

PKI, Encryption, Certificates and You.

CommandPrompt



David Ball

“Security is mostly a superstition. It does not exist in nature.”

- Helen Keller

A Brief History of Encryption

(circa 1900 BC - 1970 AD)

Symmetric Encryption

- Use the same key for encryption/decryption
- Simplest and oldest form of encryption



Egyptian Hieroglyphs - circa 1900 BC

(Substitution cipher)



Caesar Cipher (circa 1st Century AD)

- Shift Substitution Cipher
- Attributed to Julius Caesar
- Each character shifted up by a fixed interval

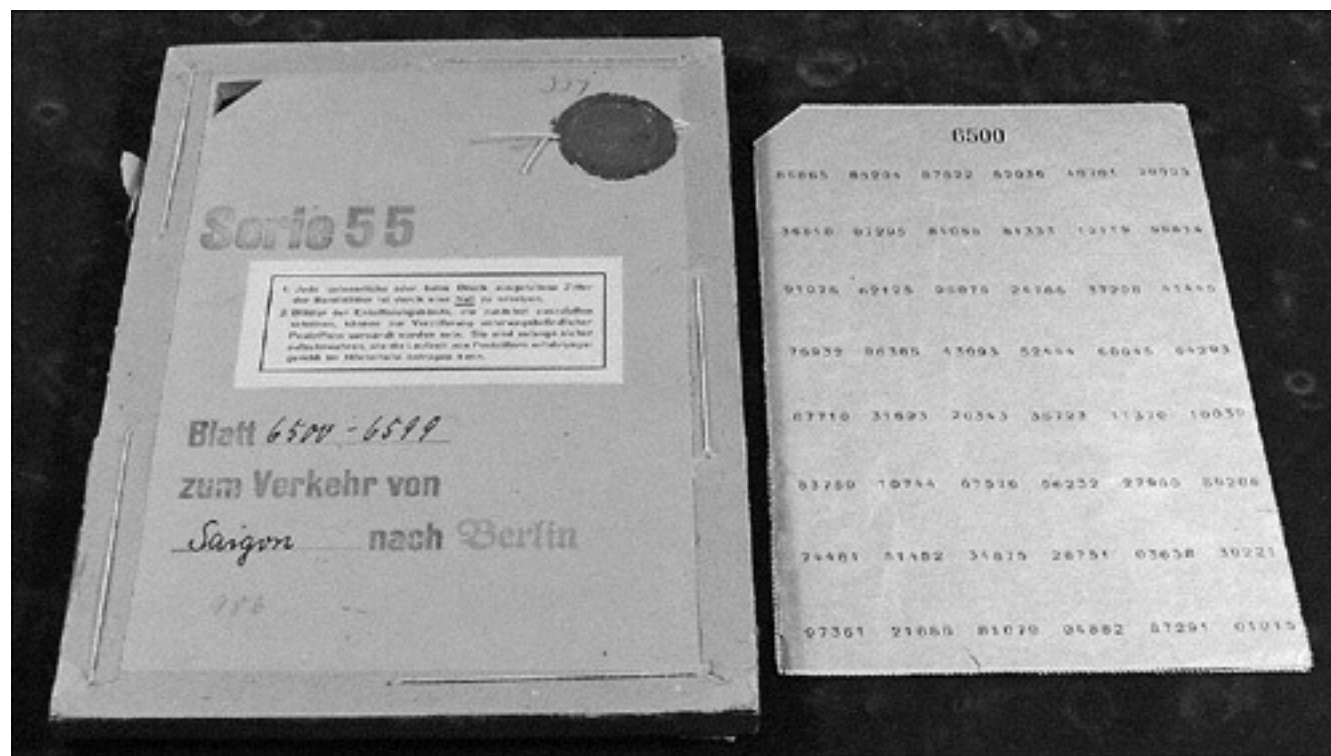
“I came, I saw, I conquered”

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---

(Shift each letter up five places)

E	F	G	H	I	J	K	L	M	N	O	P
---	---	---	---	---	---	---	---	---	---	---	---

“M geqi, M wea, M grqtyivih”



One Time Pad (1882)

- Substitution cipher
- Initially created for Telegraphic transmission
- Used extensively during WW II
- Given a long enough key (and correctly used) is practically unbreakable

Value:

8	5	12	12	15	23	15	18	12	4
---	---	----	----	----	----	----	----	----	---

+

Key Value:

17	10	3	15	14	22	22	26	25	18
----	----	---	----	----	----	----	----	----	----

=

Message + Key:

25	15	15	27	29	45	37	44	37	22
----	----	----	----	----	----	----	----	----	----

Cipher:

y	o	o	a	c	s	k	r	k	v
---	---	---	---	---	---	---	---	---	---

Message:

h	e	l	l	o	w	o	r	l	d
---	---	---	---	---	---	---	---	---	---

Cipher:

y	o	o	a	c	s	k	r	k	v
---	---	---	---	---	---	---	---	---	---

Kama Sutra

Straddling Checkerboard

Playfair

Null Cipher

Four Square

Pigpen

Rasterschlüssel 44

Rail Fence

...and many more.

In a nutshell:

- Apply a shared key to a message
- Shared key can be a string, character or integer
- Shared key can be an operation

1. Bob encrypts his message with a key
2. Bob sends that key to Alice
3. Bob sends the encrypted message to Alice
4. Alice decrypts the message using Bob's key

“Bob”



“Alice”



Key Distribution Problem

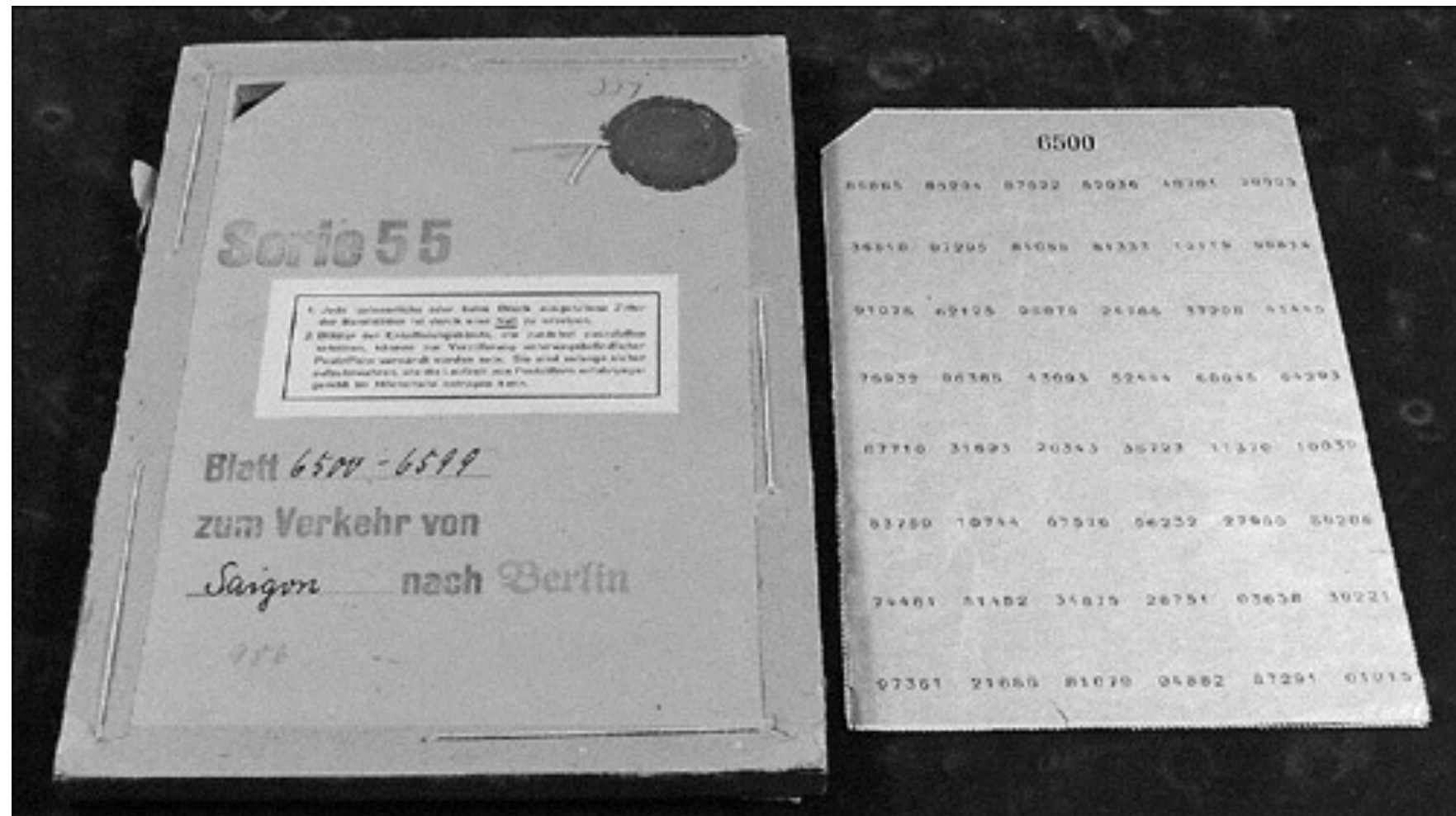
How do you distribute keys?

- Face to face
- Existing secure channel
- Trusted distribution
- Not very well

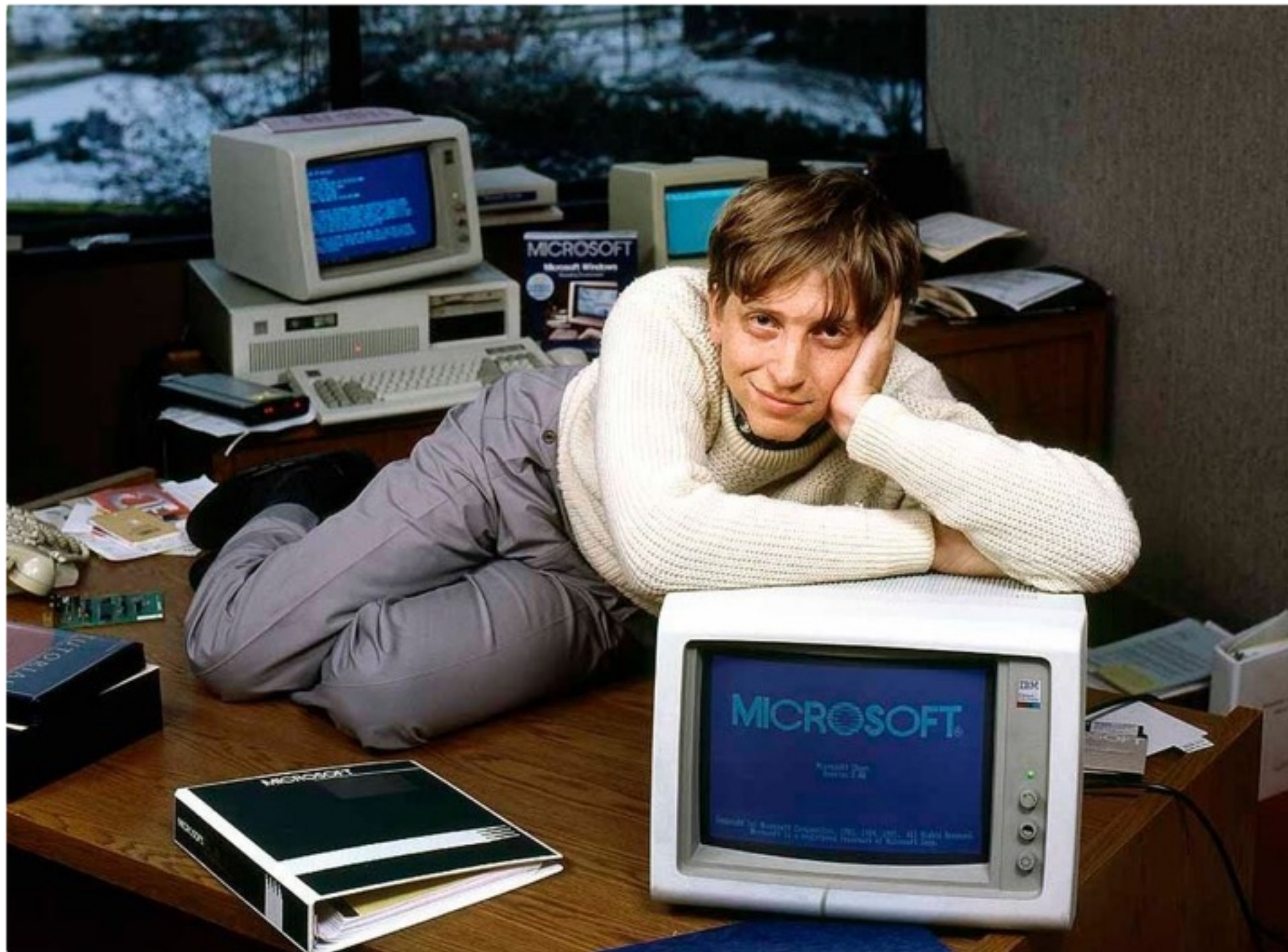
A Brief History of Encryption

(circa 1970 AD to present)

With increasing improvements in technology,
we went from this:



...to this:



Three examples:

Three examples:

- 1977 - Data Encryption Standard (DES)
 - 56-bit algorithm developed by IBM in the 1970's
 - Successfully compromised in 1998
- 1998 - Triple DES (3DES)
 - Successor to DES; ran data through DES three times
 - Still considered secure, but slow
- 1998 - Advanced Encryption Standard (AES)
 - Successor to DES - fast and secure
 - 128/192/256-bit versions
 - AES acceleration built in to Intel, AMD chips

AES

- Filevault 2 - 128-bit encryption
- iOS - 256-bit device encryption
- Bitlocker - 128/256-bit drive encryption
- IPSec - 128-bit (as standard)
- WPA - 256-bit
- SMB 3 - 128-bit

Public Key Infrastructure

History

1973 - Clifford Cocks at GCHQ comes up with a workable public key algorithm

1976 - Whitfield Diffie and Martin Hellman devise a method of key exchange

1977 - Ron Rivest, Adi Shamir, Leonard Adleman at MIT devise and publish a practical methodology for PKI (now known as RSA)

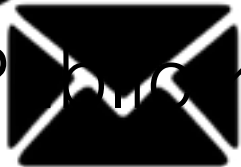
PKI 101



Private Key



Public Key



SSL & TLS

SSL = Secure Sockets Layer

- v 2.0 - Insecure

- v 3.0 - Currently secure - used by OS X and CAs

TLS = Transport Layer Security

- v 1.0 - designed as an upgrade to SSL 3.0

- v 1.1 - Update to TLS 1.0

- v 1.2 - Currently secure.

Certificates & Keys



Certificates verify ownership of public keys

Keys are generated through Certificate creation

Certificates






CA ✓



Certificates for most Certificate Authorities come preconfigured on most modern OSes (Mac, PC, iOS etc)




















TWCA Root Certification Authority

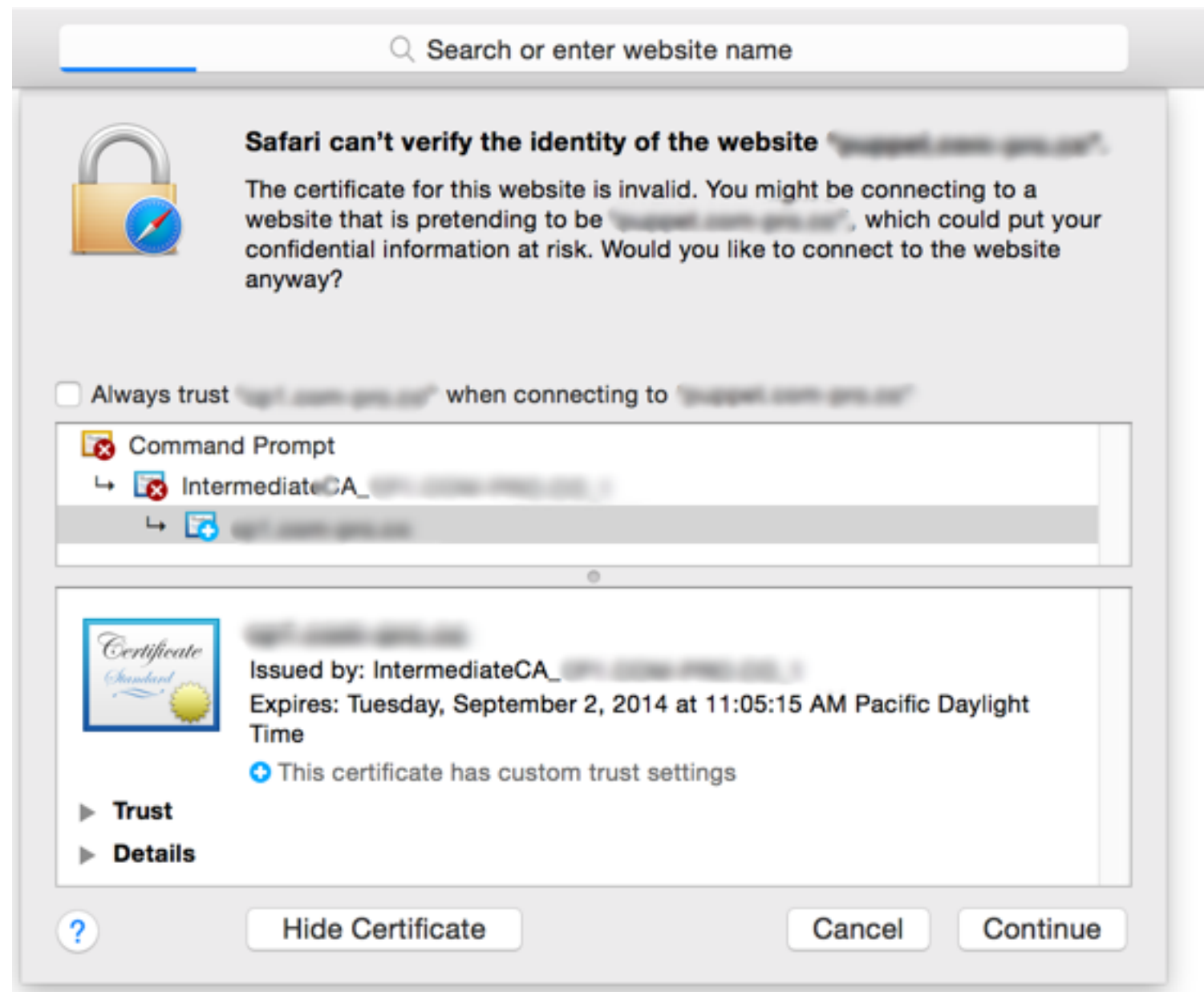
Root certificate authority

Expires: Tuesday, December 31, 2030 at 7:59:59 AM Pacific Standard Time

✔ This certificate is valid

Name	Kind	Date Modified	Expires
 Certigna	certificate	--	Jun 29, 2027, 8:13:05 AM
 Certinomis - Autorité Racine	certificate	--	Sep 17, 2028, 1:28:59 AM
 Certinomis - Root CA	certificate	--	Oct 21, 2033, 2:17:18 AM
 certSIGN ROOT CA	certificate	--	Jul 4, 2031, 10:20:04 AM
 Certum CA	certificate	--	Jun 11, 2027, 3:46:39 AM
 Certum Trusted Network CA	certificate	--	Dec 31, 2029, 4:07:37 AM
 Certum Trusted Network CA 2	certificate	--	Oct 6, 2046, 1:39:56 AM
 Chambers of Commerce Root	certificate	--	Sep 30, 2037, 9:13:44 AM
 Chambers of Commerce Root - 2008	certificate	--	Jul 31, 2038, 5:29:50 AM
 China Internet Network Information Center EV Certificates Root	certificate	--	Aug 31, 2030, 12:11:25 AM
 Cisco Root CA 2048	certificate	--	May 14, 2029, 1:25:42 PM
 Class 1 Public Primary Certification Authority	certificate	--	Aug 1, 2028, 4:59:59 PM
 Class 1 Public Primary Certification Authority	certificate	--	Aug 2, 2028, 4:59:59 PM
 Class 1 Public Primary Certification Authority - G2	certificate	--	Aug 1, 2028, 4:59:59 PM
 Class 2 Primary CA	certificate	--	Jul 6, 2019, 4:59:59 PM
 Class 2 Public Primary Certification Authority	certificate	--	Aug 1, 2028, 4:59:59 PM
 Class 2 Public Primary Certification Authority	certificate	--	Aug 2, 2028, 4:59:59 PM

Not infallible (see: DigiNotar, Comodo & Yahoo, Mozilla, Wordpress, Tor etcetera)



Self-signed certificate

Not a trusted root certificate - not necessarily insecure

Company ID badge, not a passport

Self-signed Certificates used in OS X Server for:

- Mail (IMAP, POP & SMTP)
- Calendar & Contacts
- Messages
- Open Directory
- Websites

Keys



- Created using OpenSSL

```
openssl genrsa -des3 -out mynewkey.key 2048
```

- Created using Keychain Access (Certificate Assistant)
- Created using Server.app

How Keys are made:

A computer...



⋮

...uses terrifying math

Terrifying Math

⋮

...run through a key generation algorithm

Key Generation Algorithm (e.g., RSA)

...to generate two matched keys



Public Key



Private Key

Session Keys



Private Key



Public Key



Public keys can encrypt data that can only be unlocked using a Private key.

Q: Can a Private key encrypt data that can be unlocked by a Public Key? Does it work the other way round?

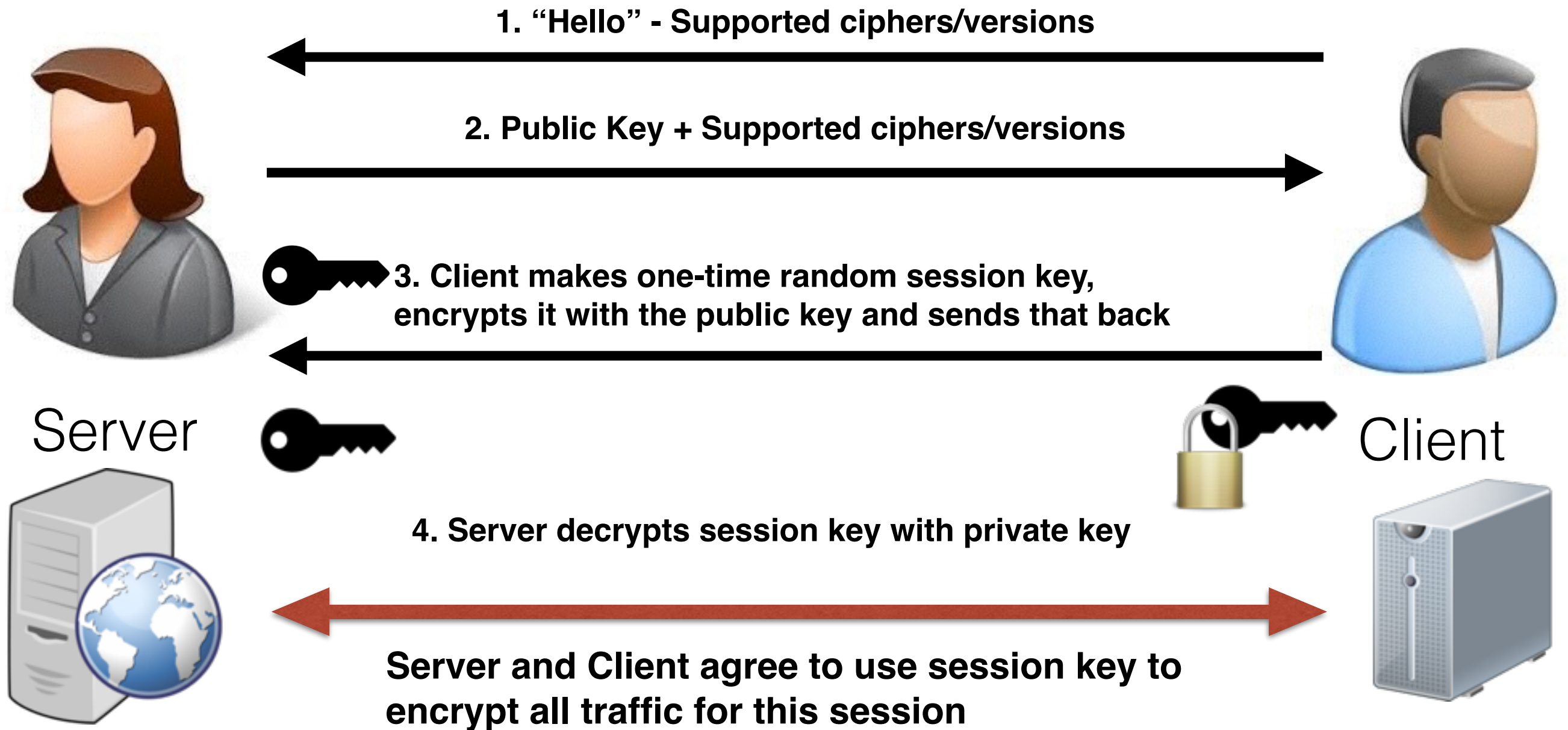
A: (Short answer) No.

A: (Longer answer) Kind of. It's not secure, but it's done for a few very specific uses (e.g., Digital signatures)

Session Keys

- SSL/TLS both use a combination of symmetric and asymmetric encryption
- Asymmetric/PKI encryption is far stronger for authentication
- Symmetric encryption is far simpler and far faster
- Why not have the best of both worlds?

Anatomy of an SSL/TLS session



s/mime

“Secure/Multipurpose Internet Mail Extensions”

- PKI for email
- Built in to iOS/Mac OS X/Outlook/Thunderbird
- Two delicious flavors:

Class 1 - Verifies sender identity

Class 2 - Verifies identity & signs message
with Digital Signature

Signatures and hashes

Hash = Mathematical computation based on the content of the data

Hello World in ASCII =

072 101 108 108 111 032 087 111 114 108 100 = **1052**

If my message arrives with a hashed value of 1052 then I know it's okay, right? **Wrong**

Digital Signature = Hash signed with private key

- Message not tampered with, identifies the sender

Good: SHA-2 -256, 384, 512 bit flavors

Bad: SHA-1, MD5

Other: PBKDF2 (Used in to convert passwords to keys in Filevault 2)

You.

(Or: where it all goes horribly wrong.)

“Given a long enough key (and correctly used) is practically unbreakable”

Human beings are not
reliable.

The Enigma Machine

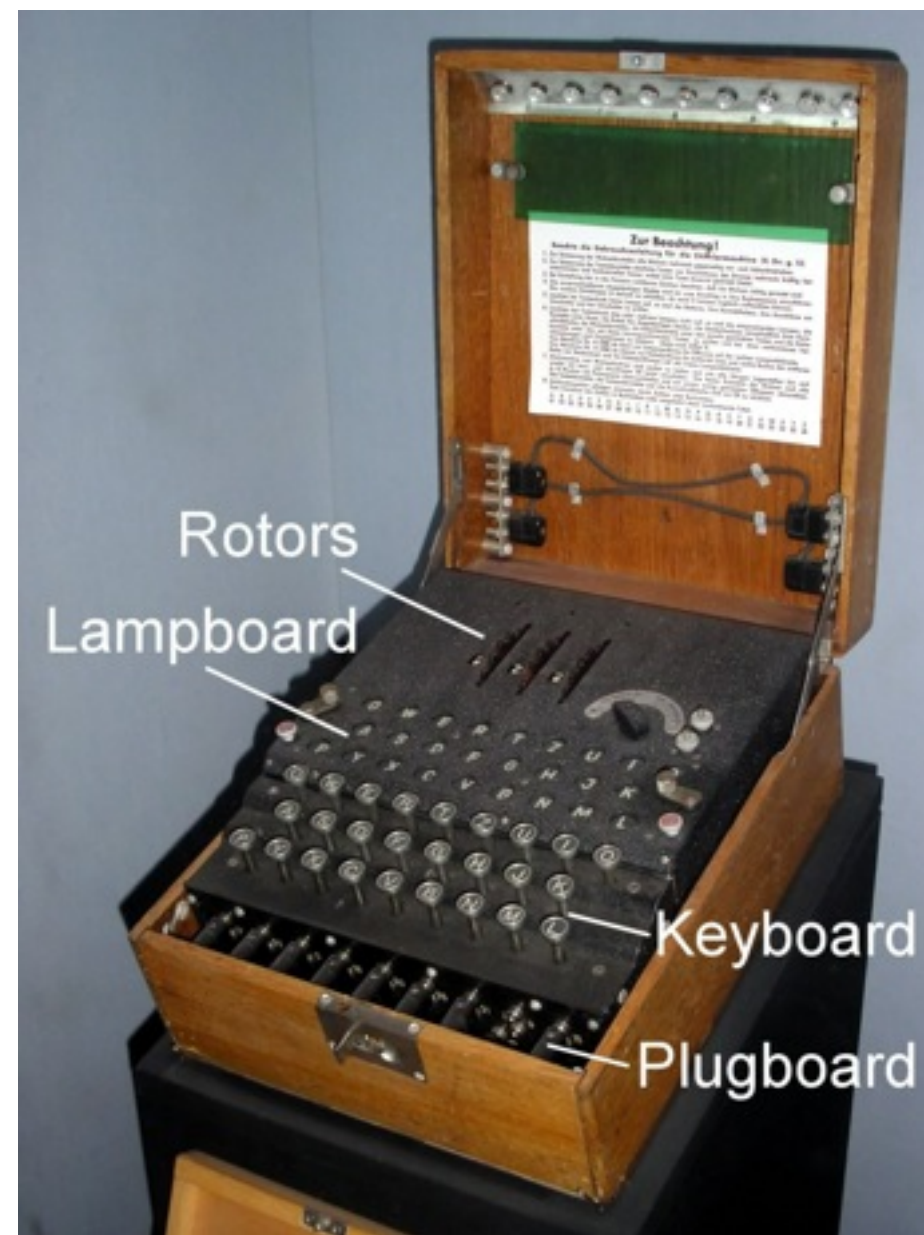
Up to 100,000,000 pair arrangements

Operators often poorly trained

Training manual contained actual ciphers and keys

Send the same message using multiple ciphers

Operators hitting the same key to set each rotor





[Augusto Barros](#)

@apbarros



[Follow](#)

Wanna know the pwd for the Brasil world cup security center WiFi nw? It's on the whiteboard ;-)
[#fail](#)

[Reply](#) [Retweet](#) [Favorite](#) [More](#)



RETWEETS
3,266

FAVORITES
817



12:31 PM - 23 Jun 2014

Flag media

What can you do?

- Talk to people about their industry (PCI, HIPAA etc).
- Educate clients about security
- Encourage proper documentation
- Use the right tool for the job
- Invest time in educating yourself (and staying up to date)
- Have a lot of insurance
- Eat your own dog food

“Security is mostly a superstition. It does not exist in nature.”

- Helen Keller

Resources

afp548.com

giac.org

<http://www.sans.org/security-resources/blogs>

rsa.com

HIPAA - <http://www.hhs.gov/ocr/privacy/>

mactech.com

“Seizing the Enigma” - David Kahn