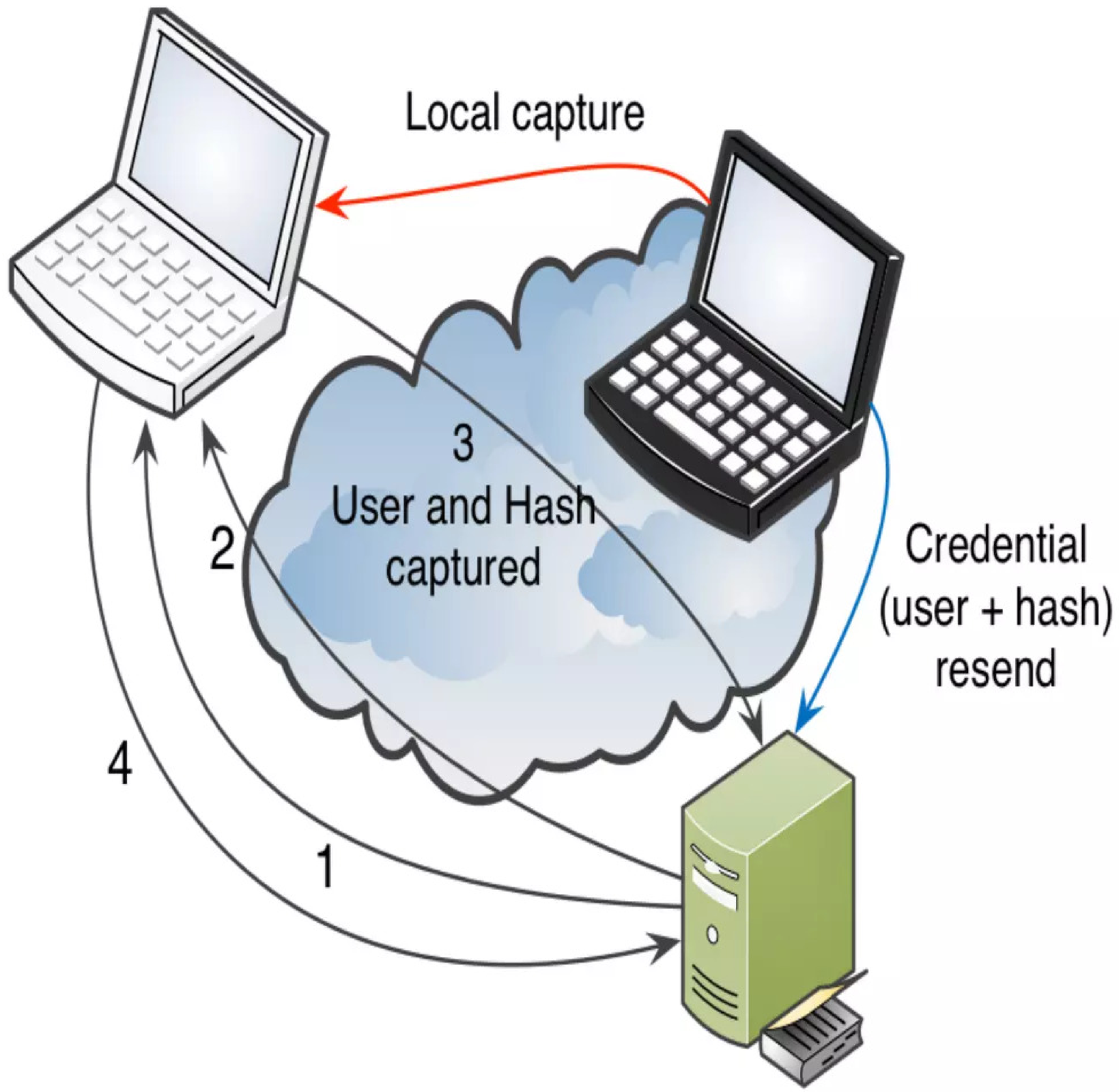
- Distribution among workstations
  - Boinc
  - Software that distributes the load

- Increase in capacity
  - SSD drives
  - Parallel GPUs
  - Processor upgrade

- Development and GPUs
  - http://developer.nvidia.com/cuda-tools-ecosystem
  - http://developer.amd.com/pages/default.aspx

Attacks

- Pass-the-hash (similar to Replay attack)



Attacks

- Defense against Replay Attacks
  - 1. One Time Password (OTP)
    - Generated from an open algorithm
    - Even knowing the previous number, it is not possible to predict the next
    - Available through hardware and software solutions

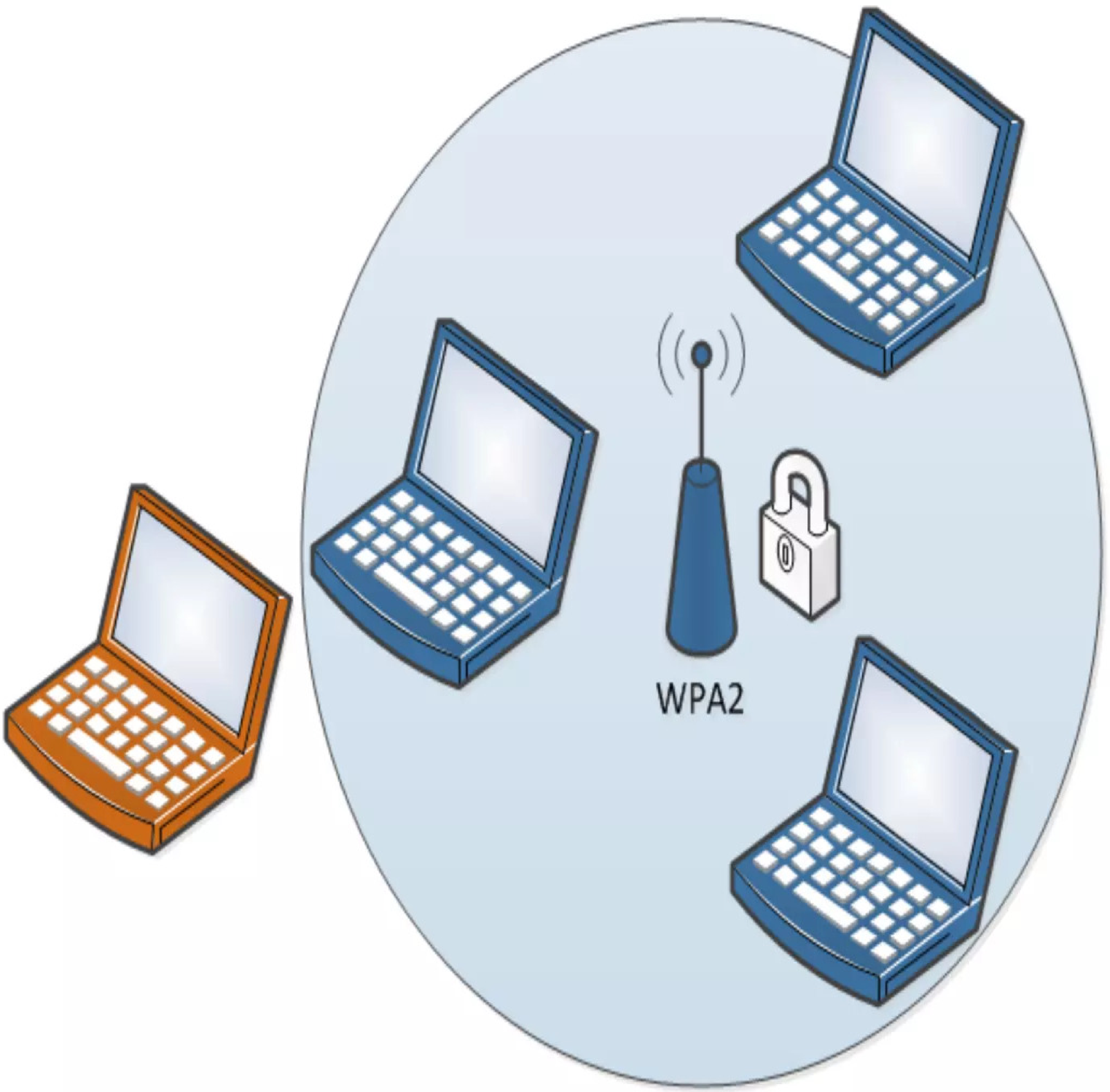




  - 2. Nonce
  - 3. Timestamps

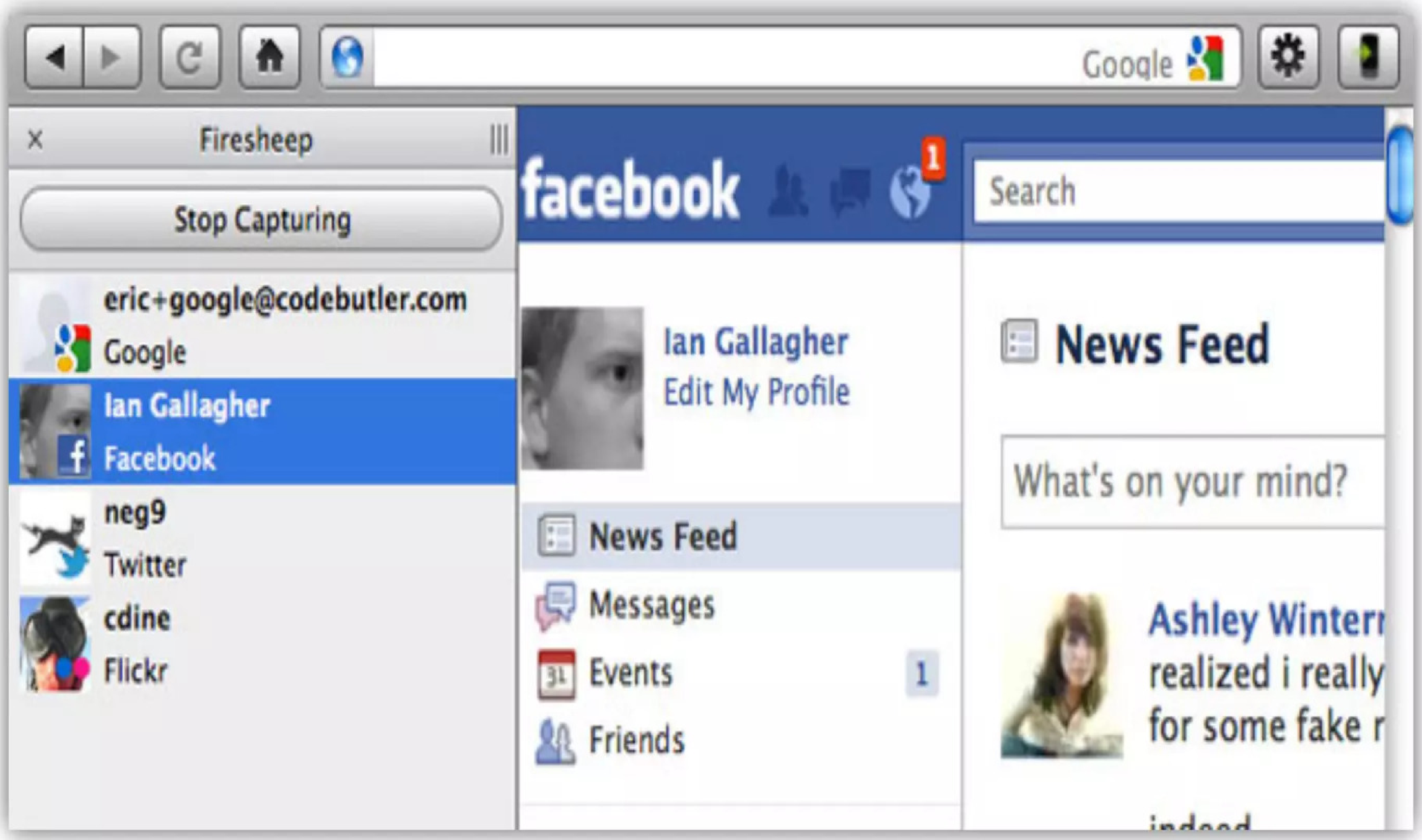
Attacks

- Wireless network with WPA2 encryption



Attacks

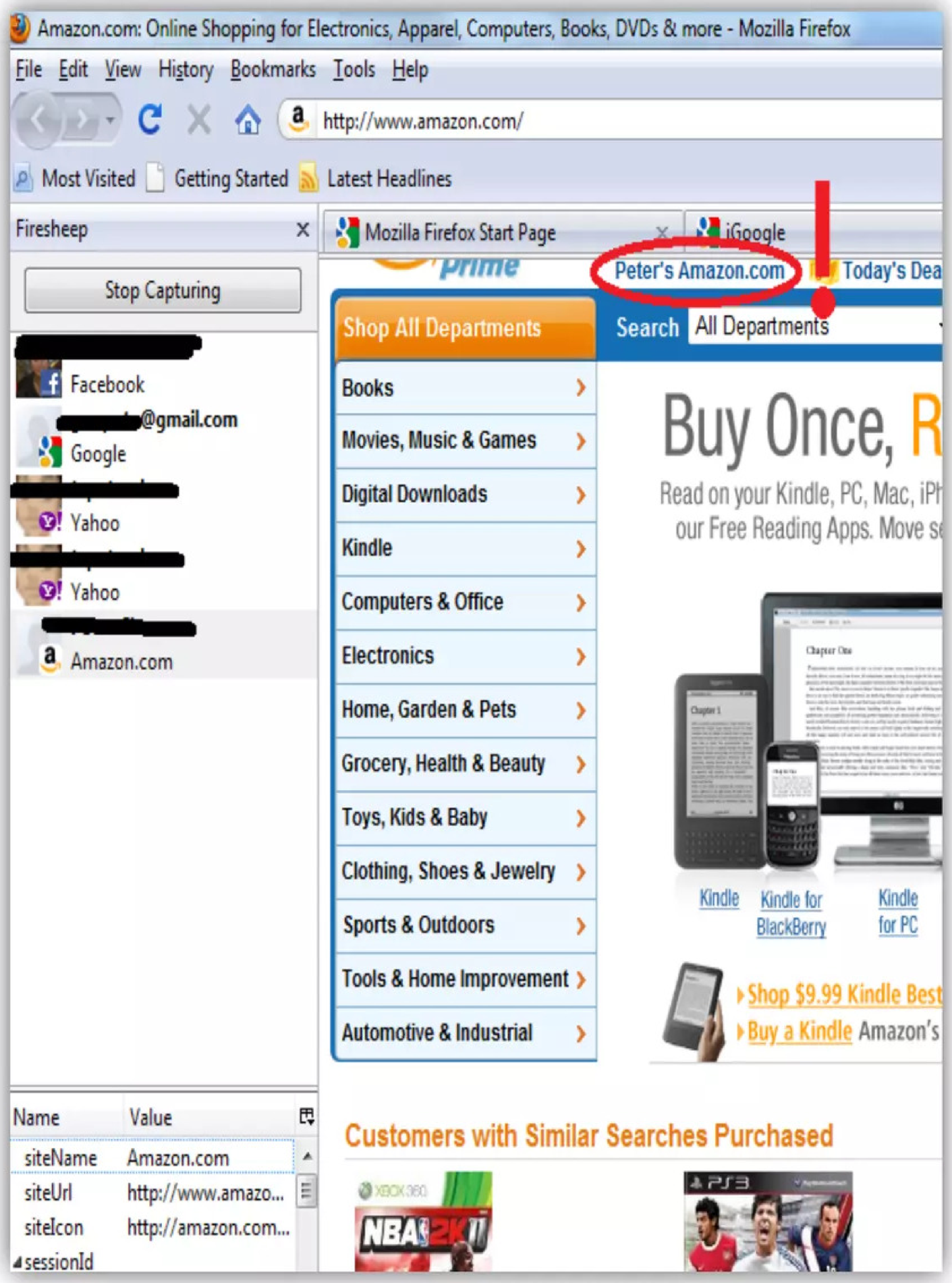
- Session hijacking (similar to Replay attack)



- Only possible when there is no SSL/TLS (HTTPS)!

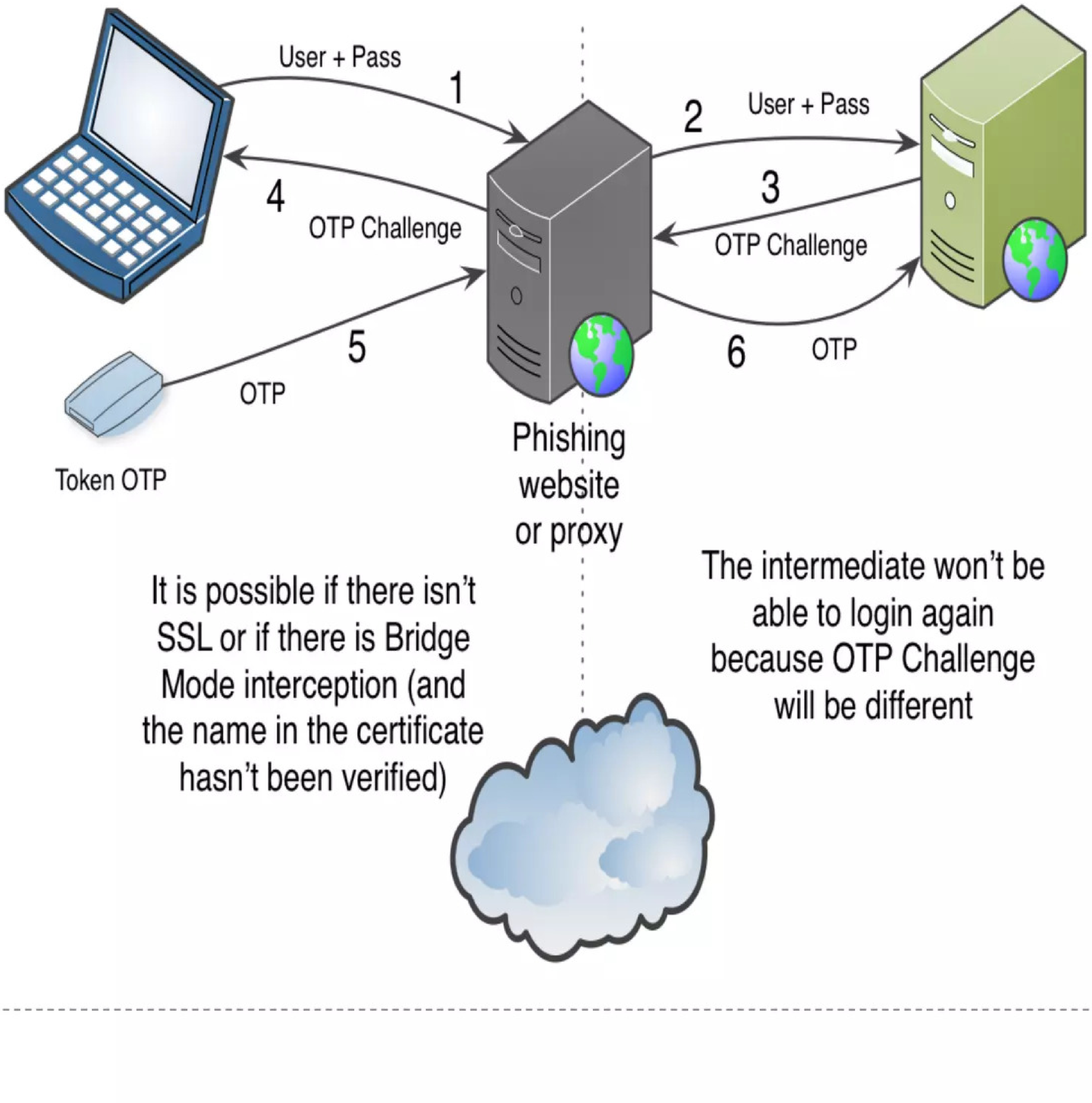
Attacks

- Session hijacking (with Firesheep)



Attacks

- OTP Realtime Man-in-the-middle



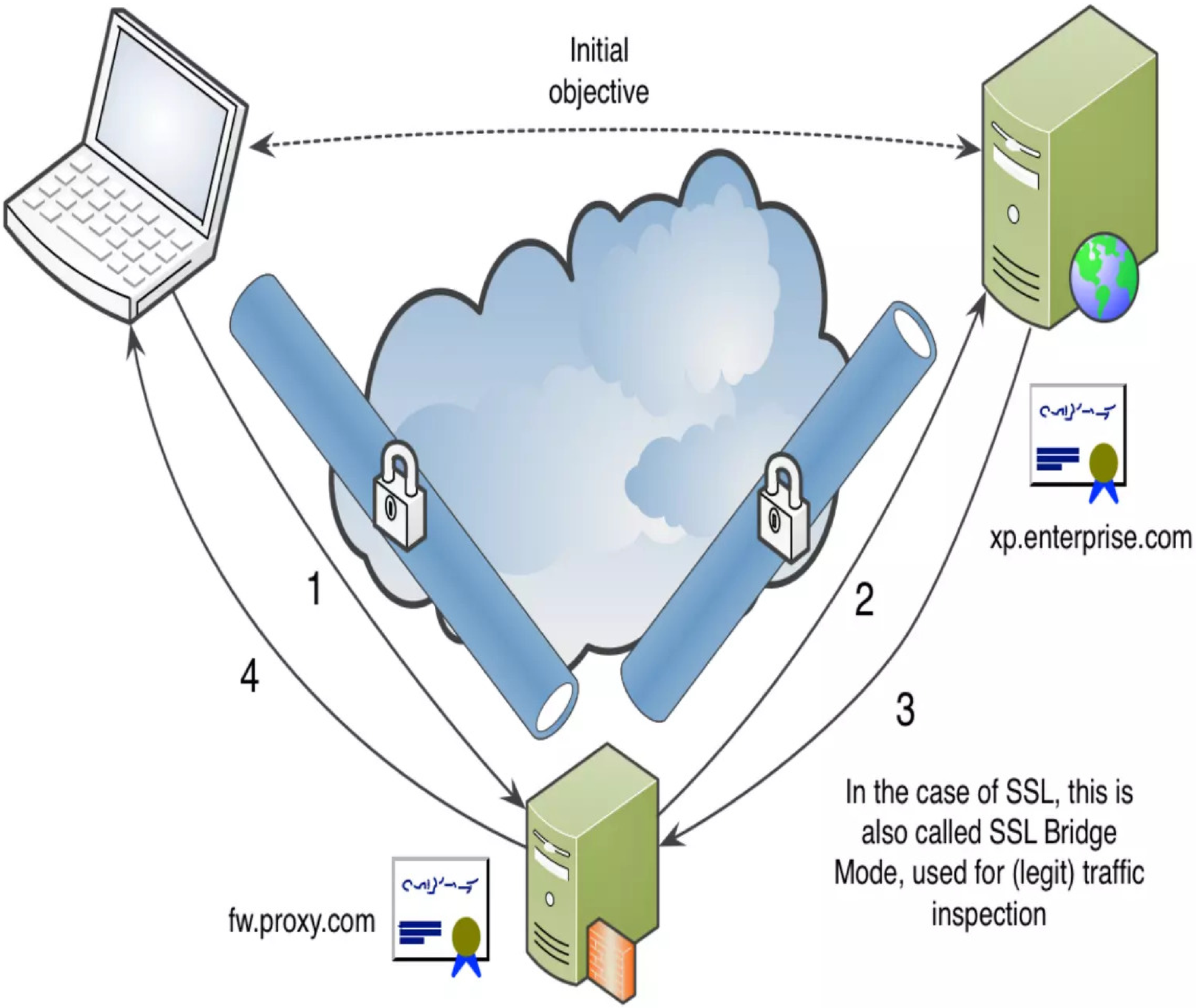
Attacks

- Private key recovery
  - Recovery of the private key generated in the workstation before being exported to the token

- File comparison
  - When the attacker has the encrypted version of the message (file) and the clear version it is possible to analyze them and obtain the encryption key in some cases

Attacks

- Tunnel interception (Man-in-the-middle)



Attacks

- Symmetric key vs. Asymmetric key
  - In 1977 56 bit DES was enough
  - The American government requires a 192- or 256-bits AES key (symmetric) for highly sensitive content
  - NIST best practices manual for key management suggests that 256 bits symmetric keys are equivalent to 15360 bits RSA keys (asymmetric)!

Attacks

NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

TDEA (Triple Data Encryption Algorithm) and AES are specified in [10].
Hash (A): Digital signatures and hash-only applications.
Hash (B): HMAC, Key Derivation Functions and Random Number Generation.
The security strength for key derivation assumes that the shared secret contains sufficient entropy to support the desired security strength. Same remark applies to the security strength for random number generation.

Date	Minimum of Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash (A)	Hash (B)
(Legacy)	80	2TDEA*	1024	160	1024	160	SHA-1**	
2016 - 2030	112	3TDEA	2048	224	2048	224	SHA-224 SHA-512/224 SHA3-224	
2016 - 2030 & beyond	128	AES-128	3072	256	3072	256	SHA-256 SHA-512/256 SHA3-256	SHA-1
2016 - 2030 & beyond	192	AES-192	7680	384	7680	384	SHA-384 SHA3-384	SHA-224 SHA-512/224
2016 - 2030 & beyond	256	AES-256	15360	512	15360	512	SHA-512 SHA3-512	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-512

Source: keylength.com

Attacks

Bundesamt
für Sicherheit in der
Informationstechnik

Date	Symmetric	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash	
2016	128	2048	256	2048	256	SHA-256 SHA-512/256 SHA-384 SHA-512	SHA3-256 SHA3-384 SHA3-512
2017 - 2021	128	3072 (*)	256	3072 (*)	256	SHA-256 SHA-512/256 SHA-384 SHA-512	SHA3-256 SHA3-384 SHA3-512

Source: keylength.com

Agenda

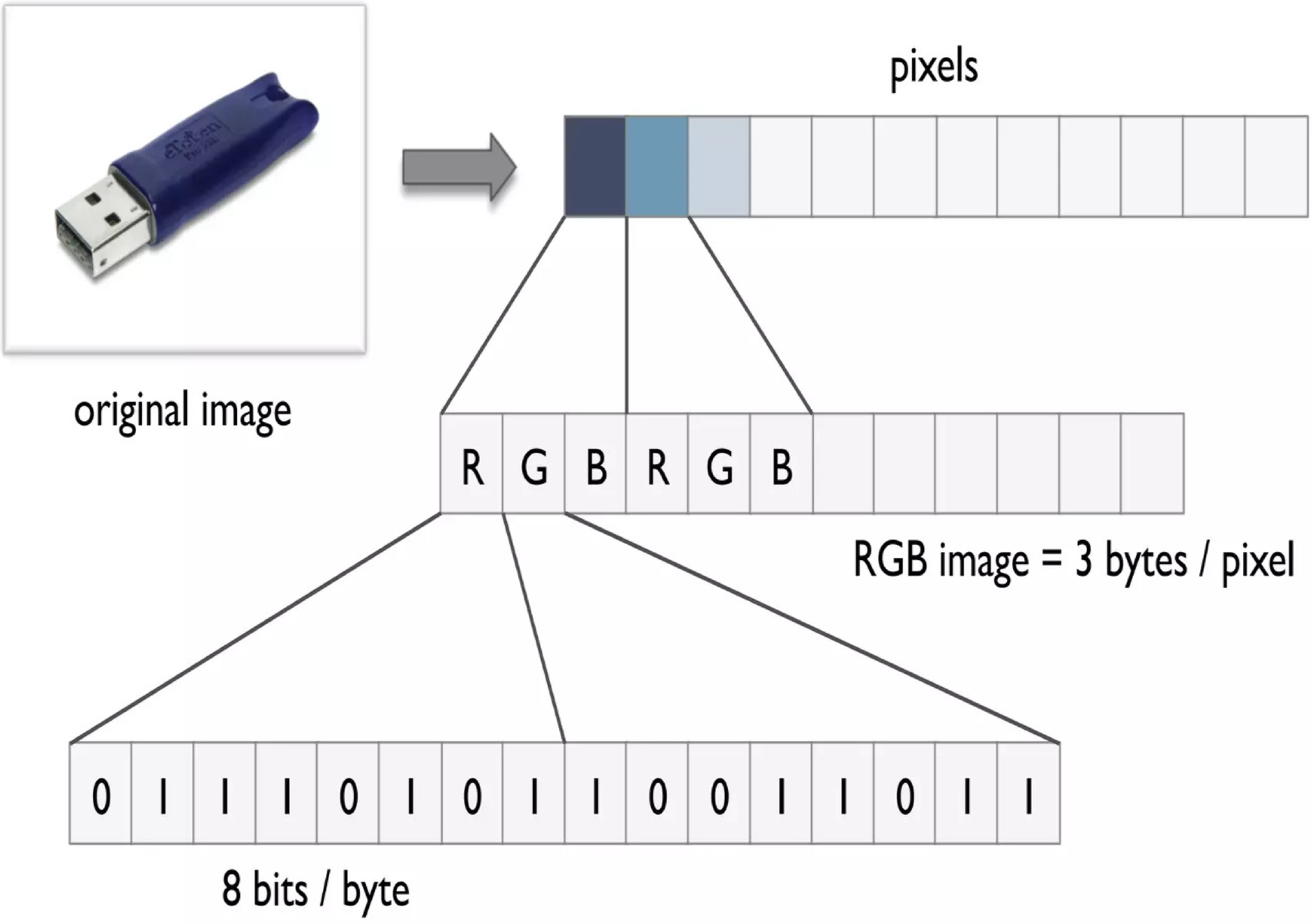
- What is cryptography?
- Hash
- Symmetric cryptography
- Asymmetric cryptography
- Public Key Infrastructure (PKI)
- Attacks
- Steganography
- Best practices: domestic use
- References

Steganography

- History
  - Combines the Greek word *steganos* (στεγανός), “hidden or protected”, with *graphei* (γράφη), “writing”
  - Security through obscurity
  - Messages written on envelopes in the area covered by postage stamps
  - During and after World War II, espionage agents used photographically produced microdots to send information back and forth
  - Today it is used to embed watermarks in images, videos and audio to protect intellectual property

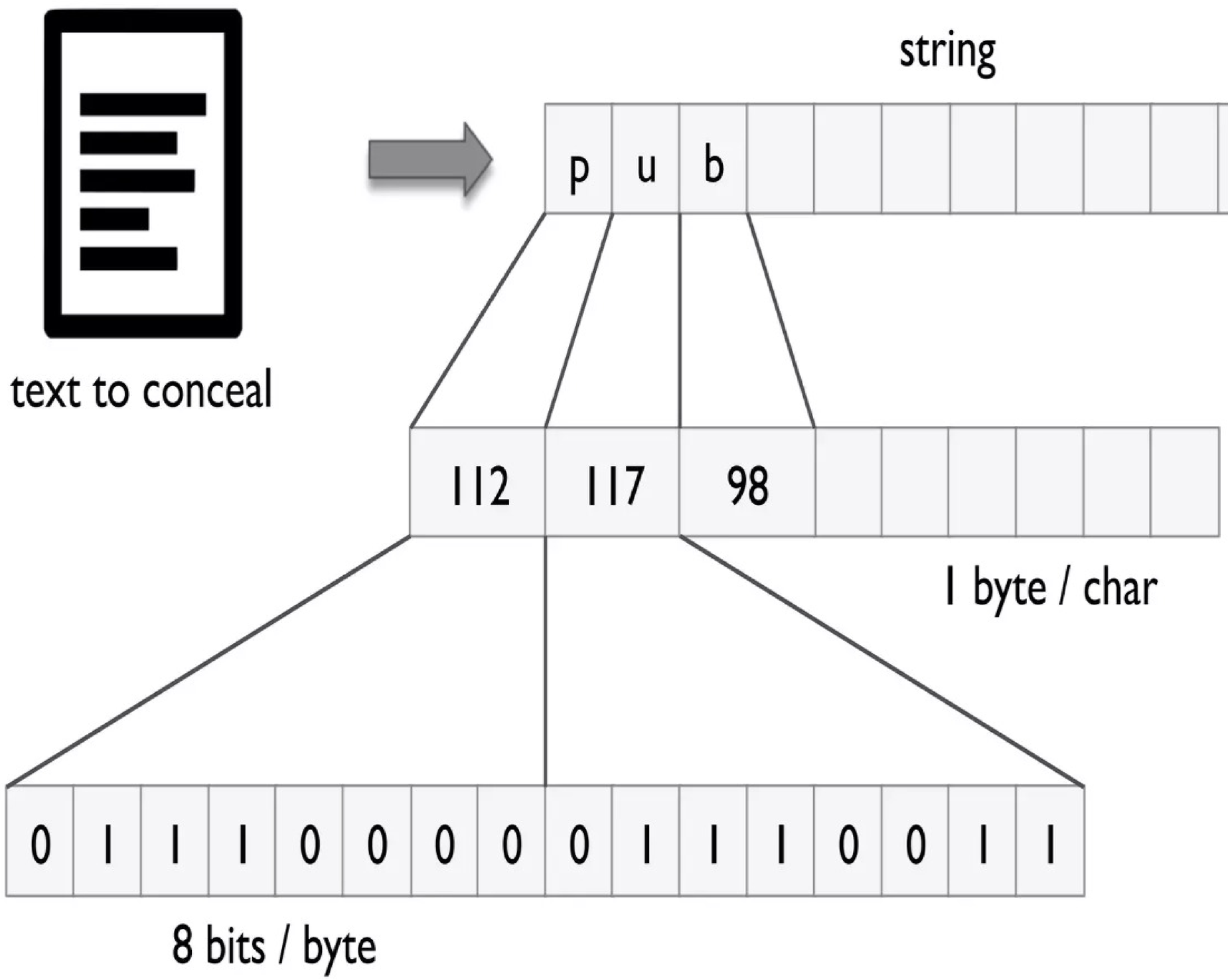
Steganography

- Most used methods
  - LSB (Least significant bits)



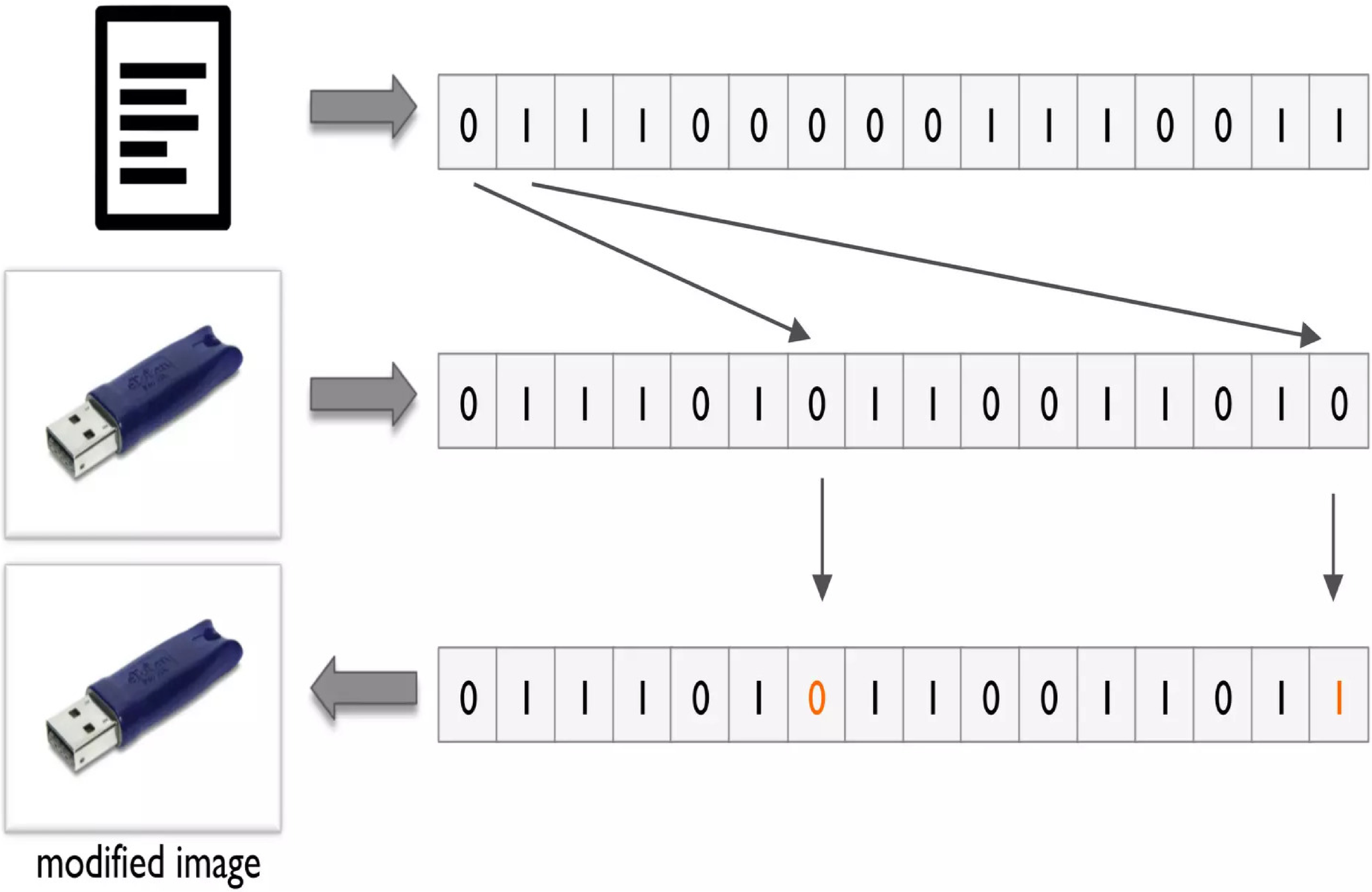
Steganography

- Most used methods
  - LSB (Least significant bits)



Steganography

- Most used methods
  - LSB (Least significant bits)

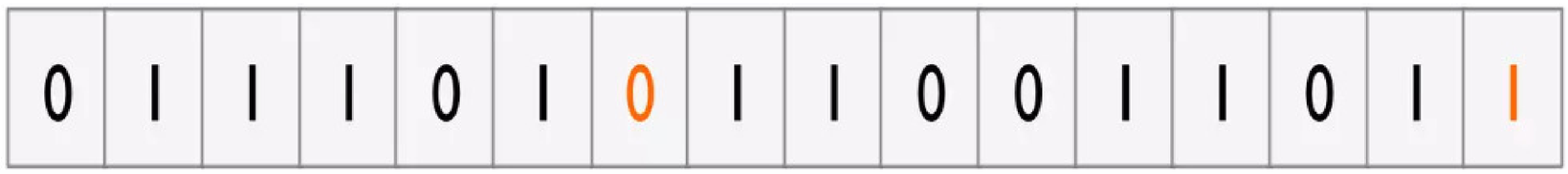


Steganography

- Most used methods
  - LSB (Least significant bits)



modified image





original text

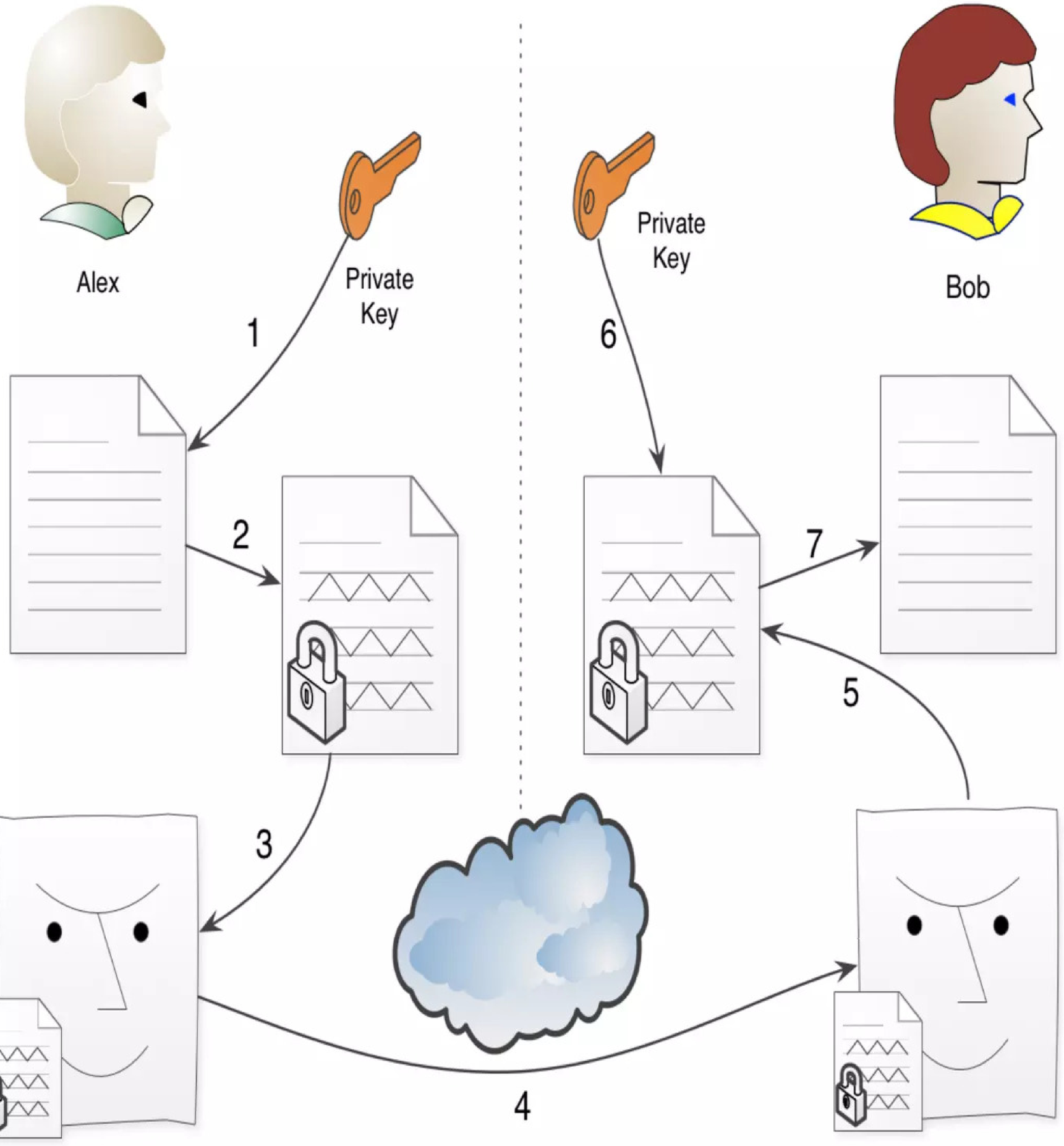


Recovery

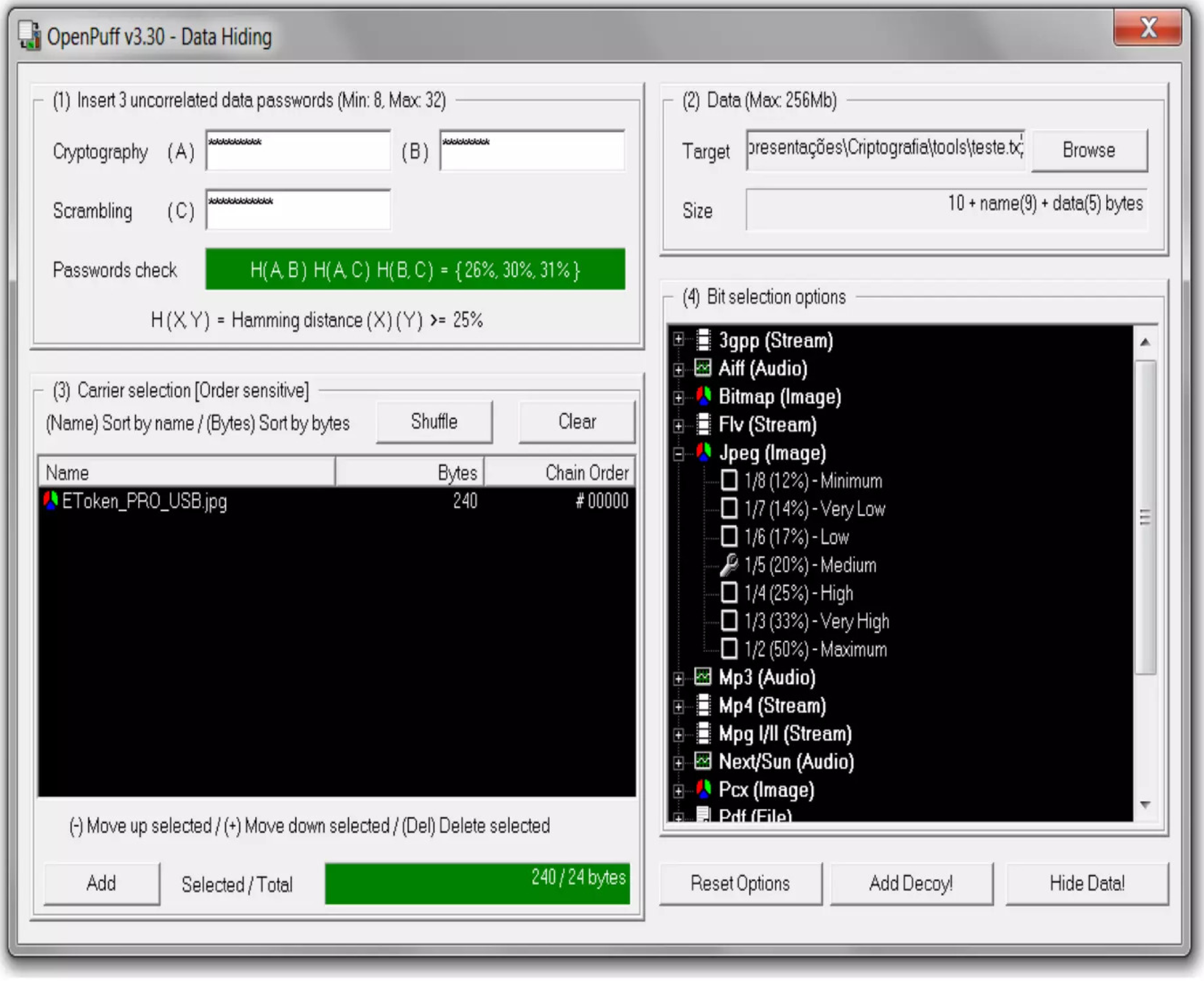
Steganography

- Most used methods
  - High frequency audio
    - CD Quality audio has frequency of 44.100 Hz
    - Effective frequency is 22.050 Hz
    - Humans can distinguish sounds from 15 Hz to 15.000 or 20.000 Hz (depending on the individual). Those who can listen up to 20.000 Hz cannot distinguish the higher frequencies very well
    - It is possible to change the information in the higher frequencies of sound and the result will be imperceptible

Steganography



Steganography



Steganography

- Original JPEG
- SHA-1:
  2ebd0b60f51e38f0f0
  1224e017e650e7b80f
  cd1f

- Modified JPEG
- SHA-1:
  254f2ed072beab5c3a
  52c12281c48df5d0e4
  8ddc





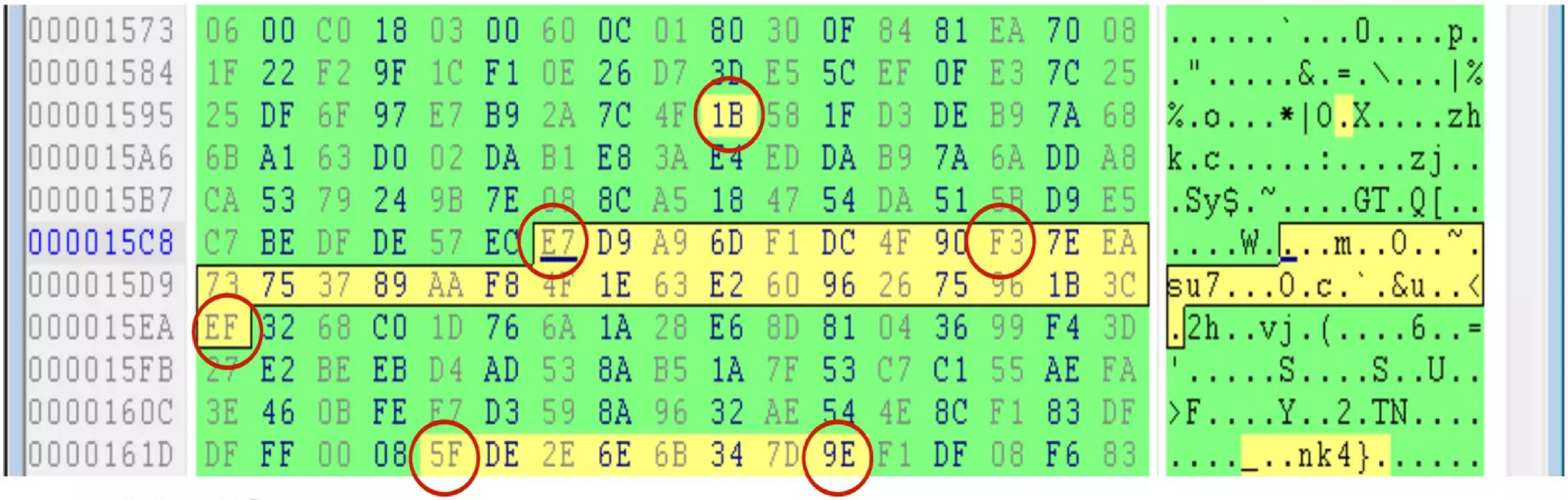
Steganography

- Steganography applications
  - Terrorism
  - Espionage
  - Fingerprinting
  - Watermark

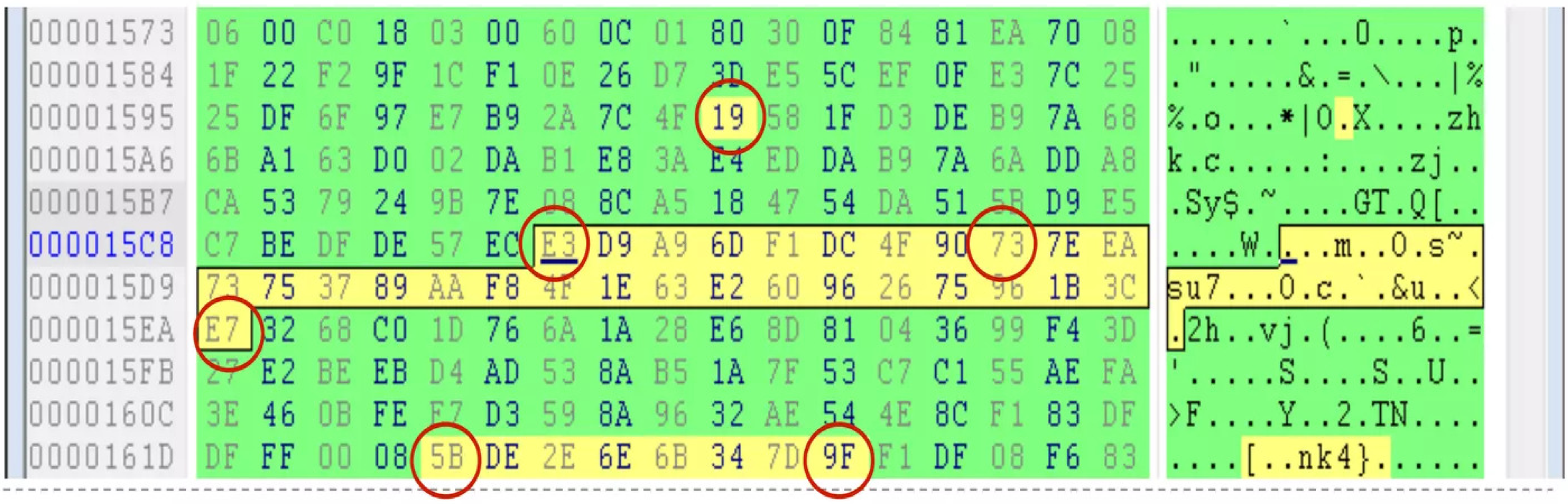
- Steganalysis
  - Uses the original file for comparison
  - Statistical analysis of files from the same device
  - Noise analysis searching for changes in the Least Significant Bits

Steganography

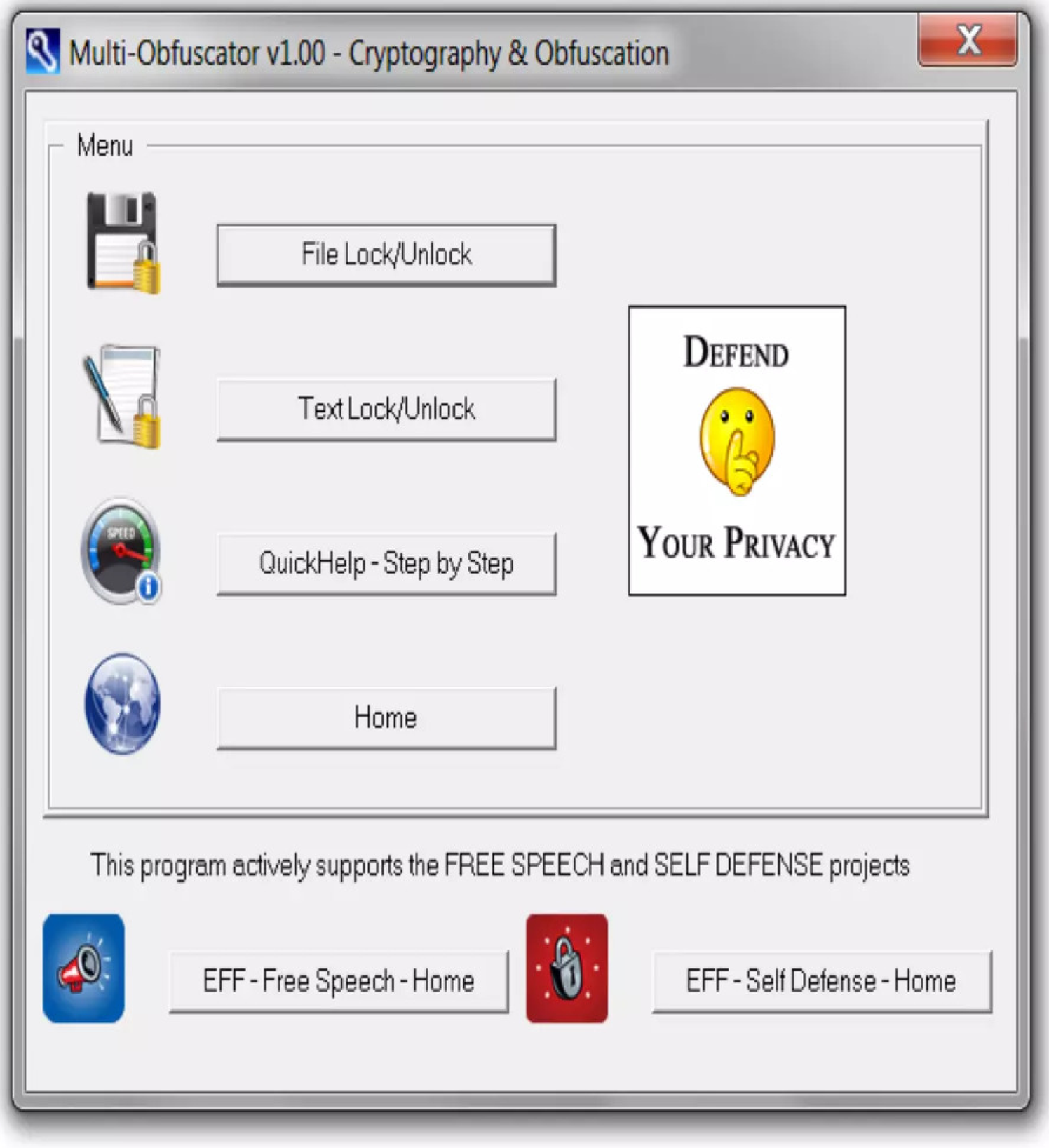
- Steganalysis
  - Original



  - Modified



Cryptography



"MultiObfuscator is a professional cryptography tool"

Agenda

- What is cryptography?
- Hash
- Symmetric cryptography
- Asymmetric cryptography
- Public Key Infrastructure (PKI)
- Attacks
- Steganography
- Best practices: domestic use
- References

Best practices: domestic use

- Digital certificate level 1 (A1): free
  - Let's Encrypt: letsencrypt.org
  - Instant SSL: www.instantssl.com
  - Aloaha: www.aloaha.com
  - CACert.org: www.cacert.org

- VeraCrypt
  - Files
  - Volumes
  - Hidden Volumes

References

- Coursera / Stanford University
  - Cryptography I
  - Prof. Dan Boneh
  - https://www.coursera.org/learn/crypto/

References

- ISO/IEC 18014 — Time-stamping services
- ISO/IEC 13888 — Non-repudiation
- ISO/IEC 10118 — Hash-functions
- ISO/IEC 27002 — Information technology - Security techniques - Code of practice for information security management
- ISO/IEC 19790 — Security Requirements
- for Cryptographic Modules

References

National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

Cryptographic algorythm	Special Publications ou FIPS
Triple Data Encryption Standard (TDES)	SP 800-67, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, and SP 800-38A, Recommendation for Block Cipher Modes of Operation – Methods and Techniques
Advanced Encryption Standard (AES)	FIPS 197, Advanced Encryption Standard, and SP 800-38A
Digital Signature Standard (DSS)	FIPS 186-3, Digital Signature Standard (DSS), dated June 2009
RSA algorithm	ANSI X9.31 and Public Key Cryptography Standards (PKCS) #1 v2.1: RSA Cryptography Standard-2002
Hashing algorithms SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512	FIPS 180-3, Secure Hash Standard (SHS), dated October 2008