



**The NSA**  
*The only part of government  
that actually listens.*

**Don't give them anything for free**

It's your home, you fight



BetterCrypto · org  
Applied Crypto Hardening

# Who

Wolfgang Breyha (uni VIE),

David Durvaux (CERT.be),

Tobias Dussa (KIT-CERT),

L. Aaron Kaplan (CERT.at),

Christian Mock (coretec),

Daniel Kovacic (A-Trust),

Manuel Koschuch (FH Campus Wien),

Adi Kriegisch (VRVis),

Ramin Sabet (A-Trust),

Aaron Zauner (azet.org),

Pepi Zawodsky (maclemon.at),

New contributors:

IAIK,

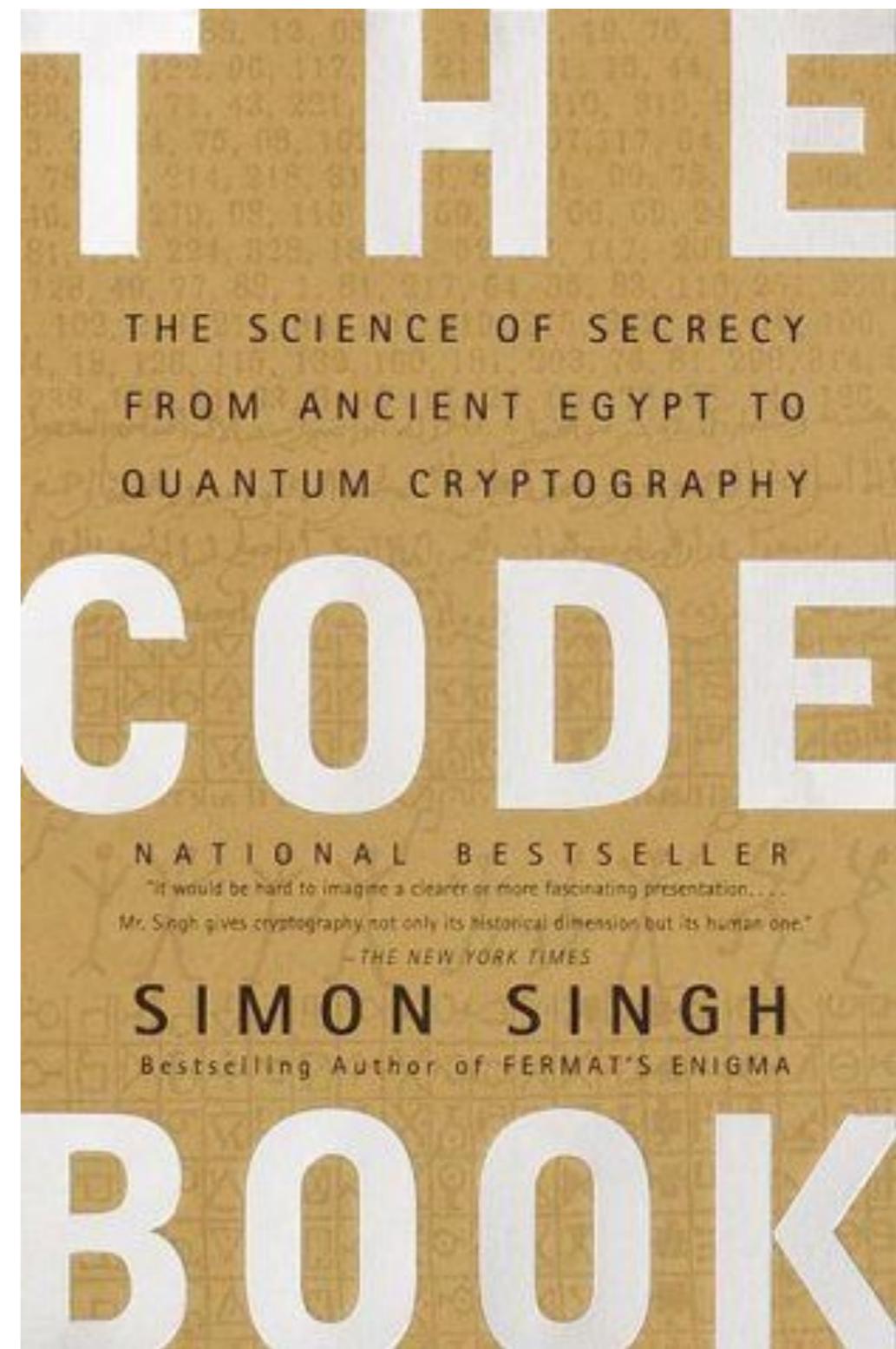
A-Sit

# Agenda

- Pieces of History
- Introduction to BetterCrypto project
- Symetric Ciphering
- Asymmetric Cryptography
- Ciphersuites
- Practical Settings
- Heartbeat
- Conclusion

# Pieces of History

- Caesar
- Mary Queen of Scots
  - Trial against Queen Elizabeth
  - Was executed after her code was broken (1587)
- 2nd World War
  - Enigma and the U-Boat
  - Secret in code book
  - Navajo Indian as radio



# BetterCrypto

- Crypto is cryptic
- A lot of difficult concepts
- A lot of algorithms
- A lot of parameters
- ...

# BetterCrypto

- Really difficult for systems administrators
  - A “cookbook” can help!
  - That’s BetterCrypto

# BetterCrypto is not...

- A crypto course
- A static document

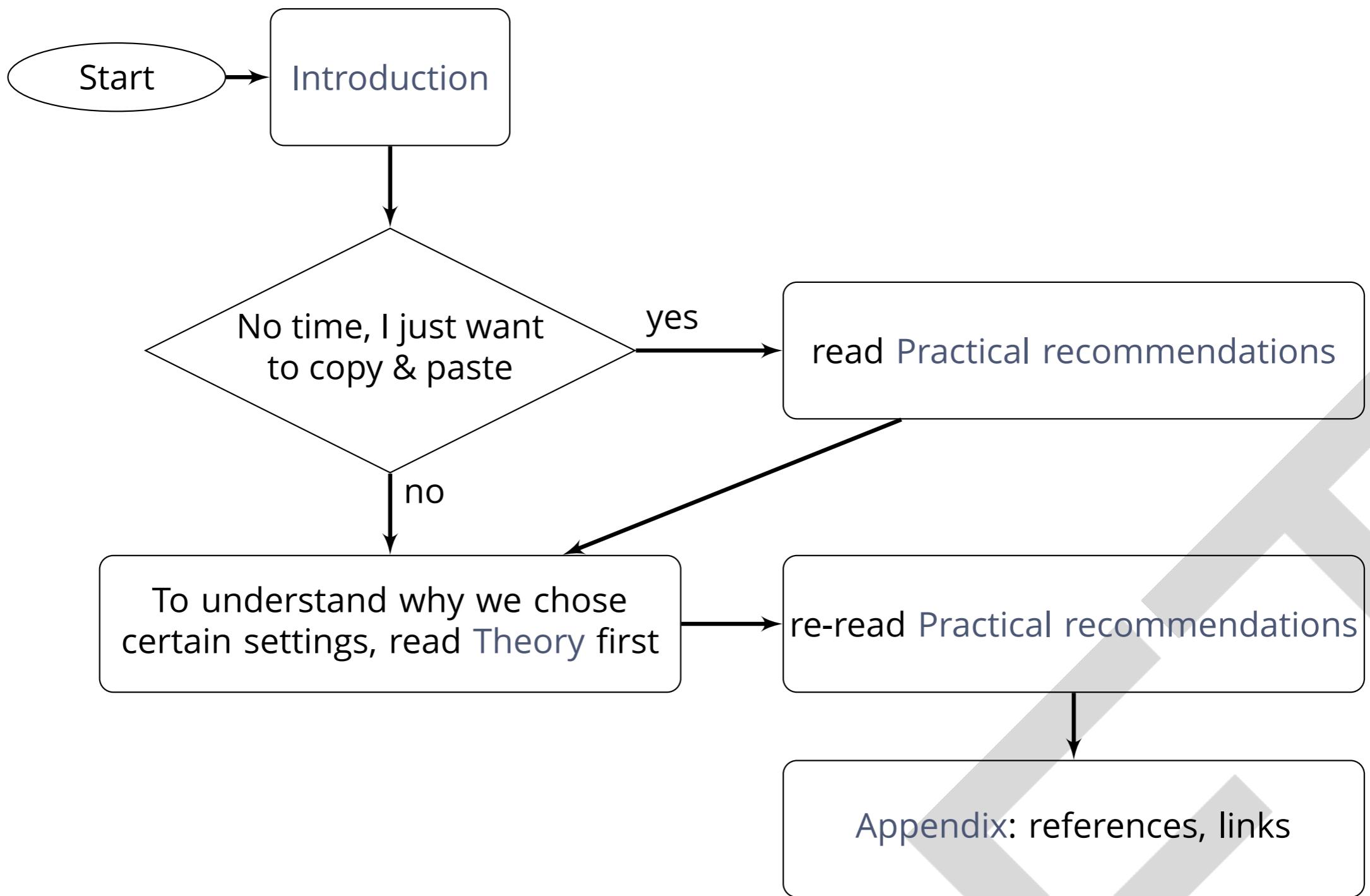
# BetterCrypto in short

- Community effort to produce best practices
- Continuous effort
- Mixed expertises
- Open to comments / suggestions / improvements

# BetterCrypto in 2 parts

- First part = configurations
  - The most important part
  - Cover as many tools as possible
- Second part = theory
  - Explain and justify choose we made
    - Transparency

# How to use?

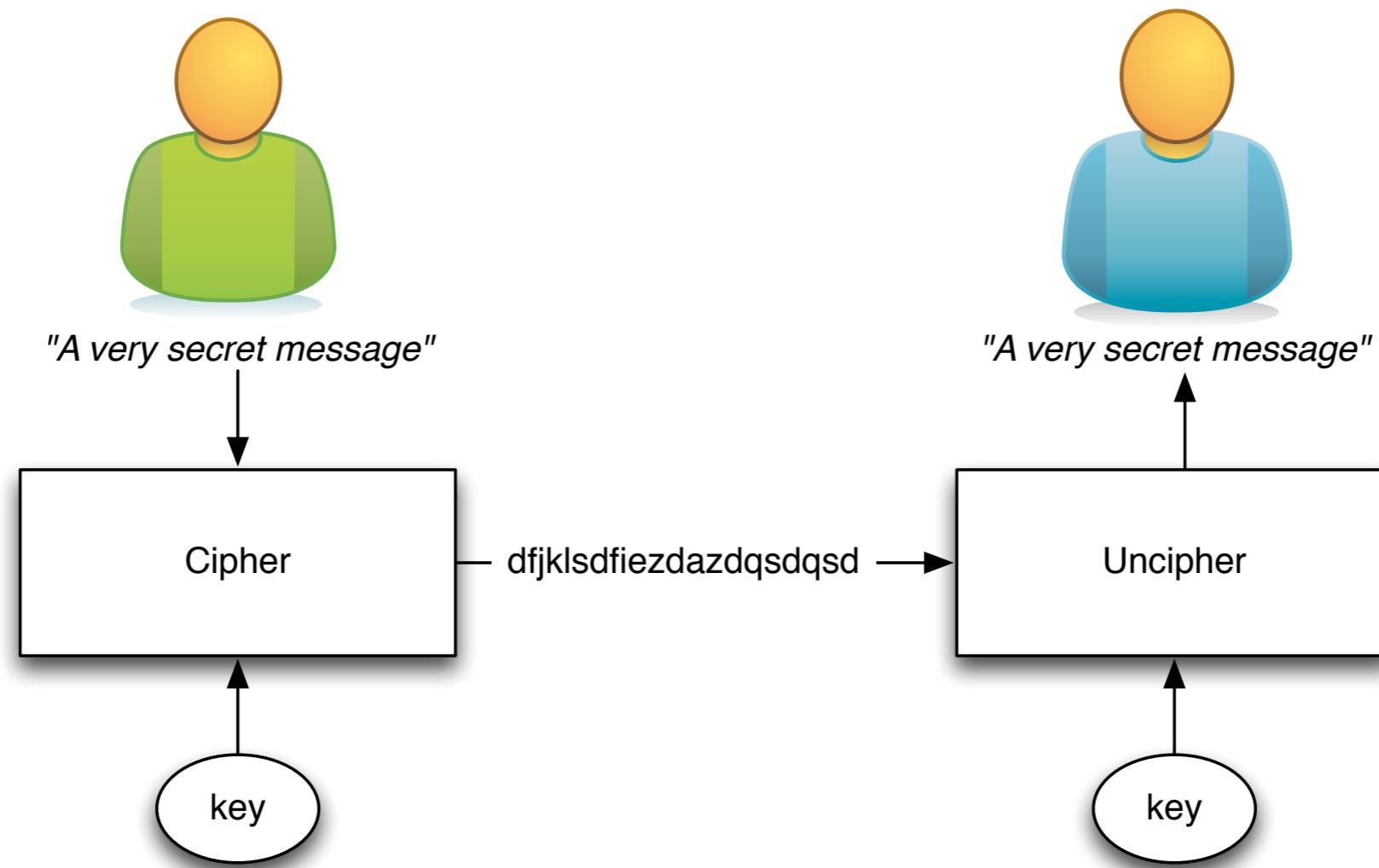


# Crypto in a nutshell

- 2 types of goals:
  - protect the contact of the message
  - identify the author
- Can be combined

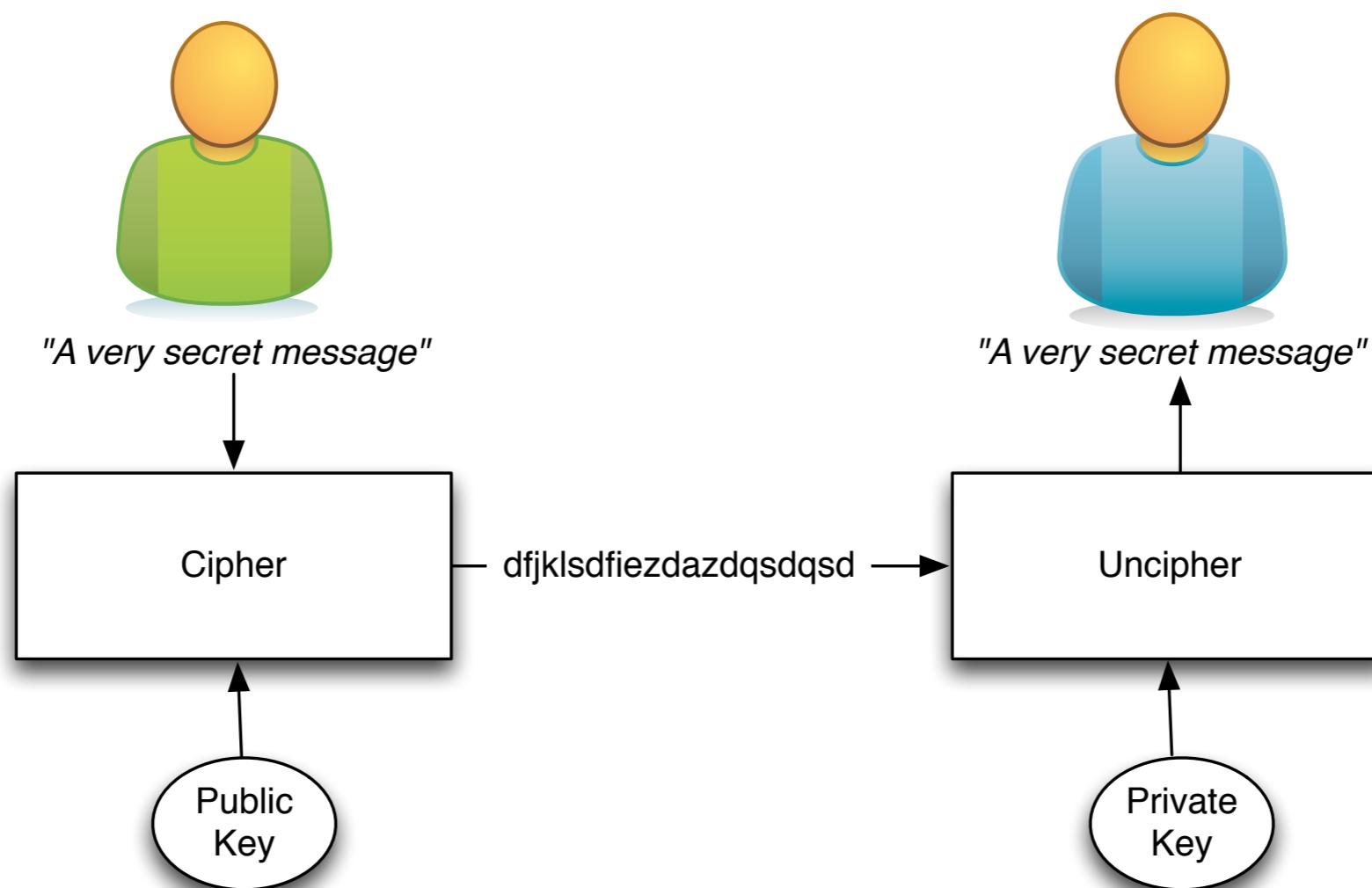
# Symmetric Ciphering

- The key is shared



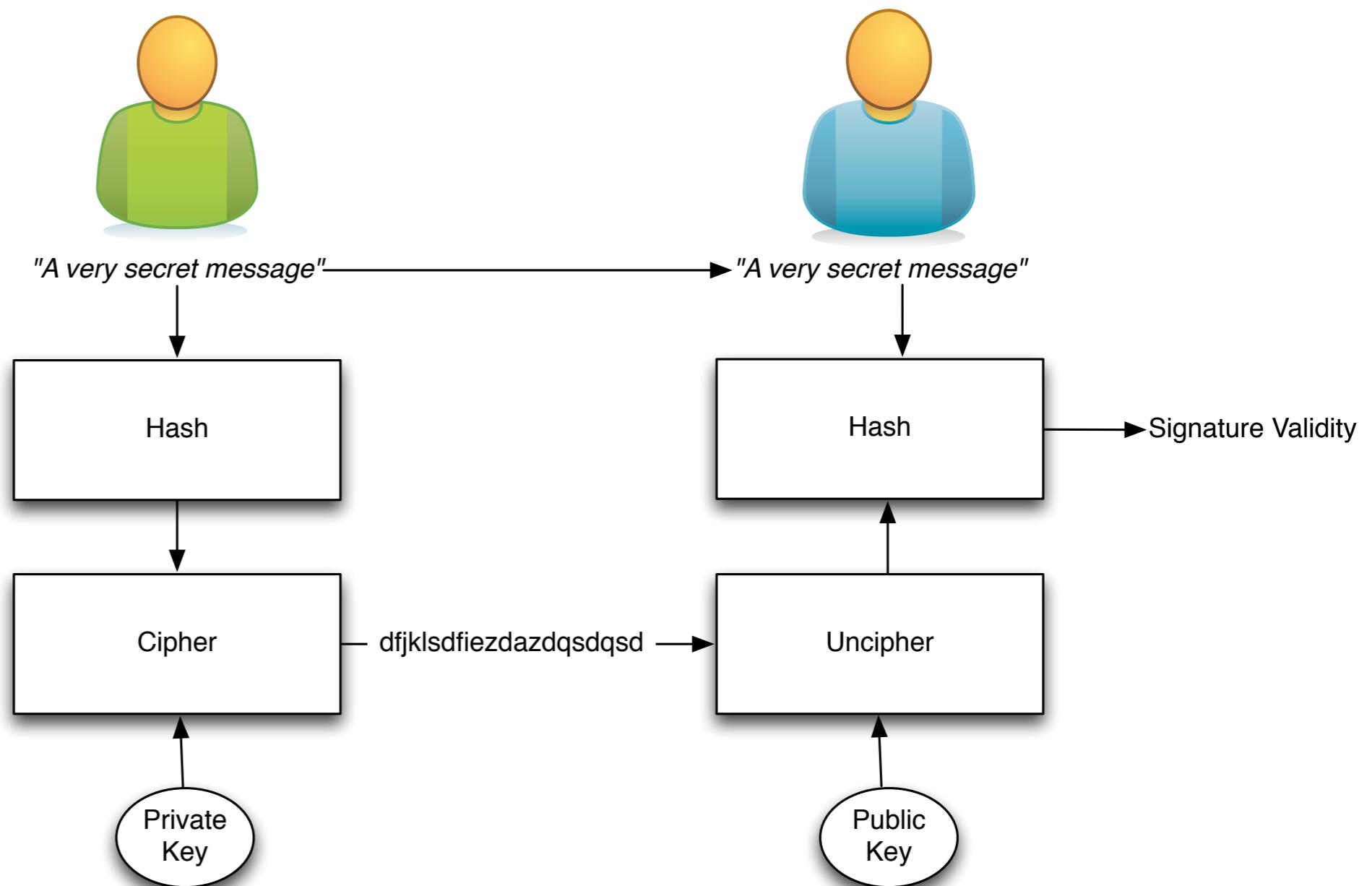
# Asymmetric Ciphering

- Public key is published
- Private key HAS to be secured



# Signing

- Author identity is proved



# The asymmetric magic

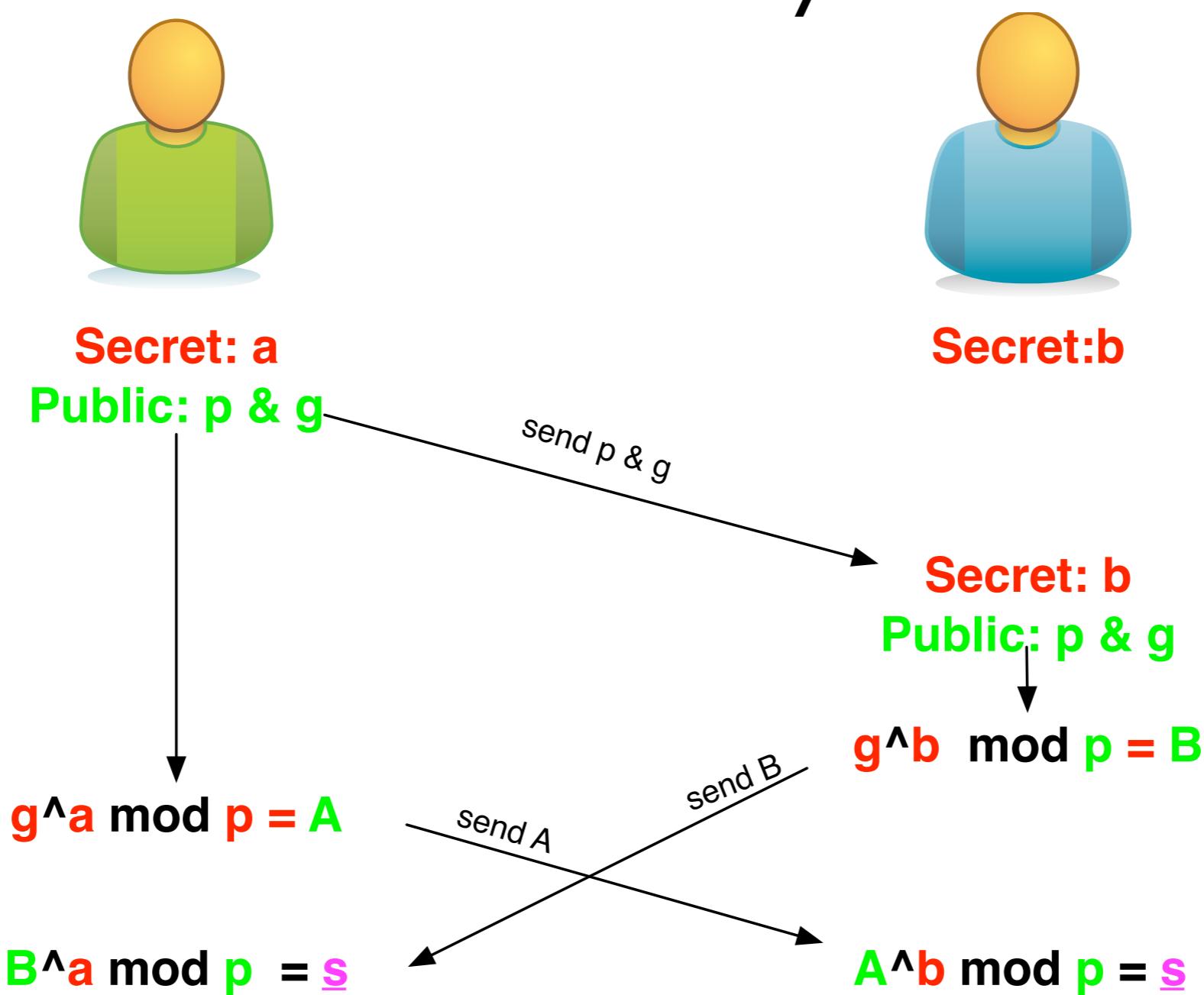
- RSA “formula” :  $c = m^e \text{ mod } n$ 
  - with
    - $c$  which is the ciphertext
    - $m$  is the cleartext message
    - $e$  and  $n$  are the public key
  - Uncipher with  $m = c^d \text{ mod } n$ 
    - $d$  being the private key

# Some algorithms

- Hash
  - SHA1
  - SHA256
  - SHA512
- Key Exchange
  - Diffie Elleman

# Diffie-Hellman

- How to share a secret key?



# Ephemeral Diffie-Hellman

- Regular mode
  - Public and private keys are kept
- Ephemeral mode
  - New keys are generated each time
    - By one of the parties at least

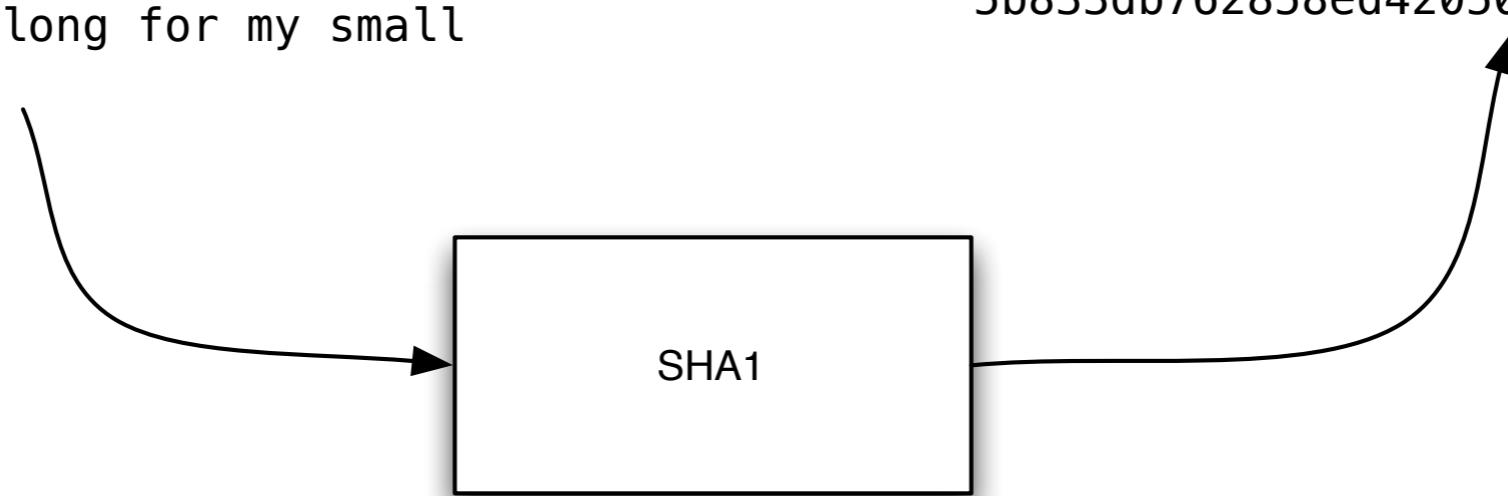
# Hashing

- Take long piece of data and produce a probably unique signature
- Probability of collision for SHA1:
  - 1 over **146150163733090291820368483271628**

This is a really long text.

I can even put a full book over there  
but it would be too long for my small  
schema ;).

5b833db762858ed42050809816e4028421b6e2a3



# SSL

- Explain

# Stream vs Block Cipher

- Stream cipher
  - Generate an “infinite” key stream
  - Difficult to correctly use
    - Re-use of keys
    - Faster
- Block cipher
  - Cipher by block with padding
  - Could include integrity protection

# Some algorithms

- Symetric
  - AES (Rijndael)
  - Camellia
- Asymmetric
  - GPG / PGP
  - RSA

# Algorithm vs Implementation!

- Heartbeat

# Heartbeat

```
/* Enter response type, length and copy payload */  
*bp++ = TLSI_HB_RESPONSE;  
s2n(payload, bp);  
memcpy(bp, pl, payload);
```

- payload (pl) and payload\_length (payload) are controlled by attacker
- memcpy will copy a part of the victim memory to the reply...

# BetterCrypto CipherSuite

- 2 cipher suites
  - version A
    - stronger
    - less supported client
  - version B
    - weaker
    - more “universal”

# Cipher Suite A

- TLS 1.2
- Perfect forward secrecy / ephemeral Diffie Hellman
- Strong MACs (SHA-2) or
- GCM as Authenticated Encryption scheme

ID	OpenSSL Name	Version	KeyEx	Auth	Cipher	MAC
0x009F	DHE-RSA-AES256-GCM-SHA384	TLSv1.2	DH	RSA	AESGCM(256)	AEAD
0x006B	DHE-RSA-AES256-SHA256	TLSv1.2	DH	RSA	AES(256) (CBC)	SHA256
0xC030	ECDHE-RSA-AES256-GCM-SHA384	TLSv1.2	ECDH	RSA	AESGCM(256)	AEAD
0xC028	ECDHE-RSA-AES256-SHA384	TLSv1.2	ECDH	RSA	AES(256) (CBC)	SHA384

# CiperSuite B

- TLS 1.2, TLS 1.1, TLS 1.0
- Allowing SHA-1

# Cipher Suite B

ID	OpenSSL Name	Version	KeyEx	Auth	Cipher	MAC
0x009F	DHE-RSA-AES256-GCM-SHA384	TLSv1.2	DH	RSA	AESGCM(256)	AEAD
0x006B	DHE-RSA-AES256-SHA256	TLSv1.2	DH	RSA	AES(256)	SHA256
0xC030	ECDHE-RSA-AES256-GCM-SHA384	TLSv1.2	ECDH	RSA	AESGCM(256)	AEAD
0xC028	ECDHE-RSA-AES256-SHA384	TLSv1.2	ECDH	RSA	AES(256)	SHA384
0x009E	DHE-RSA-AES128-GCM-SHA256	TLSv1.2	DH	RSA	AESGCM(128)	AEAD
0x0067	DHE-RSA-AES128-SHA256	TLSv1.2	DH	RSA	AES(128)	SHA256
0xC02F	ECDHE-RSA-AES128-GCM-SHA256	TLSv1.2	ECDH	RSA	AESGCM(128)	AEAD
0xC027	ECDHE-RSA-AES128-SHA256	TLSv1.2	ECDH	RSA	AES(128)	SHA256
0x0088	DHE-RSA-CAMELLIA256-SHA	SSLv3	DH	RSA	Camellia(256)	SHA1
0x0039	DHE-RSA-AES256-SHA	SSLv3	DH	RSA	AES(256)	SHA1
0xC014	ECDHE-RSA-AES256-SHA	SSLv3	ECDH	RSA	AES(256)	SHA1
0x0045	DHE-RSA-CAMELLIA128-SHA	SSLv3	DH	RSA	Camellia(128)	SHA1
0x0033	DHE-RSA-AES128-SHA	SSLv3	DH	RSA	AES(128)	SHA1
0xC013	ECDHE-RSA-AES128-SHA	SSLv3	ECDH	RSA	AES(128)	SHA1
0x0084	CAMELLIA256-SHA	SSLv3	RSA	RSA	Camellia(256)	SHA1
0x0035	AES256-SHA	SSLv3	RSA	RSA	AES(256)	SHA1
0x0041	CAMELLIA128-SHA	SSLv3	RSA	RSA	Camellia(128)	SHA1
0x002F	AES128-SHA	SSLv3	RSA	RSA	AES(128)	SHA1

# Key Length

*On the choice between AES256 and AES128: I would never consider using AES256, just like I don't wear a helmet when I sit inside my car. It's too much bother for the epsilon improvement in security.”*

— Vincent Rijmen in a personal mail exchange Dec 2013

- Symetric
  - 128 bits
- Aysmetric
  - 3248 bits (RSA)

### Choose a Method

- Lenstra and Verheul Equations (2000)
- Lenstra Updated Equations (2004)
- ECRYPT II Recommendations (2012)
- NIST Recommendations (2012)
- ANSSI Recommendations (2010)
- Fact Sheet NSA Suite B Cryptography (2013)
- Network Working Group RFC3766 (2004)
- BSI Recommendations (2014)

[Compare all Methods](#)

### 1 Reference for the comparison

You can enter the year until when your system should be protected and see the corresponding key sizes or you can enter a key/hash/group size and see until when you would be protected.

Enter an elliptic curve key size:  bits

2 Compare

Method	Date	Symmetric	Asymmetric	Discrete Logarithm Key	Group	Elliptic Curve	Hash
[1] Lenstra / Verheul <a href="#">?</a>	2084	135	7813 6816	241	7813	257	269
[2] Lenstra Updated <a href="#">?</a>	2090	128	4440 6974	256	4440	256	256
[3] ECRYPT II	2031 - 2040	128	3248	256	3248	256	256
[4] NIST	> 2030	128	3072	256	3072	256	256
[5] ANSSI	> 2020	128	4096	200	4096	256	256
[6] NSA	-	128	-	-	-	256	256
[7] RFC3766 <a href="#">?</a>	-	136	3707	272	3707	257	-
[8] BSI (signature only)	> 2020	-	1976	256	2048	250	256

# Compatibility (B suite)



## Handshake Simulation

<a href="#">Bing Oct 2013</a>	TLS 1.0	TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) FS	256
<a href="#">Chrome 31 / Win 7</a>	<b>TLS 1.2</b>	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) FS	256
<a href="#">Firefox 10.0.12 ESR / Win 7</a>	TLS 1.0	TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x88) FS	256
<a href="#">Firefox 17.0.7 ESR / Win 7</a>	TLS 1.0	TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x88) FS	256
<a href="#">Firefox 21 / Fedora 19</a>	TLS 1.0	TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x88) FS	256
<a href="#">Firefox 24 / Win 7</a>	TLS 1.0	TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x88) FS	256
<a href="#">Googlebot Oct 2013</a>	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) FS	256
<a href="#">IE 6 / XP</a> No FS <sup>1</sup> No SNI <sup>2</sup>			Fail <sup>3</sup>
<a href="#">IE 7 / Vista</a>	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) FS	256
<a href="#">IE 8 / XP</a> No FS <sup>1</sup> No SNI <sup>2</sup>			Fail <sup>3</sup>
<a href="#">IE 8-10 / Win 7</a>	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) FS	256
<a href="#">IE 11 / Win 7</a>	<b>TLS 1.2</b>	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) FS	256
<a href="#">IE 11 / Win 8.1</a>	<b>TLS 1.2</b>	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) FS	256
<a href="#">Java 6u45</a> No SNI <sup>2</sup>			Fail <sup>3</sup>
<a href="#">Java 7u25</a>			Fail <sup>3</sup>
<a href="#">OpenSSL 0.9.8y</a>	TLS 1.0	TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) FS	256
<a href="#">OpenSSL 1.0.1e</a>	<b>TLS 1.2</b>	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) FS	256
<a href="#">Opera 17 / Win 7</a>	<b>TLS 1.2</b>	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (0x6b) FS	256
<a href="#">Safari 5.1.9 / OS X 10.6.8</a>	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) FS	256
<a href="#">Safari 6 / iOS 6.0.1</a>	<b>TLS 1.2</b>	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) FS	256
<a href="#">Safari 6.0.4 / OS X 10.8.4</a>	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) FS	256
<a href="#">Safari 7 / OS X 10.9</a>	<b>TLS 1.2</b>	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) FS	256
<a href="#">Tor 17.0.9 / Win 7</a>	TLS 1.0	TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x88) FS	256
<a href="#">Yahoo Slurp Oct 2013</a>	TLS 1.0	TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x88) FS	256

# Tools covered

- Webservers
  - Apache
  - lighttpd
  - nginx
  - Microsoft IIS

# Tools covered

- SSH
  - Open SSH
  - Cisco ASA
  - Cisco IOS

# Tools covered

- Mail servers
  - Dovecot
  - cyrus-imapd
  - Postfix
  - Exim

# Tools covered

- VPN
  - IPSec
  - CheckPoint Firewall- I
  - OpenVPN
  - PPPTP
  - Cisco ASA
  - OpenSWAN
  - tinc

# Tools covered

- PGP/GPG
- IPMI/ILO
- Instant Messaging
  - ejabberd
  - OTR
  - Charybdis
  - SILC

# Tools covered

- Database systems
  - Oracle
  - MySQL
  - DB2
  - PostgreSQL

# Tools covered

- Proxy
  - squid
  - Bluecoat
  - Pound
  - Kerberos

# Mail Encryption

- GPG / PGG
- [ a little bit over there ]

# Futur / Idea

- Configuration Generator (online)
- A friendly copy/paste version
- Other tools

# Conclusion

# References

- BetterCrypto.org
- <https://git.bettercrypto.org/ach-master.git>
- <http://lists.cert.at/cgi-bin/mailman/listinfo/ach>
- Contact
  - [david@autopsit.org](mailto:david@autopsit.org)
  - [@ddurvaux](https://twitter.com/ddurvaux)