“Understanding tools for a more secure internet” 2nd cyber-security week @ CIC-IPN.

Luis J. Dominguez Perez
CONACyT. CIMAT

luis.dominguez@cimat.mx

CONACYT

Centro de Investigación
en Computación
Instituto Politécnico Nacional

CIMAT

Public Key Cryptography

- In 1976, Whitfield Diffie, and Martin Hellman published their famous article: “New Directions in Cryptography”

- A bit before, Ralph Merkle invented a public key construction for his lectures: “Secure communication over insecure channels” in 1982

- Originally, the concepts were discovered by James Ellis; however, these were kept secret as the were classified information by the GCHQ from 1969 to 1997.

- Additionally, Malcolm Williamson, and Clifford Cocks from the GCHQ, also discovered the Diffie-Hellman key exchange, and the RSA encryption

Public Key Cryptography

- In 1976, Whitfield Diffie, and Martin Hellman published their famous article: "New Directions in Cryptography"

- A bit before, Ralph Merkle invented a public key construction for his lectures: "Secure communication over insecure channels" in 1982

- Originally, the concepts were discovered by James Ellis; however, these were kept secret as the were classified information by the GCHQ from 1969 to 1997.

- Additionally, Malcolm Williamson, and Clifford Cocks from the GCHQ, also discovered the Diffie-Hellman key exchange, and the RSA encryption

ey Cryptography

76, Whitfield Diffie, and Martin Hellman shed their famous article: “New tions in Cryptography”

before, Ralph Merkle invented a public onstruction for his lectures: “Secure nunication over insecure channels” in 1982

- Originally, the concepts were discovered by James Ellis; however, these were kept secret as the were classified information by the GCHQ from 1969 to 1997.
- Additionally, Malcolm Williamson, and Clifford Cocks from the GCHQ, also discovered the Diffie-Hellman key exchange, and the RSA encryption

- Befor
  on cr
  Natic
- Up to
  algori
  USA)

phy

fie, and Martin Hellman
; article: “New
aphy”

erkle invented a public
s lectures: “Secure
nsecure channels” in 1982

- Originally, the concepts were discovered by James Ellis; however, these were kept secret as the were classified information by the GCHQ from 1969 to 1997.
- Additionally, Malcolm Williamson, and Clifford Cocks from the GCHQ, also discovered the Diffie-Hellman key exchange, and the RSA encryption

- Before “New Directior
on cryptography in the
National Security Ager
- Up to mid 1990’s, exp
algorithms was conside
USA).

ellman

ublic
re
in 1982

- Originally, the concepts were discovered by James Ellis; however, these were kept secret as the were classified information by the GCHQ from 1969 to 1997.
- Additionally, Malcolm Williamson, and Clifford Cocks from the GCHQ, also discovered the Diffie-Hellman key exchange, and the RSA encryption

- Before “New Directions. . .”, all of the
on cryptography in the USA was under
National Security Agency (NSA)
- Up to mid 1990’s, exporting cryptograp
algorithms was considered treason (in t
USA).

- Originally, the concepts were discovered by James Ellis; however, these were kept secret as the were classified information by the GCHQ from 1969 to 1997.

- Additionally, Malcolm Williamson, and Clifford Cocks from the GCHQ, also discovered the Diffie-Hellman key exchange, and the RSA encryption

- Before "New Directions...", all of the research on cryptography in the USA was under the National Security Agency (NSA)

- Up to mid 1990's, exporting cryptographic algorithms was considered treason (in the USA).

- Originally, the concepts were discovered by James Ellis; however, these were kept secret as the were classified information by the GCHQ from 1969 to 1997.

- Additionally, Malcolm Williamson, and Clifford Cocks from the GCHQ, also discovered the Diffie-Hellman key exchange, and the RSA encryption

- Before "New Directions. . .", all of the research on cryptography in the USA was under the National Security Agency (NSA)

- Up to mid 1990's, exporting cryptographic algorithms was considered treason (in the USA).

- Afterwards, the prohibition was only kept for high security algorithms if they were readable by machines (source code, binary executable file, etc.).

- Originally, the concepts were discovered by James Ellis; however, these were kept secret as the were classified information by the GCHQ from 1969 to 1997.

- Additionally, Malcolm Williamson, and Clifford Cocks from the GCHQ, also discovered the Diffie-Hellman key exchange, and the RSA encryption

- Before "New Directions. . .", all of the research on cryptography in the USA was under the National Security Agency (NSA)

- Up to mid 1990's, exporting cryptographic algorithms was considered treason (in the USA).

- Afterwards, the prohibition was only kept for high security algorithms if they were readable by machines (source code, binary executable file, etc.).

e “New Directions. . .”, all of the research
yptography in the USA was under the
onal Security Agency (NSA)

o mid 1990’s, exporting cryptographic
thms was considered treason (in the
.

wards, the prohibition was only kept for
security algorithms if they were readable
achines (source code, binary executable
tc.).

RSA

Designed in 1977 by Ron Rivest, Adi Shamir, and Leonar Adleman.

- Let p y q be two different random large prime numbers
- The modulus n is the product of p, and q
- The function $\Phi(n) = (p-1)(q-1)$
- Choose $1 < e < \Phi(n)$, such that $\mathrm{GCD}(e, \Phi(n)) = 1$; $e = 2^{16}+1$ typically
- Compute $d \equiv e^{-1} \bmod \Phi(n)$

The public key is (e, n). The private key is (d, p, q).

RSA enc

Given a

- **Encr**
- **Decr**

ns. . .”, all of the research
e USA was under the
ncy (NSA)

orting cryptographic
ered treason (in the

ition was only kept for
ns if they were readable
ode, binary executable

RSA

Designed in 1977 by Ron Rivest, Adi Shamir, and Leonar Adleman.

- Let p y q be two different random large prime numbers
- The modulus n is the product of p, and q
- The function $\Phi(n) = (p-1)(q-1)$
- Choose $1 < e < \Phi(n)$, such that $\text{GCD}(e, \Phi(n)) = 1$; $e = 2^{16} + 1$ typically
- Compute $d \equiv e^{-1} \bmod \Phi(n)$

The public key is (e, n). The private key is (d, p, q).

RSA encryption, and

Given a message $M < n$

- **Encryption**. $C = M^e$
- **Decryption**. $M = C^d$

research
the

hic
he

pt for
adable
table

RSA

Designed in 1977 by Ron Rivest, Adi Shamir, and Leonar Adleman.

- Let p y q be two different random large prime numbers
- The modulus n is the product of p, and q
- The function $\Phi(n) = (p-1)(q-1)$
- Choose $1 < e < \Phi(n)$, such that $\mathrm{GCD}(e, \Phi(n)) = 1$; $e = 2^{16} + 1$ typically
- Compute $d \equiv e^{-1} \bmod \Phi(n)$

The public key is (e, n). The private key is (d, p, q).

RSA encryption, and decryption

Given a message $M < n$

- **Encryption**. $C = M^e \bmod n$
- **Decryption**. $M = C^d \bmod n$

RSA

Designed in 1977 by Ron Rivest, Adi Shamir, and Leonar Adleman.

- Let p y q be two different random large prime numbers
- The modulus n is the product of p, and q
- The function $\Phi(n) = (p-1)(q-1)$
- Choose $1 < e < \Phi(n)$, such that $\mathrm{GCD}(e, \Phi(n)) = 1$; $e = 2^{16} + 1$ typically
- Compute $d \equiv e^{-1} \bmod \Phi(n)$

The public key is (e, n). The private key is (d, p, q).

RSA encryption, and decryption

Given a message $M < n$

- **Encryption**. $C = M^e \bmod n$
- **Decryption**. $M = C^d \bmod n$

d in 1977 by Ron Rivest, Adi Shamir, and Adleman.

y q be two different random large prime ers

modulus n is the product of p, and q

function $\Phi(n) = (p-1)(q-1)$

se $1 < e < \Phi(n)$, such that $(e, \Phi(n)) = 1$; $e = 2^{16} + 1$ typically

pute $d \equiv e^{-1} \bmod \Phi(n)$

lic key is (e, n). The private key is .

RSA encryption, and decryption

Given a message $M < n$

- **Encryption**. $C = M^e \bmod n$
- **Decryption**. $M = C^d \bmod n$

Example

- $p = 1$
- $n = p$
- $\Phi(n)$
- GCD
- $d = e$

- Publi
- Priva

Rivest, Adi Shamir, and

rent random large prime

product of p, and q

$(p-1)(q-1)$

such that

$= 2^{16}+1$ typically

d $\Phi(n)$

The private key is

RSA encryption, and decryption

Given a message $M < n$

- **Encryption**. $C = M^e \bmod n$
- **Decryption**. $M = C^d \bmod n$

Example

- $p = 11,\ q = 13$
- $n = p \cdot q = 11 \cdot 13 =$
- $\Phi(n) = (p-1)(q-1$
- GCD$(e, \Phi(n))$ =GCD(
- $d = e^{-1} \bmod \Phi(n) =$

- Public Key $= (e, n) =$
- Private Key $= (d, p, q$

mir, and

e prime

d q

y

s

RSA encryption, and decryption

Given a message $M < n$

- **Encryption**. $C = M^e \bmod n$
- **Decryption**. $M = C^d \bmod n$

Example

- $p = 11, \ q = 13$
- $n = p \cdot q = 11 \cdot 13 = 143$
- $\Phi(n) = (p-1)(q-1) = 10 \cdot 12 = 120$
- GCD$(e, \Phi(n))$ =GCD$(e, 120) = 1$; $e =$
- $d = e^{-1} \bmod \Phi(n) = 17^{-1} \bmod 120 =$

- Public Key $= (e, n) = (17, 143)$
- Private Key $= (d, p, q) = (113, 11, 23)$

RSA encryption, and decryption

Given a message $M < n$

- **Encryption**. $C = M^e \bmod n$
- **Decryption**. $M = C^d \bmod n$

Example

- $p = 11,\ q = 13$
- $n = p \cdot q = 11 \cdot 13 = 143$
- $\Phi(n) = (p - 1)(q - 1) = 10 \cdot 12 = 120$
- $\mathrm{GCD}(e, \Phi(n)) = \mathrm{GCD}(e, 120) = 1;\ e = 17$
- $d = e^{-1} \bmod \Phi(n) = 17^{-1} \bmod 120 = 113$

- Public Key $= (e, n) = (17, 143)$
- Private Key $= (d, p, q) = (113, 11, 23)$

ryption, and decryption

- message $M < n$
- **yption**. $C = M^e$ mod n
- **yption**. $M = C^d$ mod n

Example

- $p = 11,\ q = 13$
- $n = p \cdot q = 11 \cdot 13 = 143$
- $\Phi(n) = (p-1)(q-1) = 10 \cdot 12 = 120$
- GCD$(e, \Phi(n))$ =GCD$(e, 120) = 1;\ e = 17$
- $d = e^{-1}$ mod $\Phi(n) = 17^{-1}$ mod $120 = 113$

- Public Key $= (e, n) = (17, 143)$
- Private Key $= (d, p, q) = (113, 11, 23)$

... exam

- Mess
- **Encr**
  $C =$
- **Decr**
  $M =$

seems e
would h

decryption

mod n

mod n

Example

- $p = 11,\ q = 13$
- $n = p \cdot q = 11 \cdot 13 = 143$
- $\Phi(n) = (p-1)(q-1) = 10 \cdot 12 = 120$
- $\text{GCD}(e, \Phi(n)) = \text{GCD}(e, 120) = 1;\ e = 17$
- $d = e^{-1} \bmod \Phi(n) = 17^{-1} \bmod 120 = 113$

- Public Key $= (e, n) = (17, 143)$
- Private Key $= (d, p, q) = (113, 11, 23)$

… example

- Message $M = 50$

- **Encryption:**
  $C = M^e \bmod n = 50^{1}$

- **Decryption:**
  $M = C^d \bmod n = 85^{1}$

seems easy; however, obs
would happen with very

Example

- $p = 11,\ q = 13$
- $n = p \cdot q = 11 \cdot 13 = 143$
- $\Phi(n) = (p-1)(q-1) = 10 \cdot 12 = 120$
- $\mathrm{GCD}(e, \Phi(n)) = \mathrm{GCD}(e, 120) = 1;\ e = 17$
- $d = e^{-1} \bmod \Phi(n) = 17^{-1} \bmod 120 = 113$

- Public Key $= (e, n) = (17, 143)$
- Private Key $= (d, p, q) = (113, 11, 23)$

... example

- Message $M = 50$

- **Encryption:**
  $C = M^e \bmod n = 50^{17} \bmod 143 = 85$

- **Decryption:**
  $M = C^d \bmod n = 85^{113} \bmod 143 = 50$

seems easy; however, observe the 85^{113}, w
would happen with very large numbers?

Example

- $p = 11,\ q = 13$
- $n = p \cdot q = 11 \cdot 13 = 143$
- $\Phi(n) = (p-1)(q-1) = 10 \cdot 12 = 120$
- GCD$(e, \Phi(n))$ =GCD$(e, 120) = 1;\ e = 17$
- $d = e^{-1} \bmod \Phi(n) = 17^{-1} \bmod 120 = 113$

- Public Key $= (e, n) = (17, 143)$
- Private Key $= (d, p, q) = (113, 11, 23)$

... example

- Message $M = 50$
- **Encryption:**
  $C = M^e \bmod n = 50^{17} \bmod 143 = 85$
- **Decryption:**
  $M = C^d \bmod n = 85^{113} \bmod 143 = 50$

seems easy; however, observe the 85^{113}, what would happen with very large numbers?

1, $q = 13$

$\cdot q = 11 \cdot 13 = 143$

$= (p-1)(q-1) = 10 \cdot 12 = 120$

$(e, \Phi(n))$ =GCD$(e, 120) = 1$; $e = 17$

$e^{-1} \bmod \Phi(n) = 17^{-1} \bmod 120 = 113$

c Key $= (e, n) = (17, 143)$

te Key $= (d, p, q) = (113, 11, 23)$

… example

- Message $M = 50$
- **Encryption:**
  $C = M^e \bmod n = 50^{17} \bmod 143 = 85$
- **Decryption:**
  $M = C^d \bmod n = 85^{113} \bmod 143 = 50$

seems easy; however, observe the 85^{113}, what would happen with very large numbers?

DiffieHel

(DHKE)

- The I
  expor
  funct
  comn

143

$) = 10 \cdot 12 = 120$

$e, 120) = 1;\ e = 17$

$17^{-1} \bmod 120 = 113$

$(17, 143)$

$) = (113, 11, 23)$

… example

- Message $M = 50$

- **Encryption:**
  $C = M^e \bmod n = 50^{17} \bmod 143 = 85$

- **Decryption:**
  $M = C^d \bmod n = 85^{113} \bmod 143 = 50$

seems easy; however, observe the 85^{113}, what would happen with very large numbers?

DiffieHellman Key Exc

(DHKE)

- The basic idea behind
  exponentiation in $\mathbb{Z}_p^\star$,
  function, and the expo
  commutative:

$$x \equiv (\alpha^x)^y \equiv$$

17

113

… example

- Message $M = 50$

- **Encryption:**
  $C = M^e \bmod n = 50^{17} \bmod 143 = 85$

- **Decryption:**
  $M = C^d \bmod n = 85^{113} \bmod 143 = 50$

seems easy; however, observe the 85^{113}, what would happen with very large numbers?

DiffieHellman Key Exchange

(DHKE)

- The basic idea behind DHKE is that th
  exponentiation in $\mathbb{Z}_p^\star$, a p-prime, is a o
  function, and the exponentiation is
  commutative:

$$x \equiv (\alpha^x)^y \equiv (\alpha^y)^x \bmod p$$

… example

- Message $M = 50$
- **Encryption:**
  $C = M^e \bmod n = 50^{17} \bmod 143 = 85$
- **Decryption:**
  $M = C^d \bmod n = 85^{113} \bmod 143 = 50$

seems easy; however, observe the 85^{113}, what would happen with very large numbers?

DiffieHellman Key Exchange

(DHKE)

- The basic idea behind DHKE is that the exponentiation in $\mathbb{Z}_p^\star$, a p-prime, is a one-way function, and the exponentiation is commutative:

$$x \equiv (\alpha^x)^y \equiv (\alpha^y)^x \bmod p$$

ple

age $M = 50$

yption:
$M^e \bmod n = 50^{17} \bmod 143 = 85$

yption:
$C^d \bmod n = 85^{113} \bmod 143 = 50$

asy; however, observe the 85^{113}, what
appen with very large numbers?

DiffieHellman Key Exchange

(DHKE)

- The basic idea behind DHKE is that the exponentiation in $\mathbb{Z}_p^\star$, a p-prime, is a one-way function, and the exponentiation is commutative:

$$x \equiv (\alpha^x)^y \equiv (\alpha^y)^x \bmod p$$

DHKE D

a

A

$A_{\text{pub}} \equiv$

$k_{AB} \equiv ($

^{7} mod $143 = 85$

^{13} mod $143 = 50$

erve the 85^{113}, what
arge numbers?

DiffieHellman Key Exchange

(DHKE)

- The basic idea behind DHKE is that the exponentiation in $\mathbb{Z}_p^\star$, a p-prime, is a one-way function, and the exponentiation is commutative:

$$x \equiv (\alpha^x)^y \equiv (\alpha^y)^x \bmod p$$

DHKE Diagram

Alice

$$a \in_R \mathbb{Z}_p^\star$$
$$A_{\text{priv}} = a$$
$$A_{\text{pub}} \equiv \alpha^a \bmod p$$

$$k_{AB} \equiv (B_{pub})^a \bmod p$$

hat

DiffieHellman Key Exchange

(DHKE)

- The basic idea behind DHKE is that the exponentiation in $\mathbb{Z}_p^\star$, a p-prime, is a one-way function, and the exponentiation is commutative:

$$x \equiv (\alpha^x)^y \equiv (\alpha^y)^x \bmod p$$

DHKE Diagram

Alice

Given p y α

$$a \in_R \mathbb{Z}_p^\star$$
$$A_{\text{priv}} = a$$
$$A_{\text{pub}} \equiv \alpha^a \bmod p$$

$$\xrightarrow{A_{\text{pub}}}$$

$$\xleftarrow{B_{\text{pub}}}$$

$$k_{AB} \equiv (B_{pub})^a \bmod p$$

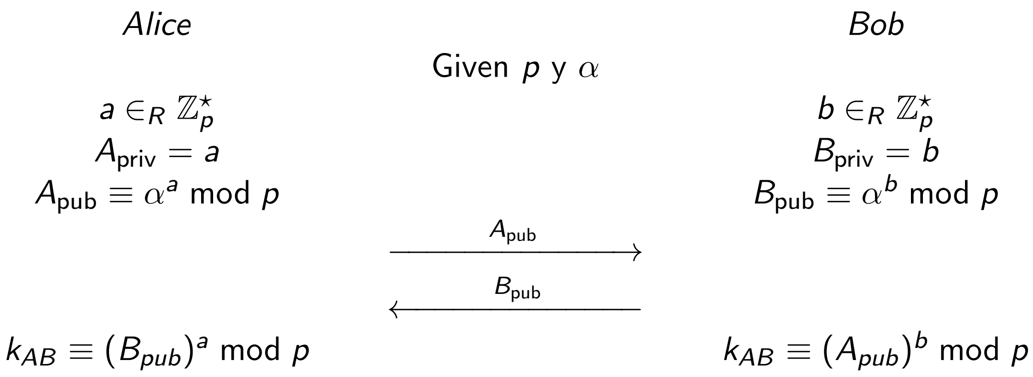
DiffieHellman Key Exchange

(DHKE)

- The basic idea behind DHKE is that the exponentiation in $\mathbb{Z}_p^\star$, a p-prime, is a one-way function, and the exponentiation is commutative:

$$x \equiv (\alpha^x)^y \equiv (\alpha^y)^x \bmod p$$

DHKE Diagram



Alice

Given p y α

Bob

$a \in_R \mathbb{Z}_p^\star$

$A_{\text{priv}} = a$

$A_{\text{pub}} \equiv \alpha^a \bmod p$

$b \in_R \mathbb{Z}_p^\star$

$B_{\text{priv}} = b$

$B_{\text{pub}} \equiv \alpha^b \bmod p$

$\xrightarrow{A_{\text{pub}}}$

$\xleftarrow{B_{\text{pub}}}$

$k_{AB} \equiv (B_{pub})^a \bmod p$

$k_{AB} \equiv (A_{pub})^b \bmod p$

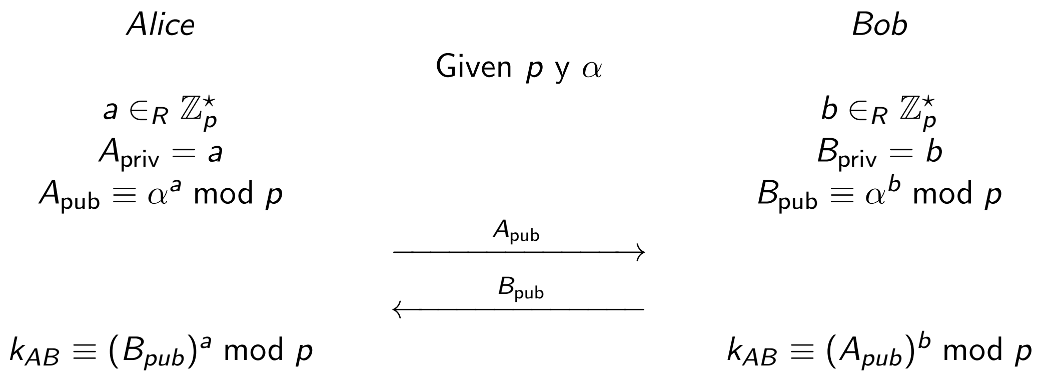
Hellman Key Exchange

)

The basic idea behind DHKE is that the xponentiation in $\mathbb{Z}_p^\star$, a p-prime, is a one-way unction, and the exponentiation is ommutative:

$$x \equiv (\alpha^x)^y \equiv (\alpha^y)^x \bmod p$$

DHKE Diagram



Alice

Given p y α

Bob

$a \in_R \mathbb{Z}_p^\star$
$A_{\text{priv}} = a$
$A_{\text{pub}} \equiv \alpha^a \bmod p$

$b \in_R \mathbb{Z}_p^\star$
$B_{\text{priv}} = b$
$B_{\text{pub}} \equiv \alpha^b \bmod p$

$\xrightarrow{A_{\text{pub}}}$

$\xleftarrow{B_{\text{pub}}}$

$k_{AB} \equiv (B_{pub})^a \bmod p$

$k_{AB} \equiv (A_{pub})^b \bmod p$

DHKE Diagram

$$\begin{array}{ccc}
\textit{Alice} & & \textit{Bob} \\
 & \text{Given } p \text{ y } \alpha & \\
a \in_R \mathbb{Z}_p^\star & & b \in_R \mathbb{Z}_p^\star \\
A_{\text{priv}} = a & & B_{\text{priv}} = b \\
A_{\text{pub}} \equiv \alpha^a \bmod p & & B_{\text{pub}} \equiv \alpha^b \bmod p \\
 & \xrightarrow{\qquad A_{\text{pub}} \qquad} & \\
 & \xleftarrow{\qquad B_{\text{pub}} \qquad} & \\
k_{AB} \equiv (B_{pub})^a \bmod p & & k_{AB} \equiv (A_{pub})^b \bmod p
\end{array}$$

Bob

$$b \in_R \mathbb{Z}_p^\star$$
$$B_{\text{priv}} = b$$
$$B_{\text{pub}} \equiv \alpha^b \bmod p$$

$\longrightarrow$

$—$

$$k_{AB} \equiv (A_{pub})^b \bmod p$$

Digital Signatures

- Demonstrating that certain person generated a message is critical so some applications.
- In the "analog" world, we use hand-written signatures (in some countries any way).
- Only the person who created the signature can reproduce it.

Bob

$$b \in_R \mathbb{Z}_p^\star$$

$$B_{\text{priv}} = b$$

$$\text{pub} \equiv \alpha^b \bmod p$$

$$\equiv (A_{pub})^b \bmod p$$

Digital Signatures

- Demonstrating that certain person generated a message is critical so some applications.
- In the "analog" world, we use hand-written signatures (in some countries any way).
- Only the person who created the signature can reproduce it.

Diagram

In the digital w
key cryptograp
private key, and
public key to ve

Bob

$$b \in_R \mathbb{Z}_p^\star$$

$$B_{\text{priv}} = b$$

$$\text{pub} \equiv \alpha^b \bmod p$$

$$\equiv (A_{pub})^b \bmod p$$

Digital Signatures

- Demonstrating that certain person generated a message is critical so some applications.
- In the “analog” world, we use hand-written signatures (in some countries any way).
- Only the person who created the signature can reproduce it.

Diagram

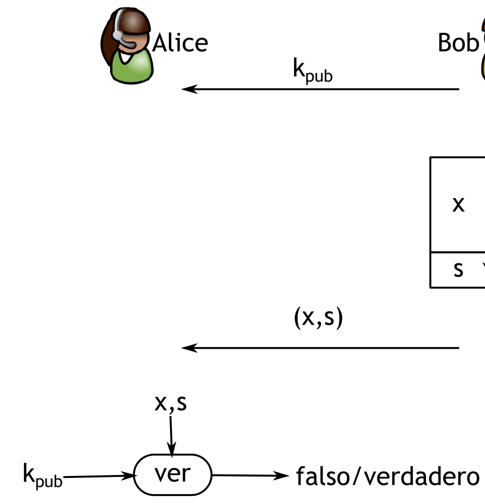
In the digital w
key cryptograp
private key, and
public key to v

Digital Signatures

- Demonstrating that certain person generated a message is critical so some applications.
- In the "analog" world, we use hand-written signatures (in some countries any way).
- Only the person who created the signature can reproduce it.

Diagram

In the digital world this is possib key cryptography. The signatory private key, and addressee uses t public key to verify.

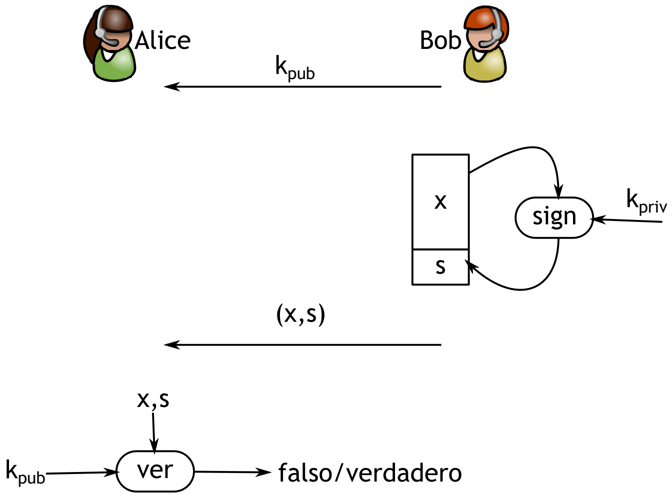


Digital Signatures

- Demonstrating that certain person generated a message is critical so some applications.
- In the "analog" world, we use hand-written signatures (in some countries any way).
- Only the person who created the signature can reproduce it.

Diagram

In the digital world this is possible by using public key cryptography. The signatory signs with her private key, and addressee uses the corresponding public key to verify.



ignatures

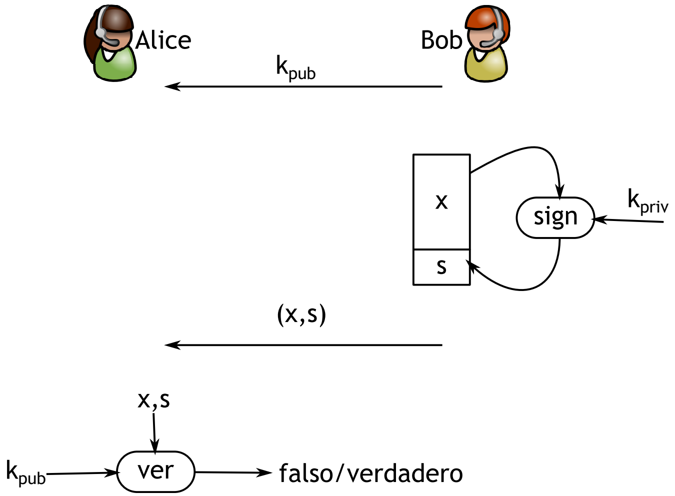
onstrating that certain person generated a age is critical so some applications.

e “analog” world, we use hand-written tures (in some countries any way).

the person who created the signature can duce it.

Diagram

In the digital world this is possible by using public key cryptography. The signatory signs with her private key, and addressee uses the corresponding public key to verify.



General

Basic RSA

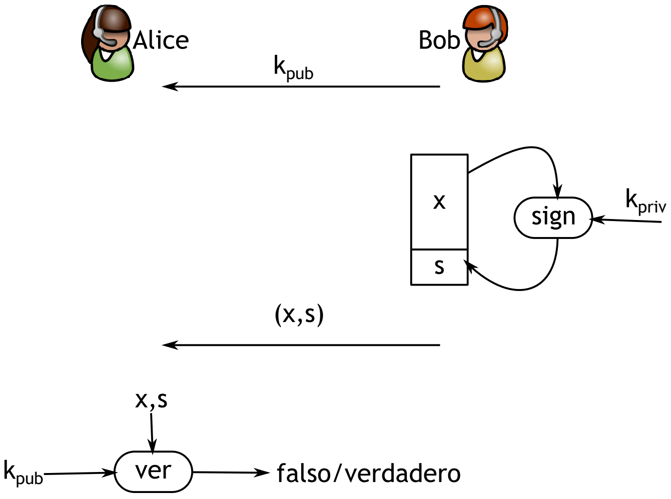
ertain person generated a
some applications.

we use hand-written
untries any way).

created the signature can

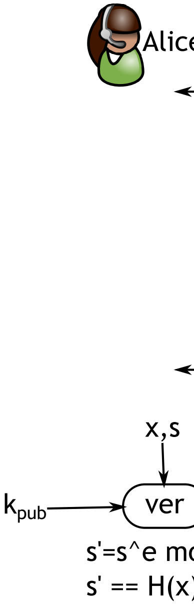
Diagram

In the digital world this is possible by using public key cryptography. The signatory signs with her private key, and addressee uses the corresponding public key to verify.



General diagram for R

Basic RSA signature



erated a

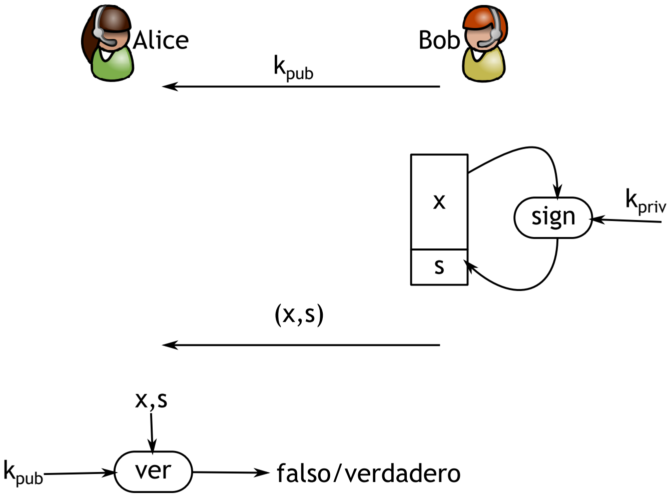
.

tten

ture can

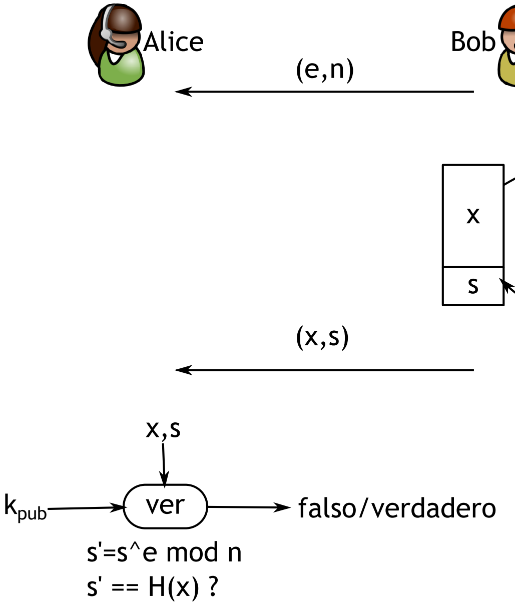
Diagram

In the digital world this is possible by using public key cryptography. The signatory signs with her private key, and addressee uses the corresponding public key to verify.



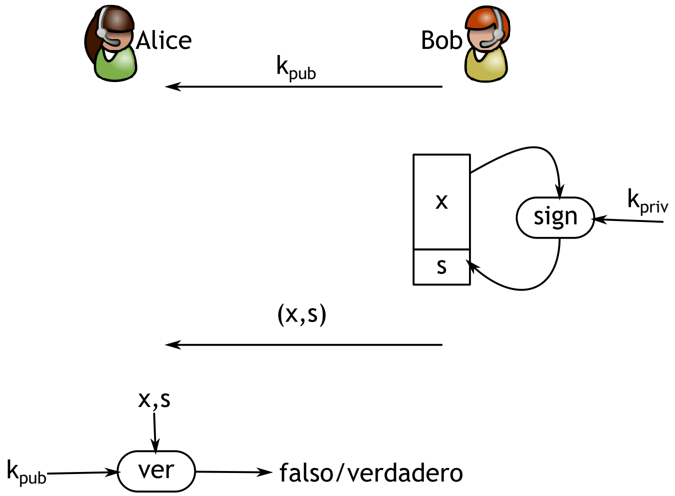
General diagram for RSA signature

Basic RSA signature



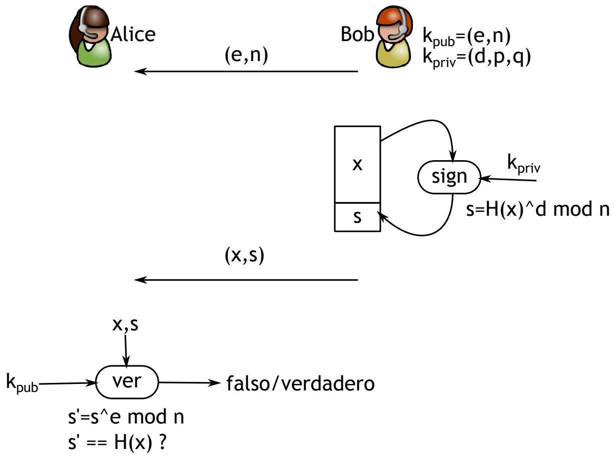
Diagram

In the digital world this is possible by using public key cryptography. The signatory signs with her private key, and addressee uses the corresponding public key to verify.



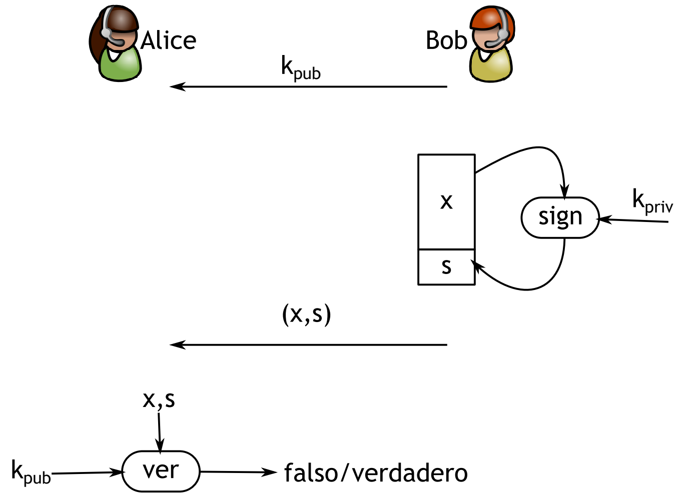
General diagram for RSA signature

Basic RSA signature



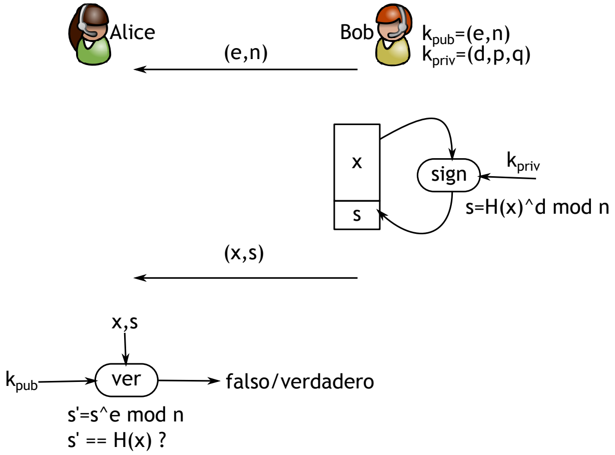
ram

he digital world this is possible by using public cryptography. The signatory signs with her ate key, and addressee uses the corresponding lic key to verify.



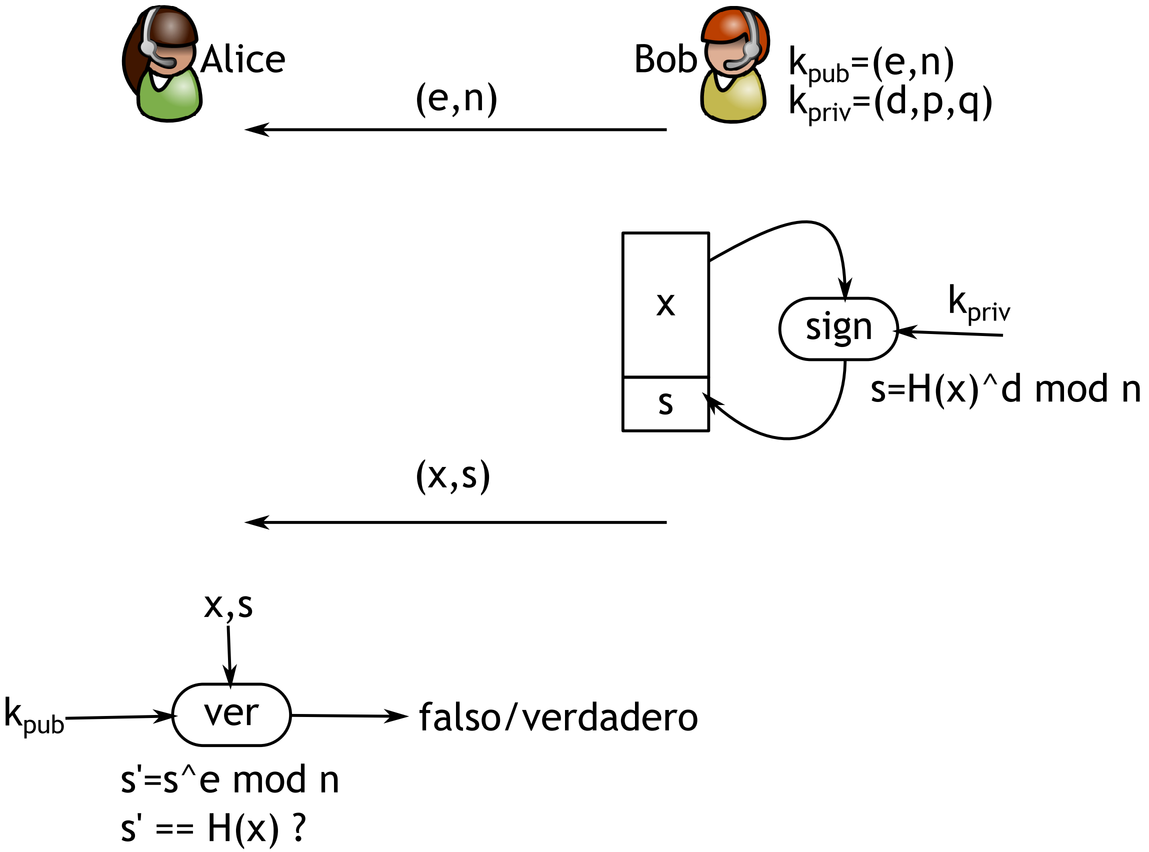
General diagram for RSA signature

Basic RSA signature

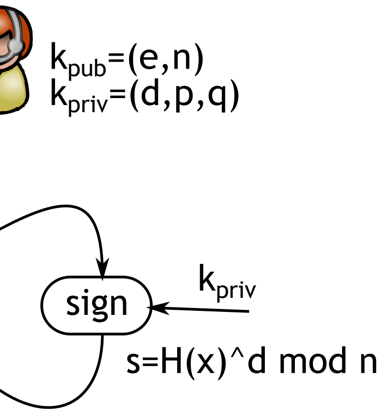


General diagram for RSA signature

Basic RSA signature



e



k_{pub}=(e,n)
k_{priv}=(d,p,q)

sign

k_{priv}

s=H(x)^d mod n

ElGamal

- The Elgamal encryption was proposed by Taher Elgamal in 1985.
- It can be seen as an extension of the Diffie-Hellman Key Exchange (DHKE)

od n

ElGamal

- The Elgamal encryption was proposed by Taher Elgamal in 1985.
- It can be seen as an extension of the Diffie-Hellman Key Exchange (DHKE)

Elgamal signat

- Key generat
  - Prime nun
  - Find an el
  - Choose a r
  - Compute ,
- Message sig
  - Given a m
  - Choose an $0 < k_E <$
  - Compute
  - Compute

od n

ElGamal

- The Elgamal encryption was proposed by Taher Elgamal in 1985.
- It can be seen as an extension of the Diffie-Hellman Key Exchange (DHKE)

Elgamal signa

- Key generat
  - Prime nun
  - Find an el
  - Choose a r
  - Compute
- Message sig
  - Given a m
  - Choose an $0 < k_E <$
  - Compute
  - Compute

ElGamal

- The Elgamal encryption was proposed by Taher Elgamal in 1985.
- It can be seen as an extension of the Diffie-Hellman Key Exchange (DHKE)

Elgamal signature

- Key generation:
  - Prime number generation p
  - Find an element $\alpha \in \mathbb{Z}_p^\star$
  - Choose a random element d,
  - Compute $\beta = \alpha^d \bmod p$
- Message signing:
  - Given a message M
  - Choose an ephemeral key k_E $0 < k_E < p - 2$, with GCD(
  - Compute $r \equiv a^{k_E} \bmod p$
  - Compute $s \equiv (M - d \cdot r)k_E^{-1}$

ElGamal

- The Elgamal encryption was proposed by Taher Elgamal in 1985.
- It can be seen as an extension of the Diffie-Hellman Key Exchange (DHKE)

Elgamal signature

- Key generation:
  - Prime number generation p
  - Find an element $\alpha \in \mathbb{Z}_p^\star$
  - Choose a random element d, with $2 < d < p - 2$
  - Compute $\beta = \alpha^d \bmod p$
- Message signing:
  - Given a message M
  - Choose an ephemeral key k_E, with $0 < k_E < p - 2$, with $\text{GCD}(k_E, p - 1) = 1$
  - Compute $r \equiv a^{k_E} \bmod p$
  - Compute $s \equiv (M - d \cdot r)k_E^{-1} \bmod p - 1$

ElGamal

- The Elgamal encryption was proposed by Taher Elgamal in 1985.
- It can be seen as an extension of the Diffie-Hellman Key Exchange (DHKE)

Elgamal signature

- Key generation:
  - Prime number generation p
  - Find an element $\alpha \in \mathbb{Z}_p^\star$
  - Choose a random element d, with $2 < d < p - 2$
  - Compute $\beta = \alpha^d \bmod p$
- Message signing:
  - Given a message M
  - Choose an ephemeral key k_E, with $0 < k_E < p - 2$, with $\mathrm{GCD}(k_E, p - 1) = 1$
  - Compute $r \equiv a^{k_E} \bmod p$
  - Compute $s \equiv (M - d \cdot r)k_E^{-1} \bmod p - 1$
- The signature of M is (r, s)

signature

generation:
- ime number generation p
- nd an element $\alpha \in \mathbb{Z}_p^\star$
- oose a random element d, with $2 < d < p - 2$
- mpute $\beta = \alpha^d \bmod p$

age signing:
- ven a message M
- oose an ephemeral key k_E, with $< k_E < p - 2$, with $\mathrm{GCD}(k_E, p-1) = 1$
- mpute $r \equiv a^{k_E} \bmod p$
- mpute $s \equiv (M - d \cdot r)k_E^{-1} \bmod p - 1$

signature of M is (r, s)

Elgamal signature

- Key generation:
  - Prime number generation p
  - Find an element $\alpha \in \mathbb{Z}_p^\star$
  - Choose a random element d, with $2 < d < p - 2$
  - Compute $\beta = \alpha^d \bmod p$
- Message signing:
  - Given a message M
  - Choose an ephemeral key k_E, with $0 < k_E < p - 2$, with $\mathrm{GCD}(k_E, p-1) = 1$
  - Compute $r \equiv a^{k_E} \bmod p$
  - Compute $s \equiv (M - d \cdot r)k_E^{-1} \bmod p - 1$
- The signature of M is (r, s)

Elgamal

- Signa
  - Co
- If $t \equiv$

tion p

$\mathbb{Z}_p^\star$

ment d, with $2 < d < p-2$

d p

key k_E, with

GCD$(k_E, p-1) = 1$

d p

$\cdot r)k_E^{-1}$ mod $p-1$

(r, s)

Elgamal signature

- Key generation:
  - Prime number generation p
  - Find an element $\alpha \in \mathbb{Z}_p^\star$
  - Choose a random element d, with $2 < d < p-2$
  - Compute $\beta = \alpha^d$ mod p

- Message signing:
  - Given a message M
  - Choose an ephemeral key k_E, with $0 < k_E < p-2$, with GCD$(k_E, p-1) = 1$
  - Compute $r \equiv a^{k_E}$ mod p
  - Compute $s \equiv (M - d \cdot r)k_E^{-1}$ mod $p-1$

- The signature of M is (r, s)

Elgamal Signature Ve

- Signature Verification:
  - Compute $t \equiv \beta^r \cdot r^s$

- If $t \equiv \alpha^x$ mod pq, the

$< p - 2$

$= 1$

1

Elgamal signature

- Key generation:
  - Prime number generation p
  - Find an element $\alpha \in \mathbb{Z}_p^\star$
  - Choose a random element d, with $2 < d < p - 2$
  - Compute $\beta = \alpha^d \bmod p$
- Message signing:
  - Given a message M
  - Choose an ephemeral key k_E, with $0 < k_E < p - 2$, with $\mathrm{GCD}(k_E, p-1) = 1$
  - Compute $r \equiv a^{k_E} \bmod p$
  - Compute $s \equiv (M - d \cdot r) k_E^{-1} \bmod p - 1$
- The signature of M is (r, s)

Elgamal Signature Verification

- Signature Verification:
  - Compute $t \equiv \beta^r \cdot r^s \bmod p$
- If $t \equiv \alpha^x \bmod pq$, the signature verifies

Elgamal signature

- Key generation:
  - Prime number generation p
  - Find an element $\alpha \in \mathbb{Z}_p^\star$
  - Choose a random element d, with $2 < d < p - 2$
  - Compute $\beta = \alpha^d \bmod p$

- Message signing:
  - Given a message M
  - Choose an ephemeral key k_E, with $0 < k_E < p - 2$, with $\mathrm{GCD}(k_E, p - 1) = 1$
  - Compute $r \equiv a^{k_E} \bmod p$
  - Compute $s \equiv (M - d \cdot r)k_E^{-1} \bmod p - 1$

- The signature of M is (r, s)

Elgamal Signature Verification

- Signature Verification:
  - Compute $t \equiv \beta^r \cdot r^s \bmod p$

- If $t \equiv \alpha^x \bmod pq$, the signature verifies.

signature

generation:

- me number generation p
- d an element $\alpha \in \mathbb{Z}_p^\star$
- oose a random element d, with $2 < d < p - 2$
- mpute $\beta = \alpha^d \bmod p$

age signing:

- ven a message M
- oose an ephemeral key k_E, with $< k_E < p - 2$, with $\mathrm{GCD}(k_E, p - 1) = 1$
- mpute $r \equiv a^{k_E} \bmod p$
- mpute $s \equiv (M - d \cdot r)k_E^{-1} \bmod p - 1$

signature of M is (r, s)

Elgamal Signature Verification

- Signature Verification:
  - Compute $t \equiv \beta^r \cdot r^s \bmod p$

- If $t \equiv \alpha^x \bmod pq$, the signature verifies.

Example

$t = 7^3 \cdot 3$

$\alpha^x \equiv 2^2$

$t \equiv \alpha^x$

tion p

$\mathbb{Z}_p^\star$

ment d, with $2 < d < p - 2$

d p

key k_E, with

$\text{GCD}(k_E, p-1) = 1$

d p

$\cdot\, r) k_E^{-1} \bmod p - 1$

(r, s)

Elgamal Signature Verification

- Signature Verification:
  - Compute $t \equiv \beta^r \cdot r^s \bmod p$
- If $t \equiv \alpha^x \bmod pq$, the signature verifies.

Example, sign M

$\leftarrow$

$\leftarrow$

$$t = 7^3 \cdot 3^{26} \equiv 22$$
$$\alpha^x \equiv 2^{26} \equiv 22$$
$$t \equiv \alpha^x \Rightarrow \text{OK}$$

$/ < p - 2$

$= 1$

1

Elgamal Signature Verification

- Signature Verification:
  - Compute $t \equiv \beta^r \cdot r^s \bmod p$
- If $t \equiv \alpha^x \bmod pq$, the signature verifies.

Example, sign M

$\xleftarrow{k_{\text{pub}}(p,\alpha,\beta)=(29,2,}$

$\xleftarrow{(26,(3,26))}$

$$t = 7^3 \cdot 3^{26} \equiv 22$$
$$\alpha^x \equiv 2^{26} \equiv 22$$
$$t \equiv \alpha^x \Rightarrow \text{OK}$$

Elgamal Signature Verification

- Signature Verification:
  - Compute $t \equiv \beta^r \cdot r^s \bmod p$

- If $t \equiv \alpha^x \bmod pq$, the signature verifies.

Example, sign M

$$p = 29,\ \alpha = 2$$
$$d = 12$$
$$\beta = \alpha^d \equiv 7$$

$$\xleftarrow{k_{\text{pub}}(p,\alpha,\beta)=(29,2,7)}$$

$$k_E = 5$$
$$(5, 28) = 1$$
$$x = 26$$
$$r = 2^5 \equiv 3$$
$$s = -10 \cdot 7 \equiv 26$$

$$\xleftarrow{(26,(3,26))}$$

$$t = 7^3 \cdot 3^{26} \equiv 22$$
$$\alpha^x \equiv 2^{26} \equiv 22$$
$$t \equiv \alpha^x \Rightarrow \text{OK}$$

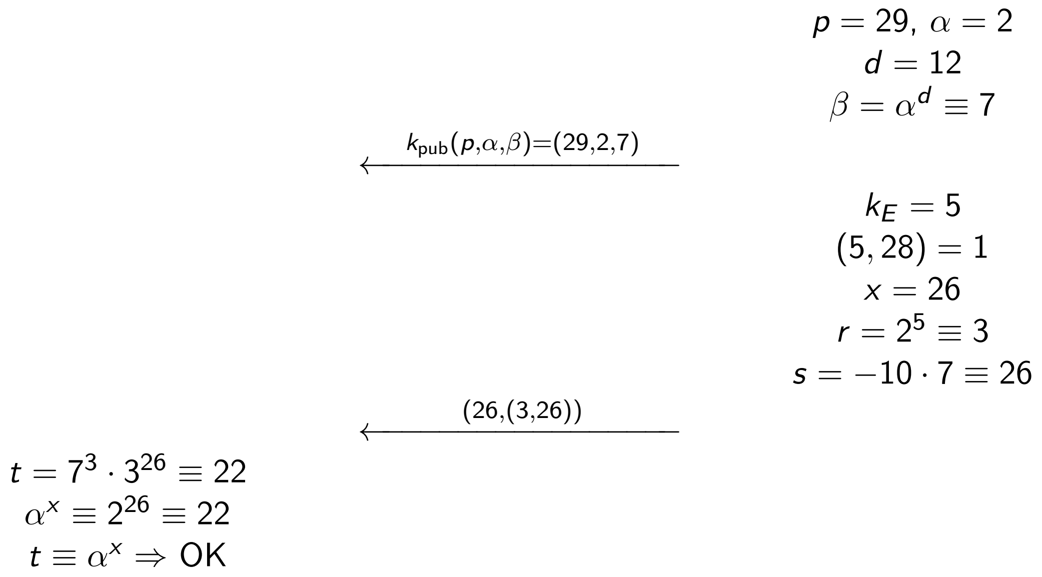
mal Signature Verification

ignature Verification:

- Compute $t \equiv \beta^r \cdot r^s \bmod p$

f $t \equiv \alpha^x \bmod pq$, the signature verifies.

Example, sign M



$p = 29,\ \alpha = 2$

$d = 12$

$\beta = \alpha^d \equiv 7$

$k_{\text{pub}}(p,\alpha,\beta)=(29,2,7)$

$k_E = 5$

$(5, 28) = 1$

$x = 26$

$r = 2^5 \equiv 3$

$s = -10 \cdot 7 \equiv 26$

(26,(3,26))

$t = 7^3 \cdot 3^{26} \equiv 22$

$\alpha^x \equiv 2^{26} \equiv 22$

$t \equiv \alpha^x \Rightarrow \text{OK}$

Example, sign M

$$p = 29,\ \alpha = 2$$
$$d = 12$$
$$\beta = \alpha^d \equiv 7$$

$$\xleftarrow{\qquad k_{\text{pub}}(p,\alpha,\beta)=(29,2,7) \qquad}$$

$$k_E = 5$$
$$(5, 28) = 1$$
$$x = 26$$
$$r = 2^5 \equiv 3$$
$$s = -10 \cdot 7 \equiv 26$$

$$\xleftarrow{\qquad (26,(3,26)) \qquad}$$

$$t = 7^3 \cdot 3^{26} \equiv 22$$
$$\alpha^x \equiv 2^{26} \equiv 22$$
$$t \equiv \alpha^x \Rightarrow \text{OK}$$

$$p = 29,\ \alpha = 2$$
$$d = 12$$
$$\beta = \alpha^d \equiv 7$$

7)

$$k_E = 5$$
$$(5, 28) = 1$$
$$x = 26$$
$$r = 2^5 \equiv 3$$
$$s = -10 \cdot 7 \equiv 26$$

DSA Signature

The standard signature DSA has the following steps:

- Key Generation:
  - Find Prime number p, with $2^{1023} < p < 2^{1024}$
  - Find a prime number q: $2^{159} < q < 2^{160}$
  - Find an element α, of order q
  - Choose a random number d, with $1 < d < q$
  - Compute $\beta = \alpha^d \bmod p$
- The key are:
  - Public: (p, q, α, β)
  - Private: d

$p = 29,\ \alpha = 2$

$d = 12$

$\beta = \alpha^d \equiv 7$

$k_E = 5$

$(5, 28) = 1$

$x = 26$

$r = 2^5 \equiv 3$

$s = -10 \cdot 7 \equiv 26$

DSA Signature

The standard signature DSA has the following steps:

- Key Generation:
  - Find Prime number p, with $2^{1023} < p < 2^{1024}$
  - Find a prime number q: $2^{159} < q < 2^{160}$
  - Find an element α, of order q
  - Choose a random number d, with $1 < d < q$
  - Compute $\beta = \alpha^d \bmod p$
- The key are:
  - Public: (p, q, α, β)
  - Private: d

DSA signature

- Message sig
  - Given a m
  - Choose an
  - Compute
  - Compute
- The signatu

$p = 29,\ \alpha = 2$

$d = 12$

$\beta = \alpha^d \equiv 7$

$k_E = 5$

$(5, 28) = 1$

$x = 26$

$r = 2^5 \equiv 3$

$s = -10 \cdot 7 \equiv 26$

DSA Signature

The standard signature DSA has the following steps:

- Key Generation:
  - Find Prime number p, with $2^{1023} < p < 2^{1024}$
  - Find a prime number q: $2^{159} < q < 2^{160}$
  - Find an element α, of order q
  - Choose a random number d, with $1 < d < q$
  - Compute $\beta = \alpha^d \bmod p$
- The key are:
  - Public: (p, q, α, β)
  - Private: d

DSA signature

- Message sig
  - Given a m
  - Choose an
  - Compute
  - Compute
- The signatu

DSA Signature

The standard signature DSA has the following steps:

- Key Generation:
  - Find Prime number p, with $2^{1023} < p < 2^{1024}$
  - Find a prime number q: $2^{159} < q < 2^{160}$
  - Find an element α, of order q
  - Choose a random number d, with $1 < d < q$
  - Compute $\beta = \alpha^d \bmod p$
- The key are:
  - Public: (p, q, α, β)
  - Private: d

DSA signature of a messag

- Message signature:
  - Given a message M
  - Choose an ephemeral key k_E
  - Compute $r \equiv (a^{k_E} \bmod p)$ m
  - Compute $s \equiv (SHA(M) + d$
- The signature of M is (r, s)

DSA Signature

The standard signature DSA has the following steps:

- Key Generation:
  - Find Prime number p, with $2^{1023} < p < 2^{1024}$
  - Find a prime number q: $2^{159} < q < 2^{160}$
  - Find an element α, of order q
  - Choose a random number d, with $1 < d < q$
  - Compute $\beta = \alpha^d \bmod p$
- The key are:
  - Public: (p, q, α, β)
  - Private: d

DSA signature of a message

- Message signature:
  - Given a message M
  - Choose an ephemeral key k_E, with $0 < k_E < q$
  - Compute $r \equiv (a^{k_E} \bmod p) \bmod q$
  - Compute $s \equiv (SHA(M) + d \cdot r)k_E^{-1} \bmod q$
- The signature of M is (r, s)

nature

ndard signature DSA has the following

Generation:

- nd Prime number p, with $2^{1023} < p < 2^{1024}$
- nd a prime number q: $2^{159} < q < 2^{160}$
- nd an element α, of order q
- oose a random number d, with $1 < d < q$
- mpute $\beta = \alpha^d \bmod p$

key are:

- blic: (p, q, α, β)
- vate: d

DSA signature of a message

- Message signature:
  - Given a message M
  - Choose an ephemeral key k_E, with $0 < k_E < q$
  - Compute $r \equiv (a^{k_E} \bmod p) \bmod q$
  - Compute $s \equiv (SHA(M) + d \cdot r)k_E^{-1} \bmod q$
- The signature of M is (r, s)

DSA sign

- Signa
  - Co
  - Co
  - Co
  - Co
- If $v \equiv$

SA has the following

, with $2^{1023} < p < 2^{1024}$

q: $2^{159} < q < 2^{160}$

f order q

mber d, with $1 < d < q$

d p

DSA signature of a message

- Message signature:
  - Given a message M
  - Choose an ephemeral key k_E, with $0 < k_E < q$
  - Compute $r \equiv (a^{k_E} \bmod p) \bmod q$
  - Compute $s \equiv (SHA(M) + d \cdot r)k_E^{-1} \bmod q$
- The signature of M is (r, s)

DSA signature verifica

- Signature verification:
  - Compute $w \equiv s^{-1}$ m
  - Compute $u_1 \equiv w \cdot SH$
  - Compute $u_2 \equiv w \cdot r$ r
  - Compute $v \equiv (\alpha^{u_1} \cdot \beta$
- If $v \equiv r \bmod q$, the si

wing

2^{1024}

0

$d < q$

DSA signature of a message

- Message signature:
  - Given a message M
  - Choose an ephemeral key k_E, with $0 < k_E < q$
  - Compute $r \equiv (a^{k_E} \bmod p) \bmod q$
  - Compute $s \equiv (SHA(M) + d \cdot r)k_E^{-1} \bmod q$
- The signature of M is (r, s)

DSA signature verification

- Signature verification:
  - Compute $w \equiv s^{-1} \bmod q$
  - Compute $u_1 \equiv w \cdot SHA(M) \bmod q$
  - Compute $u_2 \equiv w \cdot r \bmod q$
  - Compute $v \equiv (\alpha^{u_1} \cdot \beta^{u_2} \bmod p) \bmod q$
- If $v \equiv r \bmod q$, the signature verifies

DSA signature of a message

- Message signature:
  - Given a message M
  - Choose an ephemeral key k_E, with $0 < k_E < q$
  - Compute $r \equiv (a^{k_E} \bmod p) \bmod q$
  - Compute $s \equiv (SHA(M) + d \cdot r)k_E^{-1} \bmod q$
- The signature of M is (r, s)

DSA signature verification

- Signature verification:
  - Compute $w \equiv s^{-1} \bmod q$
  - Compute $u_1 \equiv w \cdot SHA(M) \bmod q$
  - Compute $u_2 \equiv w \cdot r \bmod q$
  - Compute $v \equiv (\alpha^{u_1} \cdot \beta^{u_2} \bmod p) \bmod q$
- If $v \equiv r \bmod q$, the signature verifies

nature of a message

- age signature:
  - ven a message M
  - oose an ephemeral key k_E, with $0 < k_E < q$
  - mpute $r \equiv (a^{k_E} \bmod p) \bmod q$
  - mpute $s \equiv (SHA(M) + d \cdot r)k_E^{-1} \bmod q$

- signature of M is (r, s)

DSA signature verification

- Signature verification:
  - Compute $w \equiv s^{-1} \bmod q$
  - Compute $u_1 \equiv w \cdot SHA(M) \bmod q$
  - Compute $u_2 \equiv w \cdot r \bmod q$
  - Compute $v \equiv (\alpha^{u_1} \cdot \beta^{u_2} \bmod p) \bmod q$

- If $v \equiv r \bmod q$, the signature verifies

Example

$$
\begin{aligned}
w &= 5 \\
u_1 &= 6 \cdot \\
u_2 &= 6 \\
v = 20 &\equiv
\end{aligned}
$$

$$v \equiv r$$

essage

key k_E, with $0 < k_E < q$

od p) mod q

$M) + d \cdot r)k_E^{-1}$ mod q

(r, s)

DSA signature verification

- Signature verification:
  - Compute $w \equiv s^{-1}$ mod q
  - Compute $u_1 \equiv w \cdot SHA(M)$ mod q
  - Compute $u_2 \equiv w \cdot r$ mod q
  - Compute $v \equiv (\alpha^{u_1} \cdot \beta^{u_2}$ mod $p)$ mod q
- If $v \equiv r$ mod q, the signature verifies

Example, sign messag

$\leftarrow$

$\leftarrow$

$$w = 5^{-1} \equiv 6 \bmod 29$$
$$u_1 = 6 \cdot 26 \equiv 11 \bmod 29$$
$$u_2 = 6 \cdot 20 \equiv 4 \bmod 29$$
$$v = 20 \equiv (3^{11} \cdot 4^4 \bmod 59$$
$$\bmod 29$$
$$v \equiv r \bmod 29 \Rightarrow \text{OK}$$

$k_E < q$

d q

DSA signature verification

- Signature verification:
  - Compute $w \equiv s^{-1} \bmod q$
  - Compute $u_1 \equiv w \cdot SHA(M) \bmod q$
  - Compute $u_2 \equiv w \cdot r \bmod q$
  - Compute $v \equiv (\alpha^{u_1} \cdot \beta^{u_2} \bmod p) \bmod q$
- If $v \equiv r \bmod q$, the signature verifies

Example, sign message M

$\xleftarrow{k_{\text{pub}}(p,q,\alpha,\beta)=(59,29}$

$\xleftarrow{(M,(r,s))}$

$$w = 5^{-1} \equiv 6 \bmod 29$$
$$u_1 = 6 \cdot 26 \equiv 11 \bmod 29$$
$$u_2 = 6 \cdot 20 \equiv 4 \bmod 29$$
$$v = 20 \equiv (3^{11} \cdot 4^4 \bmod 59)$$
$$\bmod 29$$
$$v \equiv r \bmod 29 \Rightarrow \text{OK}$$

DSA signature verification

- Signature verification:
  - Compute $w \equiv s^{-1} \bmod q$
  - Compute $u_1 \equiv w \cdot SHA(M) \bmod q$
  - Compute $u_2 \equiv w \cdot r \bmod q$
  - Compute $v \equiv (\alpha^{u_1} \cdot \beta^{u_2} \bmod p) \bmod q$

- If $v \equiv r \bmod q$, the signature verifies

Example, sign message M

$$p = 59,\ q = 29$$
$$\alpha = 3,\ d = 7$$
$$\beta = \alpha^d \equiv 4$$

$$\xleftarrow{k_{\text{pub}}(p,q,\alpha,\beta)=(59,29,3,4)}$$

$$k_E = 10$$
$$r = (3^{10} \bmod 59)$$
$$\equiv 20 \bmod 29$$
$$s = (26 + 7 \cdot 20) \cdot 3$$
$$\equiv 5 \bmod 29$$

$$\xleftarrow{(M,(r,s))}$$

$$w = 5^{-1} \equiv 6 \bmod 29$$
$$u_1 = 6 \cdot 26 \equiv 11 \bmod 29$$
$$u_2 = 6 \cdot 20 \equiv 4 \bmod 29$$
$$v = 20 \equiv (3^{11} \cdot 4^4 \bmod 59)$$
$$\bmod 29$$
$$v \equiv r \bmod 29 \Rightarrow \text{OK}$$

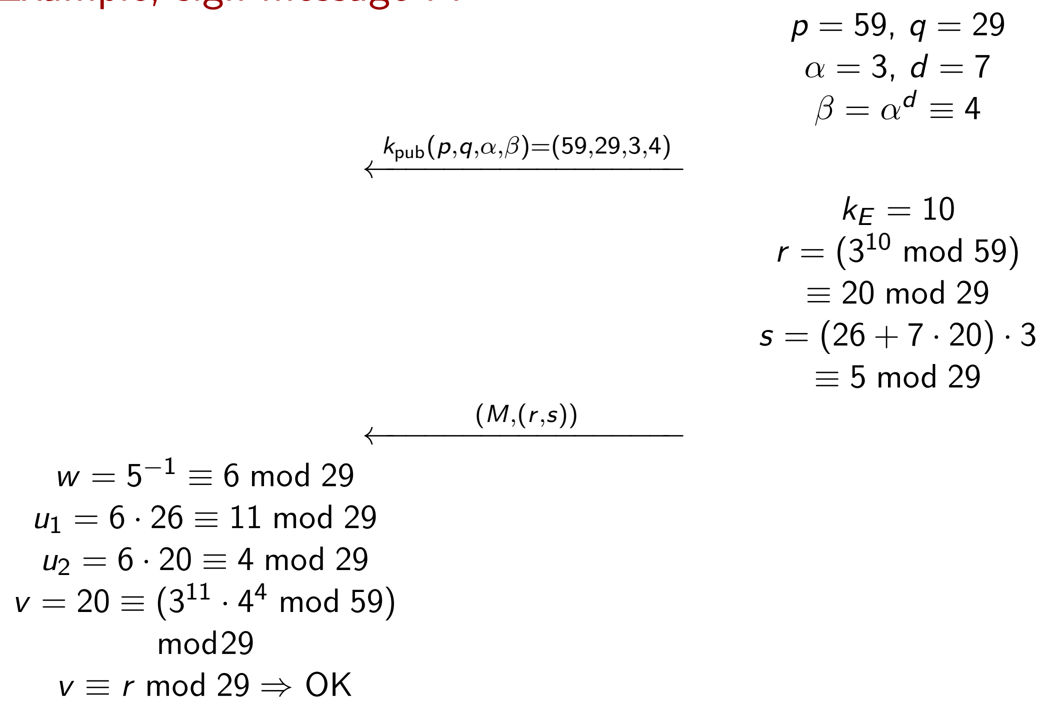
signature verification

ignature verification:

- Compute $w \equiv s^{-1} \bmod q$
- Compute $u_1 \equiv w \cdot SHA(M) \bmod q$
- Compute $u_2 \equiv w \cdot r \bmod q$
- Compute $v \equiv (\alpha^{u_1} \cdot \beta^{u_2} \bmod p) \bmod q$

f $v \equiv r \bmod q$, the signature verifies

Example, sign message M



$p = 59,\ q = 29$

$\alpha = 3,\ d = 7$

$\beta = \alpha^d \equiv 4$

$\xleftarrow{k_{\text{pub}}(p,q,\alpha,\beta)=(59,29,3,4)}$

$k_E = 10$

$r = (3^{10} \bmod 59)$

$\equiv 20 \bmod 29$

$s = (26 + 7 \cdot 20) \cdot 3$

$\equiv 5 \bmod 29$

$\xleftarrow{(M,(r,s))}$

$w = 5^{-1} \equiv 6 \bmod 29$

$u_1 = 6 \cdot 26 \equiv 11 \bmod 29$

$u_2 = 6 \cdot 20 \equiv 4 \bmod 29$

$v = 20 \equiv (3^{11} \cdot 4^4 \bmod 59)$

$\bmod 29$

$v \equiv r \bmod 29 \Rightarrow$ OK

Example, sign message M

$$p = 59,\ q = 29$$
$$\alpha = 3,\ d = 7$$
$$\beta = \alpha^d \equiv 4$$

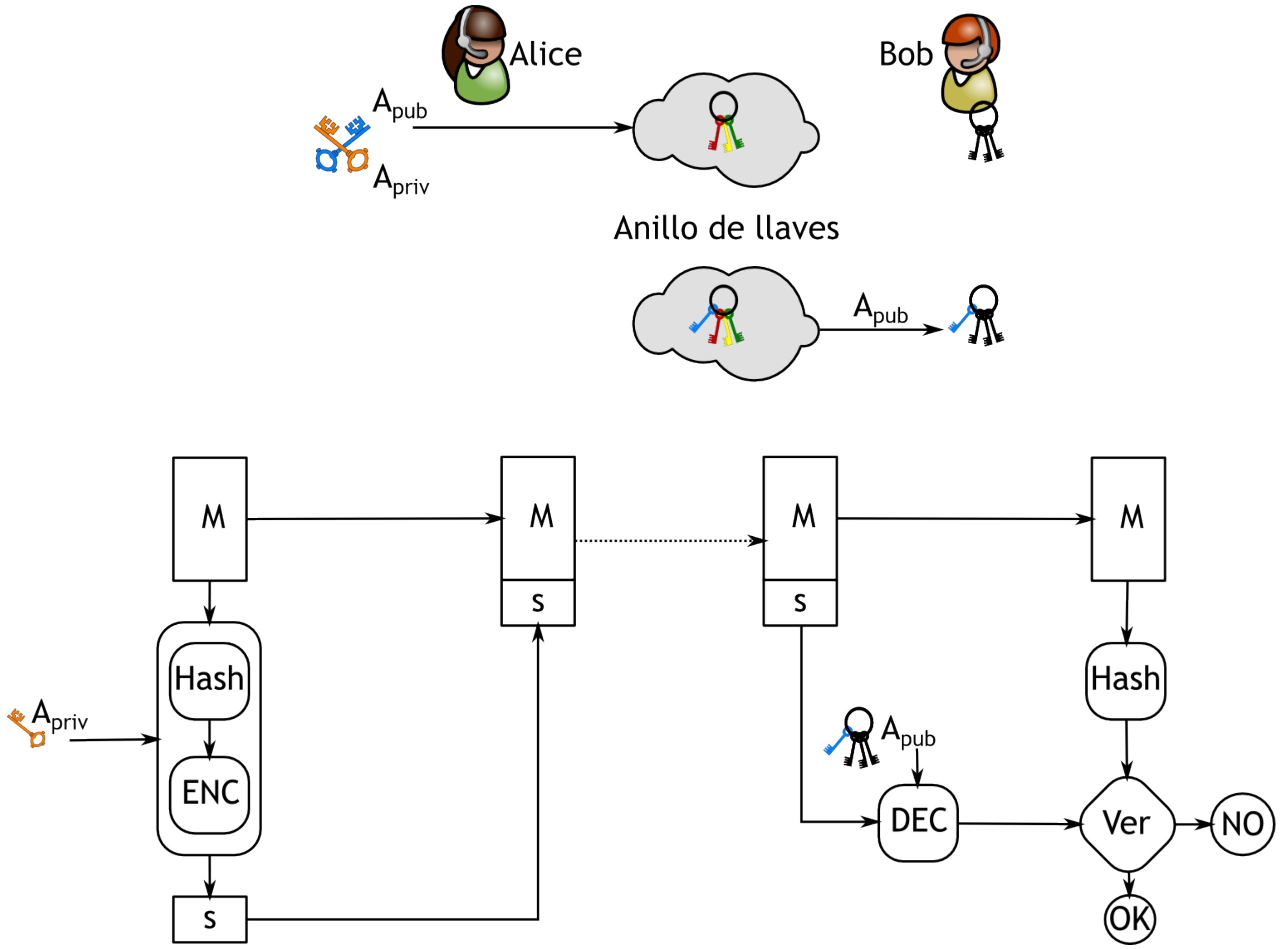
$$\xleftarrow{k_{\text{pub}}(p,q,\alpha,\beta)=(59,29,3,4)}$$

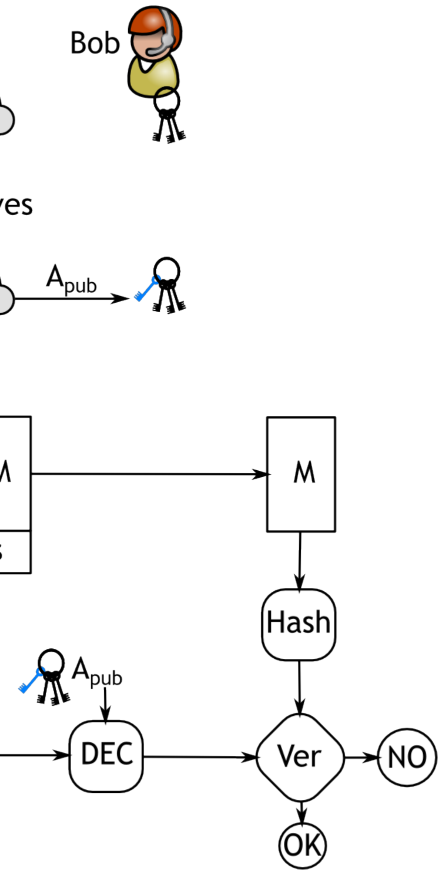
$$k_E = 10$$
$$r = (3^{10} \bmod 59)$$
$$\equiv 20 \bmod 29$$
$$s = (26 + 7 \cdot 20) \cdot 3$$
$$\equiv 5 \bmod 29$$

$$\xleftarrow{(M,(r,s))}$$

$$w = 5^{-1} \equiv 6 \bmod 29$$
$$u_1 = 6 \cdot 26 \equiv 11 \bmod 29$$
$$u_2 = 6 \cdot 20 \equiv 4 \bmod 29$$
$$v = 20 \equiv (3^{11} \cdot 4^4 \bmod 59)$$
$$\bmod 29$$
$$v \equiv r \bmod 29 \Rightarrow \text{OK}$$

Digital Signature



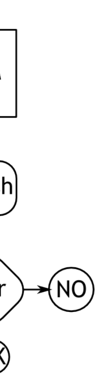


Digital certificate

Is a document in which the digital signature of a trustworthy entity, whose public key is previously stored, associates the public key to a given entity: name, organization, address, email, RFC, CURP, etc.

The certificate serves to warranty that a given public key belongs to the owner of its corresponding private key.

These certificate are granted by a trustworhty entity, a Certificate Authority.. perhaps, in practice, we delegate who to trust to Mozilla, Microsoft, Apple, BlackBerry.



Digital certificate

Is a document in which the digital signature of a trustworthy entity, whose public key is previously stored, associates the public key to a given entity: name, organization, address, email, RFC, CURP, etc.

The certificate serves to warranty that a given public key belongs to the owner of its corresponding private key.

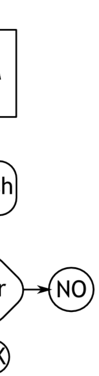
These certificate are granted by a trustworhty entity, a Certificate Authority.. perhaps, in practice, we delegate who to trust to Mozilla, Microsoft, Apple, BlackBerry.

Digital certific

Is a document
trustworthy ent
stored, associat
name, organiza
etc.

The certificate
public key belo
corresponding

These certificat
entity, a Certifi
practice, we de
Microsoft, App



Digital certificate

Is a document in which the digital signature of a trustworthy entity, whose public key is previously stored, associates the public key to a given entity: name, organization, address, email, RFC, CURP, etc.

The certificate serves to warranty that a given public key belongs to the owner of its corresponding private key.

These certificate are granted by a trustworhty entity, a Certificate Authority.. perhaps, in practice, we delegate who to trust to Mozilla, Microsoft, Apple, BlackBerry.

Digital certific

Is a document
trustworthy ent
stored, associat
name, organiza
etc.

The certificate
public key belo
corresponding p

These certificat
entity, a Certifi
practice, we de
Microsoft, App

Digital certificate

Is a document in which the digital signature of a trustworthy entity, whose public key is previously stored, associates the public key to a given entity: name, organization, address, email, RFC, CURP, etc.

The certificate serves to warranty that a given public key belongs to the owner of its corresponding private key.

These certificate are granted by a trustworhty entity, a Certificate Authority.. perhaps, in practice, we delegate who to trust to Mozilla, Microsoft, Apple, BlackBerry.

Digital certificate

Is a document in which the digit
trustworthy entity, whose public
stored, associates the public key
name, organization, address, em
etc.

The certificate serves to warrant
public key belongs to the owner
corresponding private key.

These certificate are granted by
entity, a Certificate Authority..
practice, we delegate who to tru
Microsoft, Apple, BlackBerry.

Digital certificate

Is a document in which the digital signature of a trustworthy entity, whose public key is previously stored, associates the public key to a given entity: name, organization, address, email, RFC, CURP, etc.

The certificate serves to warranty that a given public key belongs to the owner of its corresponding private key.

These certificate are granted by a trustworhty entity, a Certificate Authority.. perhaps, in practice, we delegate who to trust to Mozilla, Microsoft, Apple, BlackBerry.

Digital certificate

Is a document in which the digital signature of a trustworthy entity, whose public key is previously stored, associates the public key to a given entity: name, organization, address, email, RFC, CURP, etc.

The certificate serves to warranty that a given public key belongs to the owner of its corresponding private key.

These certificate are granted by a trustworhty entity, a Certificate Authority.. perhaps, in practice, we delegate who to trust to Mozilla, Microsoft, Apple, BlackBerry.

ertificate

ument in which the digital signature of a
thy entity, whose public key is previously
ssociates the public key to a given entity:
rganization, address, email, RFC, CURP,

tificate serves to warranty that a given
ey belongs to the owner of its
nding private key.

ertificate are granted by a trustworhty
Certificate Authority.. perhaps, in
we delegate who to trust to Mozilla,
ft, Apple, BlackBerry.

Digital certificate

Is a document in which the digital signature of a trustworthy entity, whose public key is previously stored, associates the public key to a given entity: name, organization, address, email, RFC, CURP, etc.

The certificate serves to warranty that a given public key belongs to the owner of its corresponding private key.

These certificate are granted by a trustworhty entity, a Certificate Authority.. perhaps, in practice, we delegate who to trust to Mozilla, Microsoft, Apple, BlackBerry.

Responsa

The bas

- Key g
- Certi
- CRL

he digital signature of a
public key is previously
lic key to a given entity:
ess, email, RFC, CURP,

warranty that a given
owner of its
.

ted by a trustworhty
ority.. perhaps, in
to trust to Mozilla,
erry.

Digital certificate

Is a document in which the digital signature of a trustworthy entity, whose public key is previously stored, associates the public key to a given entity: name, organization, address, email, RFC, CURP, etc.

The certificate serves to warranty that a given public key belongs to the owner of its corresponding private key.

These certificate are granted by a trustworhty entity, a Certificate Authority.. perhaps, in practice, we delegate who to trust to Mozilla, Microsoft, Apple, BlackBerry.

Responsabilities of a C

The basic responsabilities

- Key generation (Secur
- Certificate Emission
- CRL publication

re of a
viously
n entity:
CURP,

iven

rhty
n
illa,

Digital certificate

Is a document in which the digital signature of a trustworthy entity, whose public key is previously stored, associates the public key to a given entity: name, organization, address, email, RFC, CURP, etc.

The certificate serves to warranty that a given public key belongs to the owner of its corresponding private key.

These certificate are granted by a trustworhty entity, a Certificate Authority.. perhaps, in practice, we delegate who to trust to Mozilla, Microsoft, Apple, BlackBerry.

Responsabilities of a CA

The basic responsabilities are:

- Key generation (Secure exchange)
- Certificate Emission
- CRL publication

Digital certificate

Is a document in which the digital signature of a trustworthy entity, whose public key is previously stored, associates the public key to a given entity: name, organization, address, email, RFC, CURP, etc.

The certificate serves to warranty that a given public key belongs to the owner of its corresponding private key.

These certificate are granted by a trustworhty entity, a Certificate Authority.. perhaps, in practice, we delegate who to trust to Mozilla, Microsoft, Apple, BlackBerry.

Responsabilities of a CA

The basic responsabilities are:

- Key generation (Secure exchange)
- Certificate Emission
- CRL publication

ertificate

ument in which the digital signature of a thy entity, whose public key is previously ssociates the public key to a given entity: rganization, address, email, RFC, CURP,

tificate serves to warranty that a given ey belongs to the owner of its nding private key.

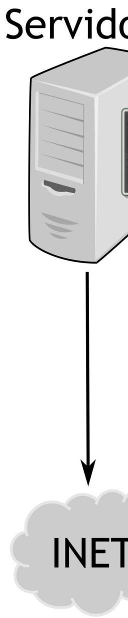
ertificate are granted by a trustworhty Certificate Authority.. perhaps, in we delegate who to trust to Mozilla, ft, Apple, BlackBerry.

Responsabilities of a CA

The basic responsabilities are:

- Key generation (Secure exchange)
- Certificate Emission
- CRL publication

Certifica



he digital signature of a
 public key is previously
lic key to a given entity:
ess, email, RFC, CURP,

warranty that a given
 owner of its
.

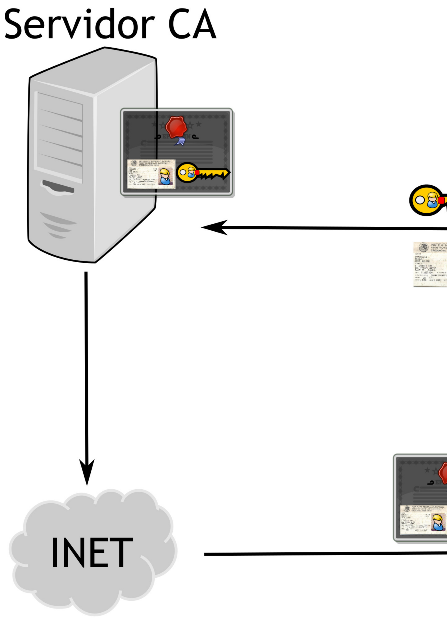
ted by a trustworhty
ority.. perhaps, in
 to trust to Mozilla,
erry.

Responsabilities of a CA

The basic responsabilities are:

- Key generation (Secure exchange)
- Certificate Emission
- CRL publication

Certificates



re of a
viously
n entity:
CURP,

iven

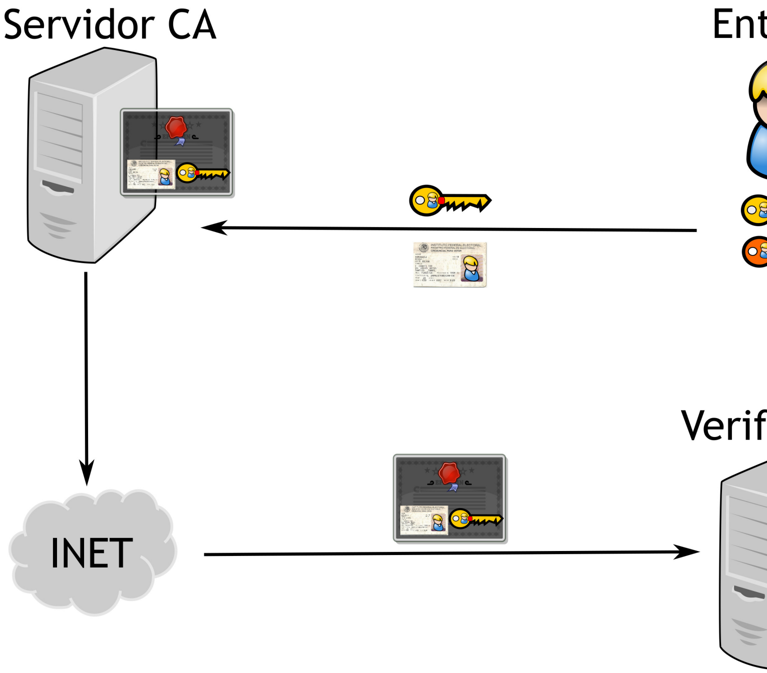
rhty
n
illa,

Responsabilities of a CA

The basic responsabilities are:

- Key generation (Secure exchange)
- Certificate Emission
- CRL publication

Certificates

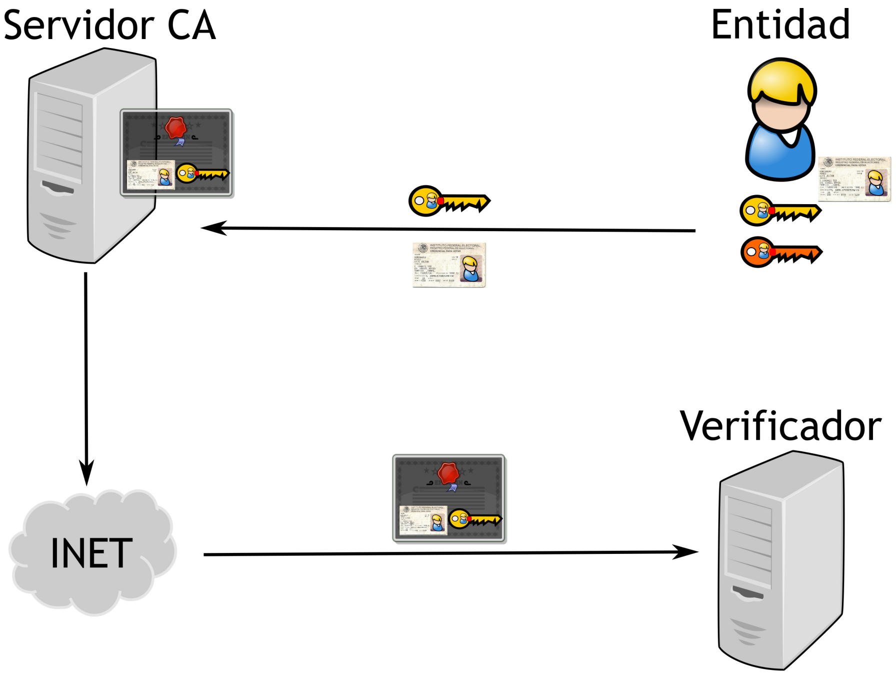


Responsabilities of a CA

The basic responsabilities are:

- Key generation (Secure exchange)
- Certificate Emission
- CRL publication

Certificates



abilities of a CA

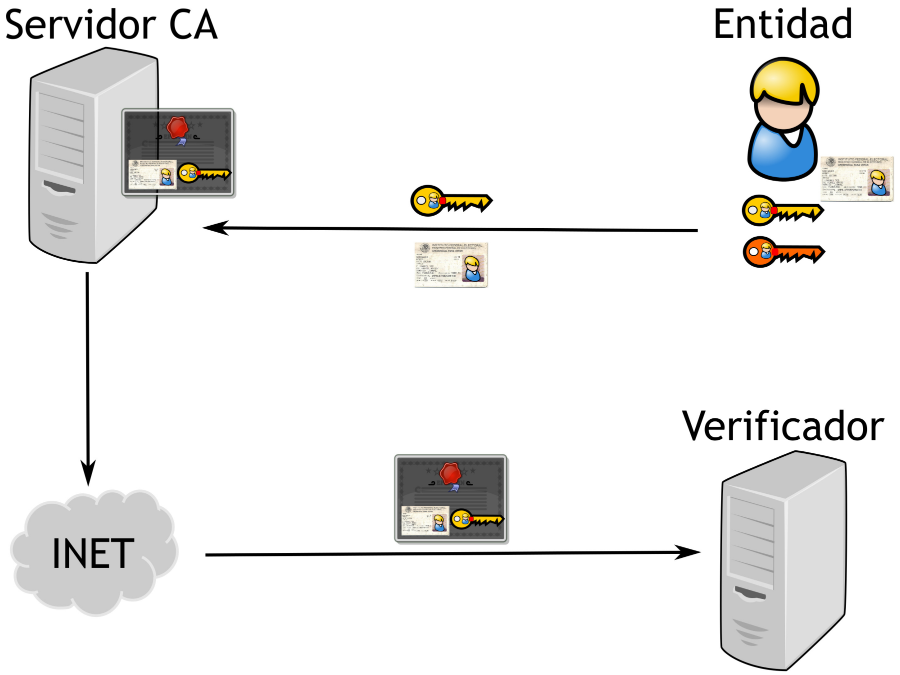
ic responsabilities are:

generation (Secure exchange)

ficate Emission

publication

Certificates



Contents

The X.5
the digit

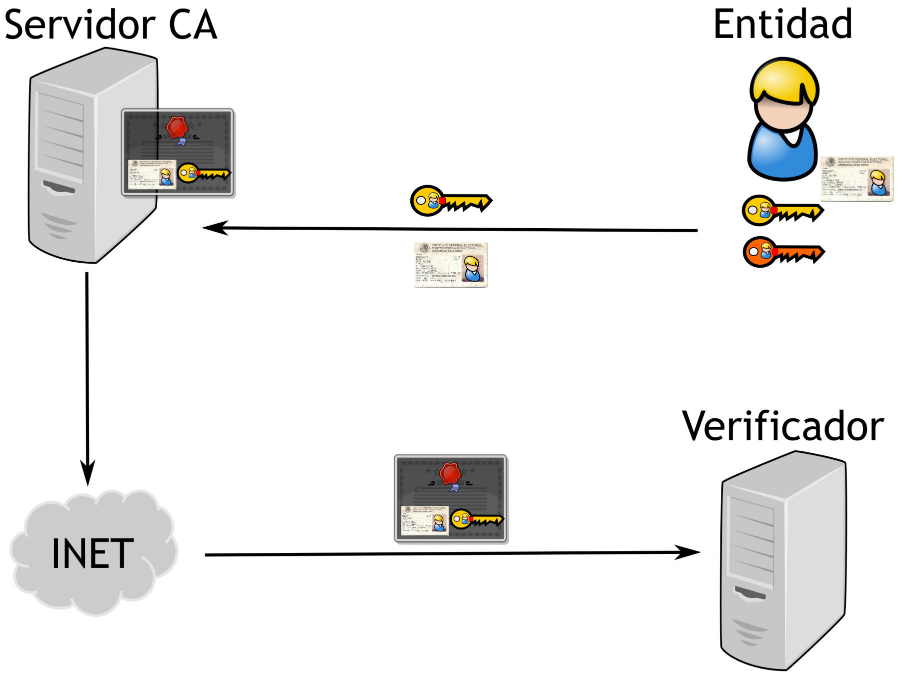
- Seria
- Subje
- Digit
- Digit
- Emitt
- Rang
- Publi
  signa
- Publi
- Hash
- Hash

CA

s are:

e exchange)

Certificates

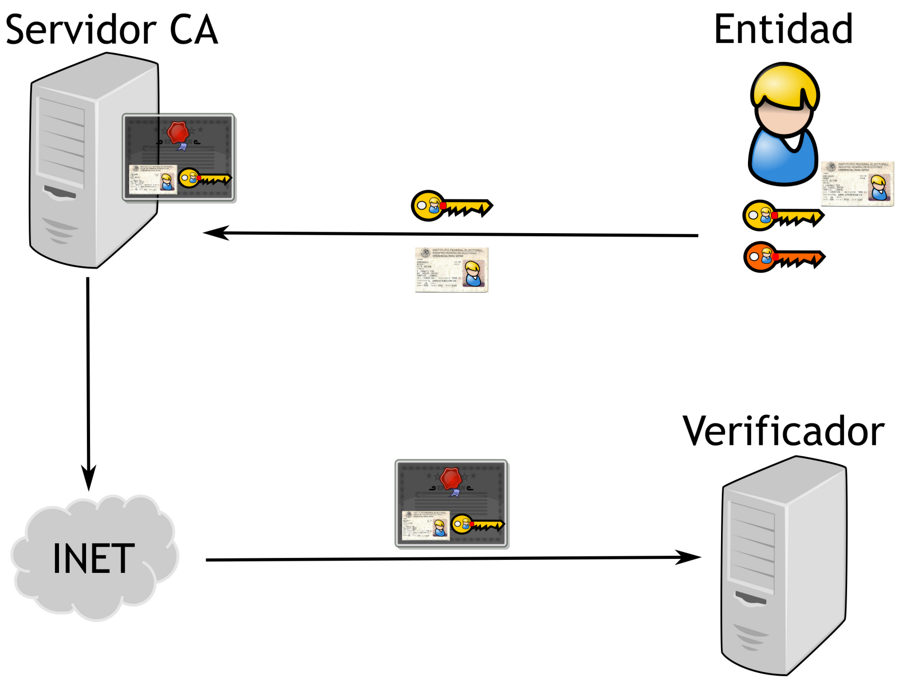


Contents of a certifica

The X.509 certificate set
the digital certificates, w

- Serial number (which
- Subject: Person, or en
- Digital Signature Algo
- Digital Signature
- Emitter
- Range of dates of vali
- Public Key allowed us
  signature, certificate e
- Public Key
- Hashing algorithm
- Hash

Certificates

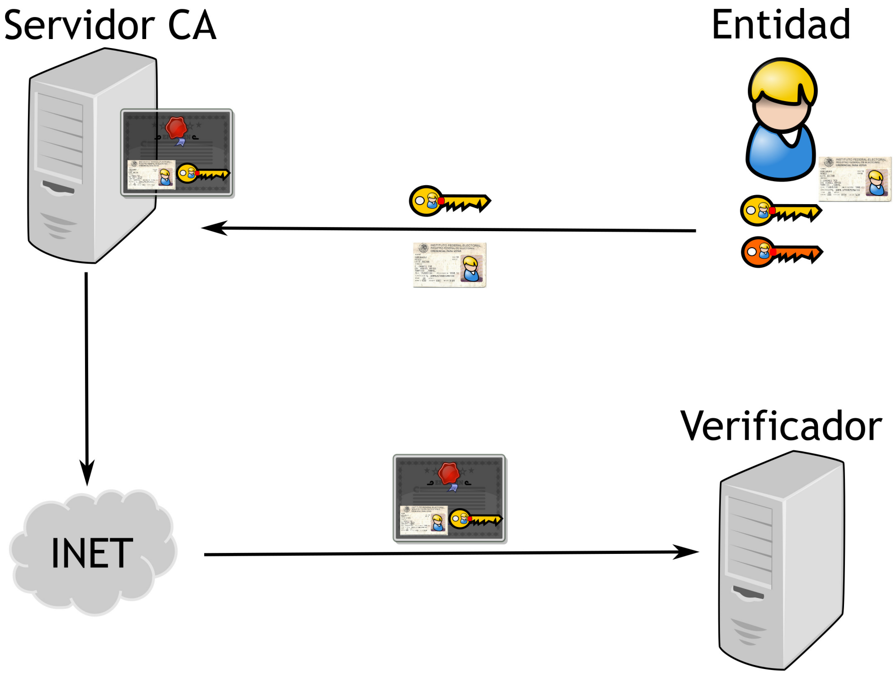


Contents of a certificate

The X.509 certificate sets the ASN1 form
the digital certificates, which contain::

- Serial number (which is no longer cons
- Subject: Person, or entity to identify
- Digital Signature Algorithm
- Digital Signature
- Emitter
- Range of dates of validity
- Public Key allowed usage: encription, signature, certificate emission
- Public Key
- Hashing algorithm
- Hash

Certificates

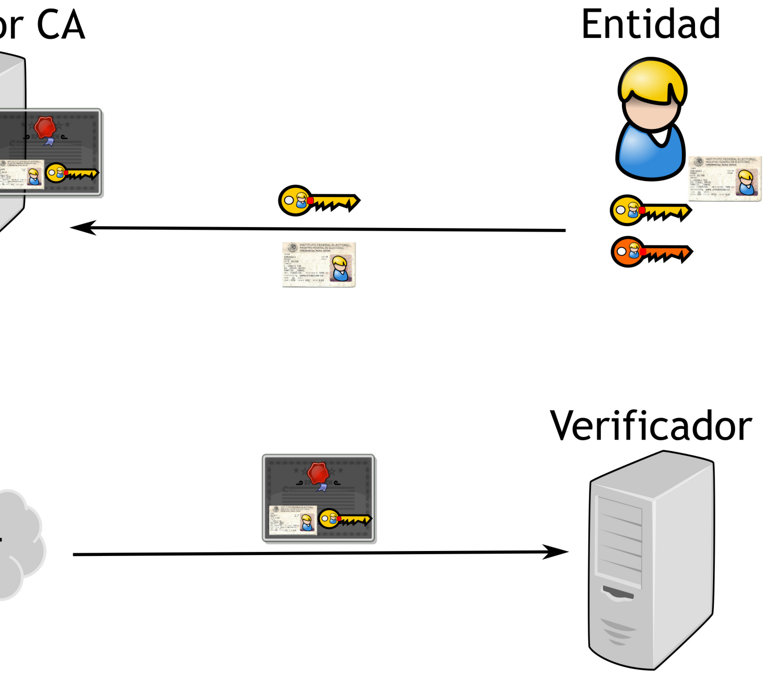


Contents of a certificate

The X.509 certificate sets the ASN1 format for the digital certificates, which contain::

- Serial number (which is no longer consecutive)
- Subject: Person, or entity to identify
- Digital Signature Algorithm
- Digital Signature
- Emitter
- Range of dates of validity
- Public Key allowed usage: encription, signature, certificate emission
- Public Key
- Hashing algorithm
- Hash

tes



Contents of a certificate

The X.509 certificate sets the ASN1 format for the digital certificates, which contain::

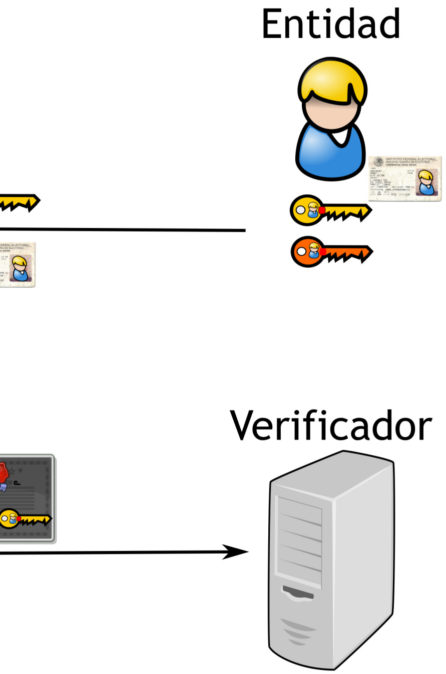
- Serial number (which is no longer consecutive)
- Subject: Person, or entity to identify
- Digital Signature Algorithm
- Digital Signature
- Emitter
- Range of dates of validity
- Public Key allowed usage: encription, signature, certificate emission
- Public Key
- Hashing algorithm
- Hash

Example

```
Certificate:
    Data:
        Vers
        Seri

    Signatur
        Issu
        Vali


        Subj
        Subj
```



Contents of a certificate

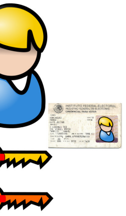
The X.509 certificate sets the ASN1 format for the digital certificates, which contain::

- Serial number (which is no longer consecutive)
- Subject: Person, or entity to identify
- Digital Signature Algorithm
- Digital Signature
- Emitter
- Range of dates of validity
- Public Key allowed usage: encription, signature, certificate emission
- Public Key
- Hashing algorithm
- Hash

Example

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            07:23:53:8d:87:6d:b6:27:
    Signature Algorithm: sha1WithRSA
        Issuer: C=US, O=DigiCert Inc
        Validity
            Not Before: Oct  8 00:00
            Not After : Dec 16 12:00
        Subject: C=MX, ST=Distrito F
        Subject Public Key Info:
            Public Key Algorithm: rs
                Public-Key: (2048 bi
                Modulus:
                    00:d8:dc:9d:1a:7
                    05:8a:c1:0b:3f:b
                    c1:59:ec:13:68:5
                    84:4a:e7:97:55:8
                    be:5c:23:2d:ab:3
                    46:23:39:20:78:d
                    8d:7d:33:98:b3:f
                    55:87:13:a5:54:b
                    1f:e6:29:01:1e:a
                    88:6f:e5:b0:4b:b
                    c7:73:ff:00:0b:6
                    0f:e9:15:70:f8:7
                    65:47:5f:a2:8f:8
                    90:12:5c:1c:46:2
                    d3:f3:53:a1:5e:a
                    2a:45:7d:73:6d:6
                    2b:a5:22:06:22:4
                    b6:a7
```

idad



icador

Contents of a certificate

The X.509 certificate sets the ASN1 format for the digital certificates, which contain::

- Serial number (which is no longer consecutive)
- Subject: Person, or entity to identify
- Digital Signature Algorithm
- Digital Signature
- Emitter
- Range of dates of validity
- Public Key allowed usage: encription, signature, certificate emission
- Public Key
- Hashing algorithm
- Hash

Example

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            07:23:53:8d:87:6d:b6:27:fc:1e:08:aa:49:96:d9:60
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, O=DigiCert Inc, OU=www.digicert.com, C
        Validity
            Not Before: Oct  8 00:00:00 2012 GMT
            Not After : Dec 16 12:00:00 2015 GMT
        Subject: C=MX, ST=Distrito Federal, L=Mexico, O=Cent
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d8:dc:9d:1a:7e:d4:6f:49:5b:7a:95:6a:5
                    05:8a:c1:0b:3f:b1:03:e0:1a:53:e5:22:8f:b
                    c1:59:ec:13:68:5e:f2:6f:44:55:21:36:8c:8
                    84:4a:e7:97:55:84:f2:cf:71:ad:e4:e5:a6:7
                    be:5c:23:2d:ab:3b:5d:b7:c3:de:2f:0a:35:7
                    46:23:39:20:78:d4:8b:47:eb:e1:d4:b4:c2:a
                    8d:7d:33:98:b3:f7:bf:3a:07:c0:64:8a:4f:a
                    55:87:13:a5:54:b5:e7:be:15:dc:da:9d:61:8
                    1f:e6:29:01:1e:ab:61:5d:bf:06:cb:ec:48:8
                    88:6f:e5:b0:4b:bf:83:bd:a0:58:bf:ff:33:0
                    c7:73:ff:00:0b:64:f2:2b:9a:69:3f:d5:74:d
                    0f:e9:15:70:f8:7c:f1:2b:5c:70:d4:49:ce:0
                    65:47:5f:a2:8f:8f:fa:af:2a:00:c9:ec:20:f
                    90:12:5c:1c:46:2b:44:24:04:77:44:82:98:2
                    d3:f3:53:a1:5e:a0:f5:f0:1f:f5:6b:22:27:9
                    2a:45:7d:73:6d:68:39:cf:d2:d2:60:3a:fd:6
                    2b:a5:22:06:22:46:c2:90:a6:8b:dd:95:61:7
                    b6:a7
```

Contents of a certificate

The X.509 certificate sets the ASN1 format for the digital certificates, which contain::

- Serial number (which is no longer consecutive)
- Subject: Person, or entity to identify
- Digital Signature Algorithm
- Digital Signature
- Emitter
- Range of dates of validity
- Public Key allowed usage: encription, signature, certificate emission
- Public Key
- Hashing algorithm
- Hash

Example

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            07:23:53:8d:87:6d:b6:27:fc:1e:08:aa:49:96:d9:60
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert High Ass
        Validity
            Not Before: Oct  8 00:00:00 2012 GMT
            Not After : Dec 16 12:00:00 2015 GMT
        Subject: C=MX, ST=Distrito Federal, L=Mexico, O=Centro de Investigacion
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d8:dc:9d:1a:7e:d4:6f:49:5b:7a:95:6a:57:6c:
                    05:8a:c1:0b:3f:b1:03:e0:1a:53:e5:22:8f:bd:6c:
                    c1:59:ec:13:68:5e:f2:6f:44:55:21:36:8c:82:d9:
                    84:4a:e7:97:55:84:f2:cf:71:ad:e4:e5:a6:73:5c:
                    be:5c:23:2d:ab:3b:5d:b7:c3:de:2f:0a:35:74:84:
                    46:23:39:20:78:d4:8b:47:eb:e1:d4:b4:c2:ab:59:
                    8d:7d:33:98:b3:f7:bf:3a:07:c0:64:8a:4f:a6:78:
                    55:87:13:a5:54:b5:e7:be:15:dc:da:9d:61:8c:06:
                    1f:e6:29:01:1e:ab:61:5d:bf:06:cb:ec:48:89:b0:
                    88:6f:e5:b0:4b:bf:83:bd:a0:58:bf:ff:33:0d:f8:
                    c7:73:ff:00:0b:64:f2:2b:9a:69:3f:d5:74:d3:12:
                    0f:e9:15:70:f8:7c:f1:2b:5c:70:d4:49:ce:01:c9:
                    65:47:5f:a2:8f:8f:fa:af:2a:00:c9:ec:20:fd:33:
                    90:12:5c:1c:46:2b:44:24:04:77:44:82:98:26:93:
                    d3:f3:53:a1:5e:a0:f5:f0:1f:f5:6b:22:27:94:a9:
                    2a:45:7d:73:6d:68:39:cf:d2:d2:60:3a:fd:6a:89:
                    2b:a5:22:06:22:46:c2:90:a6:8b:dd:95:61:7b:89:
                    b6:a7
```

s of a certificate

09 certificate sets the ASN1 format for
al certificates, which contain::

number (which is no longer consecutive)

ct: Person, or entity to identify

al Signature Algorithm

al Signature

er

e of dates of validity

c Key allowed usage: encription,
ture, certificate emission

c Key

ing algorithm

Example

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            07:23:53:8d:87:6d:b6:27:fc:1e:08:aa:49:96:d9:60
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert High Ass
        Validity
            Not Before: Oct  8 00:00:00 2012 GMT
            Not After : Dec 16 12:00:00 2015 GMT
        Subject: C=MX, ST=Distrito Federal, L=Mexico, O=Centro de Investigacion
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d8:dc:9d:1a:7e:d4:6f:49:5b:7a:95:6a:57:6c:
                    05:8a:c1:0b:3f:b1:03:e0:1a:53:e5:22:8f:bd:6c:
                    c1:59:ec:13:68:5e:f2:6f:44:55:21:36:8c:82:d9:
                    84:4a:e7:97:55:84:f2:cf:71:ad:e4:e5:a6:73:5c:
                    be:5c:23:2d:ab:3b:5d:b7:c3:de:2f:0a:35:74:84:
                    46:23:39:20:78:d4:8b:47:eb:e1:d4:b4:c2:ab:59:
                    8d:7d:33:98:b3:f7:bf:3a:07:c0:64:8a:4f:a6:78:
                    55:87:13:a5:54:b5:e7:be:15:dc:da:9d:61:8c:06:
                    1f:e6:29:01:1e:ab:61:5d:bf:06:cb:ec:48:89:b0:
                    88:6f:e5:b0:4b:bf:83:bd:a0:58:bf:ff:33:0d:f8:
                    c7:73:ff:00:0b:64:f2:2b:9a:69:3f:d5:74:d3:12:
                    0f:e9:15:70:f8:7c:f1:2b:5c:70:d4:49:ce:01:c9:
                    65:47:5f:a2:8f:8f:fa:af:2a:00:c9:ec:20:fd:33:
                    90:12:5c:1c:46:2b:44:24:04:77:44:82:98:26:93:
                    d3:f3:53:a1:5e:a0:f5:f0:1f:f5:6b:22:27:94:a9:
                    2a:45:7d:73:6d:68:39:cf:d2:d2:60:3a:fd:6a:89:
                    2b:a5:22:06:22:46:c2:90:a6:8b:dd:95:61:7b:89:
                    b6:a7
```

X50

te

s the ASN1 format for
hich contain::

is no longer consecutive)

tity to identify

rithm

dity

age: encription,
mission

Example

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            07:23:53:8d:87:6d:b6:27:fc:1e:08:aa:49:96:d9:60
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert High Ass
        Validity
            Not Before: Oct  8 00:00:00 2012 GMT
            Not After : Dec 16 12:00:00 2015 GMT
        Subject: C=MX, ST=Distrito Federal, L=Mexico, O=Centro de Investigacion
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d8:dc:9d:1a:7e:d4:6f:49:5b:7a:95:6a:57:6c:
                    05:8a:c1:0b:3f:b1:03:e0:1a:53:e5:22:8f:bd:6c:
                    c1:59:ec:13:68:5e:f2:6f:44:55:21:36:8c:82:d9:
                    84:4a:e7:97:55:84:f2:cf:71:ad:e4:e5:a6:73:5c:
                    be:5c:23:2d:ab:3b:5d:b7:c3:de:2f:0a:35:74:84:
                    46:23:39:20:78:d4:8b:47:eb:e1:d4:b4:c2:ab:59:
                    8d:7d:33:98:b3:f7:bf:3a:07:c0:64:8a:4f:a6:78:
                    55:87:13:a5:54:b5:e7:be:15:dc:da:9d:61:8c:06:
                    1f:e6:29:01:1e:ab:61:5d:bf:06:cb:ec:48:89:b0:
                    88:6f:e5:b0:4b:bf:83:bd:a0:58:bf:ff:33:0d:f8:
                    c7:73:ff:00:0b:64:f2:2b:9a:69:3f:d5:74:d3:12:
                    0f:e9:15:70:f8:7c:f1:2b:5c:70:d4:49:ce:01:c9:
                    65:47:5f:a2:8f:8f:fa:af:2a:00:c9:ec:20:fd:33:
                    90:12:5c:1c:46:2b:44:24:04:77:44:82:98:26:93:
                    d3:f3:53:a1:5e:a0:f5:f0:1f:f5:6b:22:27:94:a9:
                    2a:45:7d:73:6d:68:39:cf:d2:d2:60:3a:fd:6a:89:
                    2b:a5:22:06:22:46:c2:90:a6:8b:dd:95:61:7b:89:
                    b6:a7
```

```
                Exponent: 65537 (0x1
        X509v3 extensions:
            X509v3 Authority Key Ide
                keyid:50:EA:73:89:DB

            X509v3 Subject Key Ident
                37:92:15:14:C3:5C:87
            X509v3 Subject Alternati
                DNS:*.cinvestav.mx,
                 DNS:webmail.tamps.c
            X509v3 Key Usage: critic
                Digital Signature, K
            X509v3 Extended Key Usag
                TLS Web Server Authe
            X509v3 CRL Distribution

                Full Name:
                  URI:http://crl3.di

                Full Name:
                  URI:http://crl4.di

            X509v3 Certificate Polic
                Policy: 2.16.840.1.1
                  CPS: http://www.di
                  User Notice:
                    Explicit Text:

            Authority Information Ac
                OCSP - URI:http://oc
                CA Issuers - URI:htt

            X509v3 Basic Constraints
                CA:FALSE
```

at for

ecutive)

Example

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            07:23:53:8d:87:6d:b6:27:fc:1e:08:aa:49:96:d9:60
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert High Ass
        Validity
            Not Before: Oct  8 00:00:00 2012 GMT
            Not After : Dec 16 12:00:00 2015 GMT
        Subject: C=MX, ST=Distrito Federal, L=Mexico, O=Centro de Investigacion
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d8:dc:9d:1a:7e:d4:6f:49:5b:7a:95:6a:57:6c:
                    05:8a:c1:0b:3f:b1:03:e0:1a:53:e5:22:8f:bd:6c:
                    c1:59:ec:13:68:5e:f2:6f:44:55:21:36:8c:82:d9:
                    84:4a:e7:97:55:84:f2:cf:71:ad:e4:e5:a6:73:5c:
                    be:5c:23:2d:ab:3b:5d:b7:c3:de:2f:0a:35:74:84:
                    46:23:39:20:78:d4:8b:47:eb:e1:d4:b4:c2:ab:59:
                    8d:7d:33:98:b3:f7:bf:3a:07:c0:64:8a:4f:a6:78:
                    55:87:13:a5:54:b5:e7:be:15:dc:da:9d:61:8c:06:
                    1f:e6:29:01:1e:ab:61:5d:bf:06:cb:ec:48:89:b0:
                    88:6f:e5:b0:4b:bf:83:bd:a0:58:bf:ff:33:0d:f8:
                    c7:73:ff:00:0b:64:f2:2b:9a:69:3f:d5:74:d3:12:
                    0f:e9:15:70:f8:7c:f1:2b:5c:70:d4:49:ce:01:c9:
                    65:47:5f:a2:8f:8f:fa:af:2a:00:c9:ec:20:fd:33:
                    90:12:5c:1c:46:2b:44:24:04:77:44:82:98:26:93:
                    d3:f3:53:a1:5e:a0:f5:f0:1f:f5:6b:22:27:94:a9:
                    2a:45:7d:73:6d:68:39:cf:d2:d2:60:3a:fd:6a:89:
                    2b:a5:22:06:22:46:c2:90:a6:8b:dd:95:61:7b:89:
                    b6:a7
```

```
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Authority Key Identifier:
                keyid:50:EA:73:89:DB:29:FB:10:8F:9E:E5:01:20

            X509v3 Subject Key Identifier:
                37:92:15:14:C3:5C:87:5F:C4:63:E2:F3:20:C1:8F
            X509v3 Subject Alternative Name:
                DNS:*.cinvestav.mx, DNS:cinvestav.mx, DNS:ww
                 DNS:webmail.tamps.cinvestav.mx, DNS:noc.tam
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Clien
            X509v3 CRL Distribution Points:

                Full Name:
                  URI:http://crl3.digicert.com/ca3-g15.crl

                Full Name:
                  URI:http://crl4.digicert.com/ca3-g15.crl

            X509v3 Certificate Policies:
                Policy: 2.16.840.1.114412.1.1
                  CPS: http://www.digicert.com/ssl-cps-repos
                  User Notice:
                    Explicit Text:

            Authority Information Access:
                OCSP - URI:http://ocsp.digicert.com
                CA Issuers - URI:http://cacerts.digicert.com

            X509v3 Basic Constraints: critical
                CA:FALSE
```

Example

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            07:23:53:8d:87:6d:b6:27:fc:1e:08:aa:49:96:d9:60
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert High Ass
        Validity
            Not Before: Oct  8 00:00:00 2012 GMT
            Not After : Dec 16 12:00:00 2015 GMT
        Subject: C=MX, ST=Distrito Federal, L=Mexico, O=Centro de Investigacion
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d8:dc:9d:1a:7e:d4:6f:49:5b:7a:95:6a:57:6c:
                    05:8a:c1:0b:3f:b1:03:e0:1a:53:e5:22:8f:bd:6c:
                    c1:59:ec:13:68:5e:f2:6f:44:55:21:36:8c:82:d9:
                    84:4a:e7:97:55:84:f2:cf:71:ad:e4:e5:a6:73:5c:
                    be:5c:23:2d:ab:3b:5d:b7:c3:de:2f:0a:35:74:84:
                    46:23:39:20:78:d4:8b:47:eb:e1:d4:b4:c2:ab:59:
                    8d:7d:33:98:b3:f7:bf:3a:07:c0:64:8a:4f:a6:78:
                    55:87:13:a5:54:b5:e7:be:15:dc:da:9d:61:8c:06:
                    1f:e6:29:01:1e:ab:61:5d:bf:06:cb:ec:48:89:b0:
                    88:6f:e5:b0:4b:bf:83:bd:a0:58:bf:ff:33:0d:f8:
                    c7:73:ff:00:0b:64:f2:2b:9a:69:3f:d5:74:d3:12:
                    0f:e9:15:70:f8:7c:f1:2b:5c:70:d4:49:ce:01:c9:
                    65:47:5f:a2:8f:8f:fa:af:2a:00:c9:ec:20:fd:33:
                    90:12:5c:1c:46:2b:44:24:04:77:44:82:98:26:93:
                    d3:f3:53:a1:5e:a0:f5:f0:1f:f5:6b:22:27:94:a9:
                    2a:45:7d:73:6d:68:39:cf:d2:d2:60:3a:fd:6a:89:
                    2b:a5:22:06:22:46:c2:90:a6:8b:dd:95:61:7b:89:
                    b6:a7
```

```
        Exponent: 65537 (0x10001)
X509v3 extensions:
    X509v3 Authority Key Identifier:
        keyid:50:EA:73:89:DB:29:FB:10:8F:9E:E5:01:20:D4:DE:79:99:48:83:

    X509v3 Subject Key Identifier:
        37:92:15:14:C3:5C:87:5F:C4:63:E2:F3:20:C1:8F:0C:92:B7:BC:7D
    X509v3 Subject Alternative Name:
        DNS:*.cinvestav.mx, DNS:cinvestav.mx, DNS:www.tamps.cinvestav.m
         DNS:webmail.tamps.cinvestav.mx, DNS:noc.tamps.cinvestav.mx
    X509v3 Key Usage: critical
        Digital Signature, Key Encipherment
    X509v3 Extended Key Usage:
        TLS Web Server Authentication, TLS Web Client Authentication
    X509v3 CRL Distribution Points:

        Full Name:
          URI:http://crl3.digicert.com/ca3-g15.crl

        Full Name:
          URI:http://crl4.digicert.com/ca3-g15.crl

    X509v3 Certificate Policies:
        Policy: 2.16.840.1.114412.1.1
          CPS: http://www.digicert.com/ssl-cps-repository.htm
          User Notice:
            Explicit Text:

    Authority Information Access:
        OCSP - URI:http://ocsp.digicert.com
        CA Issuers - URI:http://cacerts.digicert.com/DigiCertHighAssura

    X509v3 Basic Constraints: critical
        CA:FALSE
```

```
sion: 3 (0x2)
al Number:
07:23:53:8d:87:6d:b6:27:fc:1e:08:aa:49:96:d9:60
re Algorithm: sha1WithRSAEncryption
uer: C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert High Ass
dity
Not Before: Oct  8 00:00:00 2012 GMT
Not After : Dec 16 12:00:00 2015 GMT
ect: C=MX, ST=Distrito Federal, L=Mexico, O=Centro de Investigacion
ect Public Key Info:
Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
    Modulus:
        00:d8:dc:9d:1a:7e:d4:6f:49:5b:7a:95:6a:57:6c:
        05:8a:c1:0b:3f:b1:03:e0:1a:53:e5:22:8f:bd:6c:
        c1:59:ec:13:68:5e:f2:6f:44:55:21:36:8c:82:d9:
        84:4a:e7:97:55:84:f2:cf:71:ad:e4:e5:a6:73:5c:
        be:5c:23:2d:ab:3b:5d:b7:c3:de:2f:0a:35:74:84:
        46:23:39:20:78:d4:8b:47:eb:e1:d4:b4:c2:ab:59:
        8d:7d:33:98:b3:f7:bf:3a:07:c0:64:8a:4f:a6:78:
        55:87:13:a5:54:b5:e7:be:15:dc:da:9d:61:8c:06:
        1f:e6:29:01:1e:ab:61:5d:bf:06:cb:ec:48:89:b0:
        88:6f:e5:b0:4b:bf:83:bd:a0:58:bf:ff:33:0d:f8:
        c7:73:ff:00:0b:64:f2:2b:9a:69:3f:d5:74:d3:12:
        0f:e9:15:70:f8:7c:f1:2b:5c:70:d4:49:ce:01:c9:
        65:47:5f:a2:8f:8f:fa:af:2a:00:c9:ec:20:fd:33:
        90:12:5c:1c:46:2b:44:24:04:77:44:82:98:26:93:
        d3:f3:53:a1:5e:a0:f5:f0:1f:f5:6b:22:27:94:a9:
        2a:45:7d:73:6d:68:39:cf:d2:d2:60:3a:fd:6a:89:
        2b:a5:22:06:22:46:c2:90:a6:8b:dd:95:61:7b:89:
        b6:a7
```

```
        Exponent: 65537 (0x10001)
X509v3 extensions:
    X509v3 Authority Key Identifier:
        keyid:50:EA:73:89:DB:29:FB:10:8F:9E:E5:01:20:D4:DE:79:99:48:83:

    X509v3 Subject Key Identifier:
        37:92:15:14:C3:5C:87:5F:C4:63:E2:F3:20:C1:8F:0C:92:B7:BC:7D
    X509v3 Subject Alternative Name:
        DNS:*.cinvestav.mx, DNS:cinvestav.mx, DNS:www.tamps.cinvestav.m
         DNS:webmail.tamps.cinvestav.mx, DNS:noc.tamps.cinvestav.mx
    X509v3 Key Usage: critical
        Digital Signature, Key Encipherment
    X509v3 Extended Key Usage:
        TLS Web Server Authentication, TLS Web Client Authentication
    X509v3 CRL Distribution Points:

        Full Name:
          URI:http://crl3.digicert.com/ca3-g15.crl

        Full Name:
          URI:http://crl4.digicert.com/ca3-g15.crl

    X509v3 Certificate Policies:
        Policy: 2.16.840.1.114412.1.1
          CPS: http://www.digicert.com/ssl-cps-repository.htm
          User Notice:
            Explicit Text:

    Authority Information Access:
        OCSP - URI:http://ocsp.digicert.com
        CA Issuers - URI:http://cacerts.digicert.com/DigiCertHighAssura

    X509v3 Basic Constraints: critical
        CA:FALSE
```

```
Signatu
     89:
     a7:
     58:
     57:
     0d:
     d2:
     e8:
     af:
     75:
     f9:
     0a:
     cd:
     bb:
     43:
     d7:
```

```
fc:1e:08:aa:49:96:d9:60
Encryption
, OU=www.digicert.com, CN=DigiCert High Ass

:00 2012 GMT
:00 2015 GMT
ederal, L=Mexico, O=Centro de Investigacion

aEncryption
t)

e:d4:6f:49:5b:7a:95:6a:57:6c:
1:03:e0:1a:53:e5:22:8f:bd:6c:
e:f2:6f:44:55:21:36:8c:82:d9:
4:f2:cf:71:ad:e4:e5:a6:73:5c:
b:5d:b7:c3:de:2f:0a:35:74:84:
4:8b:47:eb:e1:d4:b4:c2:ab:59:
7:bf:3a:07:c0:64:8a:4f:a6:78:
5:e7:be:15:dc:da:9d:61:8c:06:
b:61:5d:bf:06:cb:ec:48:89:b0:
f:83:bd:a0:58:bf:ff:33:0d:f8:
4:f2:2b:9a:69:3f:d5:74:d3:12:
c:f1:2b:5c:70:d4:49:ce:01:c9:
f:fa:af:2a:00:c9:ec:20:fd:33:
b:44:24:04:77:44:82:98:26:93:
0:f5:f0:1f:f5:6b:22:27:94:a9:
8:39:cf:d2:d2:60:3a:fd:6a:89:
6:c2:90:a6:8b:dd:95:61:7b:89:
```

```
        Exponent: 65537 (0x10001)
X509v3 extensions:
    X509v3 Authority Key Identifier:
        keyid:50:EA:73:89:DB:29:FB:10:8F:9E:E5:01:20:D4:DE:79:99:48:83:

    X509v3 Subject Key Identifier:
        37:92:15:14:C3:5C:87:5F:C4:63:E2:F3:20:C1:8F:0C:92:B7:BC:7D
    X509v3 Subject Alternative Name:
        DNS:*.cinvestav.mx, DNS:cinvestav.mx, DNS:www.tamps.cinvestav.m
         DNS:webmail.tamps.cinvestav.mx, DNS:noc.tamps.cinvestav.mx
    X509v3 Key Usage: critical
        Digital Signature, Key Encipherment
    X509v3 Extended Key Usage:
        TLS Web Server Authentication, TLS Web Client Authentication
    X509v3 CRL Distribution Points:

        Full Name:
          URI:http://crl3.digicert.com/ca3-g15.crl

        Full Name:
          URI:http://crl4.digicert.com/ca3-g15.crl

    X509v3 Certificate Policies:
        Policy: 2.16.840.1.114412.1.1
          CPS: http://www.digicert.com/ssl-cps-repository.htm
          User Notice:
            Explicit Text:

    Authority Information Access:
        OCSP - URI:http://ocsp.digicert.com
        CA Issuers - URI:http://cacerts.digicert.com/DigiCertHighAssura

    X509v3 Basic Constraints: critical
        CA:FALSE
```

```
Signature Algorithm: sha1WithRSA
     89:72:14:45:fc:52:d2:46:12:
     a7:d9:a1:6d:d4:4e:09:aa:c0:
     58:26:59:bc:95:d7:73:28:36:
     57:3d:2e:c2:9e:86:9f:08:47:
     0d:e4:f3:11:aa:69:9d:c1:6b:
     d2:b5:5e:60:ef:35:d2:bb:19:
     e8:0a:d0:d4:b0:b7:13:4f:43:
     af:3c:d7:61:89:24:6b:8a:88:
     75:ca:18:e9:11:8f:7a:c4:0a:
     f9:f5:fc:48:96:bf:e3:87:2c:
     0a:08:56:a2:be:28:ea:47:d2:
     cd:c4:14:5d:2c:13:21:6a:d0:
     bb:7c:ac:56:41:c0:64:3e:2a:
     43:02:27:eb:a5:87:71:e6:79:
     d7:3d:5f:c6
```

```
N=DigiCert High Ass

ro de Investigacion

7:6c:
d:6c:
2:d9:
3:5c:
4:84:
b:59:
6:78:
c:06:
9:b0:
d:f8:
3:12:
1:c9:
d:33:
6:93:
4:a9:
a:89:
b:89:
```

```
        Exponent: 65537 (0x10001)
X509v3 extensions:
    X509v3 Authority Key Identifier:
        keyid:50:EA:73:89:DB:29:FB:10:8F:9E:E5:01:20:D4:DE:79:99:48:83:

    X509v3 Subject Key Identifier:
        37:92:15:14:C3:5C:87:5F:C4:63:E2:F3:20:C1:8F:0C:92:B7:BC:7D
    X509v3 Subject Alternative Name:
        DNS:*.cinvestav.mx, DNS:cinvestav.mx, DNS:www.tamps.cinvestav.m
         DNS:webmail.tamps.cinvestav.mx, DNS:noc.tamps.cinvestav.mx
    X509v3 Key Usage: critical
        Digital Signature, Key Encipherment
    X509v3 Extended Key Usage:
        TLS Web Server Authentication, TLS Web Client Authentication
    X509v3 CRL Distribution Points:

        Full Name:
          URI:http://crl3.digicert.com/ca3-g15.crl

        Full Name:
          URI:http://crl4.digicert.com/ca3-g15.crl

    X509v3 Certificate Policies:
        Policy: 2.16.840.1.114412.1.1
          CPS: http://www.digicert.com/ssl-cps-repository.htm
          User Notice:
            Explicit Text:

    Authority Information Access:
        OCSP - URI:http://ocsp.digicert.com
        CA Issuers - URI:http://cacerts.digicert.com/DigiCertHighAssura

    X509v3 Basic Constraints: critical
        CA:FALSE
```

```
Signature Algorithm: sha1WithRSAEncryption
     89:72:14:45:fc:52:d2:46:12:ff:fa:f4:c5:4f:fd:7b:0e:
     a7:d9:a1:6d:d4:4e:09:aa:c0:30:2f:1a:92:eb:0c:5b:6a:
     58:26:59:bc:95:d7:73:28:36:47:d1:14:6e:e5:95:d1:ae:
     57:3d:2e:c2:9e:86:9f:08:47:a4:31:61:5d:4b:d6:3f:0a:
     0d:e4:f3:11:aa:69:9d:c1:6b:ed:ea:53:82:e0:b3:f7:cd:
     d2:b5:5e:60:ef:35:d2:bb:19:68:84:c9:c0:82:8d:e1:80:
     e8:0a:d0:d4:b0:b7:13:4f:43:24:e6:6f:37:4d:8b:f0:b9:
     af:3c:d7:61:89:24:6b:8a:88:88:82:7e:de:4c:12:8a:64:
     75:ca:18:e9:11:8f:7a:c4:0a:55:2a:d6:6a:a8:84:2e:6d:
     f9:f5:fc:48:96:bf:e3:87:2c:02:41:ab:1a:6b:ce:e3:16:
     0a:08:56:a2:be:28:ea:47:d2:03:bb:28:ab:f1:b4:ec:62:
     cd:c4:14:5d:2c:13:21:6a:d0:6e:6c:29:ba:80:9c:08:a2:
     bb:7c:ac:56:41:c0:64:3e:2a:c3:e1:44:38:a0:31:2a:68:
     43:02:27:eb:a5:87:71:e6:79:09:51:a6:82:83:28:30:0f:
     d7:3d:5f:c6
```

```
        Exponent: 65537 (0x10001)
X509v3 extensions:
    X509v3 Authority Key Identifier:
        keyid:50:EA:73:89:DB:29:FB:10:8F:9E:E5:01:20:D4:DE:79:99:48:83:

    X509v3 Subject Key Identifier:
        37:92:15:14:C3:5C:87:5F:C4:63:E2:F3:20:C1:8F:0C:92:B7:BC:7D
    X509v3 Subject Alternative Name:
        DNS:*.cinvestav.mx, DNS:cinvestav.mx, DNS:www.tamps.cinvestav.m
         DNS:webmail.tamps.cinvestav.mx, DNS:noc.tamps.cinvestav.mx
    X509v3 Key Usage: critical
        Digital Signature, Key Encipherment
    X509v3 Extended Key Usage:
        TLS Web Server Authentication, TLS Web Client Authentication
    X509v3 CRL Distribution Points:

        Full Name:
          URI:http://crl3.digicert.com/ca3-g15.crl

        Full Name:
          URI:http://crl4.digicert.com/ca3-g15.crl

    X509v3 Certificate Policies:
        Policy: 2.16.840.1.114412.1.1
          CPS: http://www.digicert.com/ssl-cps-repository.htm
          User Notice:
            Explicit Text:

    Authority Information Access:
        OCSP - URI:http://ocsp.digicert.com
        CA Issuers - URI:http://cacerts.digicert.com/DigiCertHighAssura

    X509v3 Basic Constraints: critical
        CA:FALSE
```

```
Signature Algorithm: sha1WithRSAEncryption
     89:72:14:45:fc:52:d2:46:12:ff:fa:f4:c5:4f:fd:7b:0e:e4:
     a7:d9:a1:6d:d4:4e:09:aa:c0:30:2f:1a:92:eb:0c:5b:6a:8f:
     58:26:59:bc:95:d7:73:28:36:47:d1:14:6e:e5:95:d1:ae:35:
     57:3d:2e:c2:9e:86:9f:08:47:a4:31:61:5d:4b:d6:3f:0a:60:
     0d:e4:f3:11:aa:69:9d:c1:6b:ed:ea:53:82:e0:b3:f7:cd:c4:
     d2:b5:5e:60:ef:35:d2:bb:19:68:84:c9:c0:82:8d:e1:80:e8:
     e8:0a:d0:d4:b0:b7:13:4f:43:24:e6:6f:37:4d:8b:f0:b9:0e:
     af:3c:d7:61:89:24:6b:8a:88:88:82:7e:de:4c:12:8a:64:2b:
     75:ca:18:e9:11:8f:7a:c4:0a:55:2a:d6:6a:a8:84:2e:6d:d9:
     f9:f5:fc:48:96:bf:e3:87:2c:02:41:ab:1a:6b:ce:e3:16:65:
     0a:08:56:a2:be:28:ea:47:d2:03:bb:28:ab:f1:b4:ec:62:44:
     cd:c4:14:5d:2c:13:21:6a:d0:6e:6c:29:ba:80:9c:08:a2:50:
     bb:7c:ac:56:41:c0:64:3e:2a:c3:e1:44:38:a0:31:2a:68:4b:
     43:02:27:eb:a5:87:71:e6:79:09:51:a6:82:83:28:30:0f:9a:
     d7:3d:5f:c6
```

```
    Exponent: 65537 (0x10001)
v3 extensions:
X509v3 Authority Key Identifier:
    keyid:50:EA:73:89:DB:29:FB:10:8F:9E:E5:01:20:D4:DE:79:99:48:83:

X509v3 Subject Key Identifier:
    37:92:15:14:C3:5C:87:5F:C4:63:E2:F3:20:C1:8F:0C:92:B7:BC:7D
X509v3 Subject Alternative Name:
    DNS:*.cinvestav.mx, DNS:cinvestav.mx, DNS:www.tamps.cinvestav.m
     DNS:webmail.tamps.cinvestav.mx, DNS:noc.tamps.cinvestav.mx
X509v3 Key Usage: critical
    Digital Signature, Key Encipherment
X509v3 Extended Key Usage:
    TLS Web Server Authentication, TLS Web Client Authentication
X509v3 CRL Distribution Points:

    Full Name:
      URI:http://crl3.digicert.com/ca3-g15.crl

    Full Name:
      URI:http://crl4.digicert.com/ca3-g15.crl

X509v3 Certificate Policies:
    Policy: 2.16.840.1.114412.1.1
      CPS: http://www.digicert.com/ssl-cps-repository.htm
      User Notice:
        Explicit Text:

Authority Information Access:
    OCSP - URI:http://ocsp.digicert.com
    CA Issuers - URI:http://cacerts.digicert.com/DigiCertHighAssura

X509v3 Basic Constraints: critical
    CA:FALSE
```

```
Signature Algorithm: sha1WithRSAEncryption
     89:72:14:45:fc:52:d2:46:12:ff:fa:f4:c5:4f:fd:7b:0e:e4:
     a7:d9:a1:6d:d4:4e:09:aa:c0:30:2f:1a:92:eb:0c:5b:6a:8f:
     58:26:59:bc:95:d7:73:28:36:47:d1:14:6e:e5:95:d1:ae:35:
     57:3d:2e:c2:9e:86:9f:08:47:a4:31:61:5d:4b:d6:3f:0a:60:
     0d:e4:f3:11:aa:69:9d:c1:6b:ed:ea:53:82:e0:b3:f7:cd:c4:
     d2:b5:5e:60:ef:35:d2:bb:19:68:84:c9:c0:82:8d:e1:80:e8:
     e8:0a:d0:d4:b0:b7:13:4f:43:24:e6:6f:37:4d:8b:f0:b9:0e:
     af:3c:d7:61:89:24:6b:8a:88:88:82:7e:de:4c:12:8a:64:2b:
     75:ca:18:e9:11:8f:7a:c4:0a:55:2a:d6:6a:a8:84:2e:6d:d9:
     f9:f5:fc:48:96:bf:e3:87:2c:02:41:ab:1a:6b:ce:e3:16:65:
     0a:08:56:a2:be:28:ea:47:d2:03:bb:28:ab:f1:b4:ec:62:44:
     cd:c4:14:5d:2c:13:21:6a:d0:6e:6c:29:ba:80:9c:08:a2:50:
     bb:7c:ac:56:41:c0:64:3e:2a:c3:e1:44:38:a0:31:2a:68:4b:
     43:02:27:eb:a5:87:71:e6:79:09:51:a6:82:83:28:30:0f:9a:
     d7:3d:5f:c6
```

Hands-o

Certificate

- DSA pa

  `openss`

- Key ger

  `openss`

- Self-sig

  ```
  openss
    dsar
    root
  ```

- Review

  ```
  openss
  openss
  ```

```
0001)

ntifier:
:29:FB:10:8F:9E:E5:01:20:D4:DE:79:99:48:83:

ifier:
:5F:C4:63:E2:F3:20:C1:8F:0C:92:B7:BC:7D
ve Name:
DNS:cinvestav.mx, DNS:www.tamps.cinvestav.m
investav.mx, DNS:noc.tamps.cinvestav.mx
al
ey Encipherment
e:
ntication, TLS Web Client Authentication
Points:


gicert.com/ca3-g15.crl


gicert.com/ca3-g15.crl

ies:
14412.1.1
gicert.com/ssl-cps-repository.htm




cess:
sp.digicert.com
p://cacerts.digicert.com/DigiCertHighAssura

: critical
```

```
Signature Algorithm: sha1WithRSAEncryption
     89:72:14:45:fc:52:d2:46:12:ff:fa:f4:c5:4f:fd:7b:0e:e4:
     a7:d9:a1:6d:d4:4e:09:aa:c0:30:2f:1a:92:eb:0c:5b:6a:8f:
     58:26:59:bc:95:d7:73:28:36:47:d1:14:6e:e5:95:d1:ae:35:
     57:3d:2e:c2:9e:86:9f:08:47:a4:31:61:5d:4b:d6:3f:0a:60:
     0d:e4:f3:11:aa:69:9d:c1:6b:ed:ea:53:82:e0:b3:f7:cd:c4:
     d2:b5:5e:60:ef:35:d2:bb:19:68:84:c9:c0:82:8d:e1:80:e8:
     e8:0a:d0:d4:b0:b7:13:4f:43:24:e6:6f:37:4d:8b:f0:b9:0e:
     af:3c:d7:61:89:24:6b:8a:88:88:82:7e:de:4c:12:8a:64:2b:
     75:ca:18:e9:11:8f:7a:c4:0a:55:2a:d6:6a:a8:84:2e:6d:d9:
     f9:f5:fc:48:96:bf:e3:87:2c:02:41:ab:1a:6b:ce:e3:16:65:
     0a:08:56:a2:be:28:ea:47:d2:03:bb:28:ab:f1:b4:ec:62:44:
     cd:c4:14:5d:2c:13:21:6a:d0:6e:6c:29:ba:80:9c:08:a2:50:
     bb:7c:ac:56:41:c0:64:3e:2a:c3:e1:44:38:a0:31:2a:68:4b:
     43:02:27:eb:a5:87:71:e6:79:09:51:a6:82:83:28:30:0f:9a:
     d7:3d:5f:c6
```

Hands-on

Certificate creation

- DSA parameter generati

  `openssl dsaparam 204`

- Key generation

  `openssl gendsa -out`

- Self-signed certificate ge

  ```
  openssl req -newkey
    dsarootkey.pem -ne
    rootcert.pem
  ```

- Review the certificate

  ```
  openssl x509 -text -
  openssl asn1parse -i
  ```

```
:D4:DE:79:99:48:83:

:0C:92:B7:BC:7D

v.tamps.cinvestav.m
os.cinvestav.mx

t Authentication

itory.htm

/DigiCertHighAssura
```

```
Signature Algorithm: sha1WithRSAEncryption
     89:72:14:45:fc:52:d2:46:12:ff:fa:f4:c5:4f:fd:7b:0e:e4:
     a7:d9:a1:6d:d4:4e:09:aa:c0:30:2f:1a:92:eb:0c:5b:6a:8f:
     58:26:59:bc:95:d7:73:28:36:47:d1:14:6e:e5:95:d1:ae:35:
     57:3d:2e:c2:9e:86:9f:08:47:a4:31:61:5d:4b:d6:3f:0a:60:
     0d:e4:f3:11:aa:69:9d:c1:6b:ed:ea:53:82:e0:b3:f7:cd:c4:
     d2:b5:5e:60:ef:35:d2:bb:19:68:84:c9:c0:82:8d:e1:80:e8:
     e8:0a:d0:d4:b0:b7:13:4f:43:24:e6:6f:37:4d:8b:f0:b9:0e:
     af:3c:d7:61:89:24:6b:8a:88:88:82:7e:de:4c:12:8a:64:2b:
     75:ca:18:e9:11:8f:7a:c4:0a:55:2a:d6:6a:a8:84:2e:6d:d9:
     f9:f5:fc:48:96:bf:e3:87:2c:02:41:ab:1a:6b:ce:e3:16:65:
     0a:08:56:a2:be:28:ea:47:d2:03:bb:28:ab:f1:b4:ec:62:44:
     cd:c4:14:5d:2c:13:21:6a:d0:6e:6c:29:ba:80:9c:08:a2:50:
     bb:7c:ac:56:41:c0:64:3e:2a:c3:e1:44:38:a0:31:2a:68:4b:
     43:02:27:eb:a5:87:71:e6:79:09:51:a6:82:83:28:30:0f:9a:
     d7:3d:5f:c6
```

Hands-on

Certificate creation

- DSA parameter generation

  ```
  openssl dsaparam 2048 -out dsapar
  ```

- Key generation

  ```
  openssl gendsa -out dsarootkey.pe
  ```

- Self-signed certificate generation

  ```
  openssl req -newkey dsa:dsaparams
    dsarootkey.pem -new -x509 -days
    rootcert.pem
  ```

- Review the certificate

  ```
  openssl x509 -text -in rootcert.p
  openssl asn1parse -in rootcert.pe
  ```

```
Signature Algorithm: sha1WithRSAEncryption
     89:72:14:45:fc:52:d2:46:12:ff:fa:f4:c5:4f:fd:7b:0e:e4:
     a7:d9:a1:6d:d4:4e:09:aa:c0:30:2f:1a:92:eb:0c:5b:6a:8f:
     58:26:59:bc:95:d7:73:28:36:47:d1:14:6e:e5:95:d1:ae:35:
     57:3d:2e:c2:9e:86:9f:08:47:a4:31:61:5d:4b:d6:3f:0a:60:
     0d:e4:f3:11:aa:69:9d:c1:6b:ed:ea:53:82:e0:b3:f7:cd:c4:
     d2:b5:5e:60:ef:35:d2:bb:19:68:84:c9:c0:82:8d:e1:80:e8:
     e8:0a:d0:d4:b0:b7:13:4f:43:24:e6:6f:37:4d:8b:f0:b9:0e:
     af:3c:d7:61:89:24:6b:8a:88:88:82:7e:de:4c:12:8a:64:2b:
     75:ca:18:e9:11:8f:7a:c4:0a:55:2a:d6:6a:a8:84:2e:6d:d9:
     f9:f5:fc:48:96:bf:e3:87:2c:02:41:ab:1a:6b:ce:e3:16:65:
     0a:08:56:a2:be:28:ea:47:d2:03:bb:28:ab:f1:b4:ec:62:44:
     cd:c4:14:5d:2c:13:21:6a:d0:6e:6c:29:ba:80:9c:08:a2:50:
     bb:7c:ac:56:41:c0:64:3e:2a:c3:e1:44:38:a0:31:2a:68:4b:
     43:02:27:eb:a5:87:71:e6:79:09:51:a6:82:83:28:30:0f:9a:
     d7:3d:5f:c6
```

Hands-on

Certificate creation

- DSA parameter generation

  ```
  openssl dsaparam 2048 -out dsaparams.pem
  ```

- Key generation

  ```
  openssl gendsa -out dsarootkey.pem dsaparams.pem
  ```

- Self-signed certificate generation

  ```
  openssl req -newkey dsa:dsaparams.pem -keyout
    dsarootkey.pem -new -x509 -days 365 -out
    rootcert.pem
  ```

- Review the certificate

  ```
  openssl x509 -text -in rootcert.pem | more
  openssl asn1parse -in rootcert.pem | more
  ```

```
ignature Algorithm: sha1WithRSAEncryption
    89:72:14:45:fc:52:d2:46:12:ff:fa:f4:c5:4f:fd:7b:0e:e4:
    a7:d9:a1:6d:d4:4e:09:aa:c0:30:2f:1a:92:eb:0c:5b:6a:8f:
    58:26:59:bc:95:d7:73:28:36:47:d1:14:6e:e5:95:d1:ae:35:
    57:3d:2e:c2:9e:86:9f:08:47:a4:31:61:5d:4b:d6:3f:0a:60:
    0d:e4:f3:11:aa:69:9d:c1:6b:ed:ea:53:82:e0:b3:f7:cd:c4:
    d2:b5:5e:60:ef:35:d2:bb:19:68:84:c9:c0:82:8d:e1:80:e8:
    e8:0a:d0:d4:b0:b7:13:4f:43:24:e6:6f:37:4d:8b:f0:b9:0e:
    af:3c:d7:61:89:24:6b:8a:88:88:82:7e:de:4c:12:8a:64:2b:
    75:ca:18:e9:11:8f:7a:c4:0a:55:2a:d6:6a:a8:84:2e:6d:d9:
    f9:f5:fc:48:96:bf:e3:87:2c:02:41:ab:1a:6b:ce:e3:16:65:
    0a:08:56:a2:be:28:ea:47:d2:03:bb:28:ab:f1:b4:ec:62:44:
    cd:c4:14:5d:2c:13:21:6a:d0:6e:6c:29:ba:80:9c:08:a2:50:
    bb:7c:ac:56:41:c0:64:3e:2a:c3:e1:44:38:a0:31:2a:68:4b:
    43:02:27:eb:a5:87:71:e6:79:09:51:a6:82:83:28:30:0f:9a:
    d7:3d:5f:c6
```

Hands-on

Certificate creation

- DSA parameter generation

  `openssl dsaparam 2048 -out dsaparams.pem`

- Key generation

  `openssl gendsa -out dsarootkey.pem dsaparams.pem`

- Self-signed certificate generation

  ```
  openssl req -newkey dsa:dsaparams.pem -keyout
    dsarootkey.pem -new -x509 -days 365 -out
    rootcert.pem
  ```

- Review the certificate

  ```
  openssl x509 -text -in rootcert.pem | more
  openssl asn1parse -in rootcert.pem | more
  ```

Hands-on

Certificate creation

- DSA parameter generation

```
openssl dsaparam 2048 -out dsaparams.pem
```

- Key generation

```
openssl gendsa -out dsarootkey.pem dsaparams.pem
```

- Self-signed certificate generation

```
openssl req -newkey dsa:dsaparams.pem -keyout
  dsarootkey.pem -new -x509 -days 365 -out
  rootcert.pem
```

- Review the certificate

```
openssl x509 -text -in rootcert.pem | more
openssl asn1parse -in rootcert.pem | more
```

Cliente side hands-on

- Generate a certificate for the client

```
openssl req -newkey dsa:dsaparams.pem -keyout
  dsakey.pem -new -days 365 -out dsareq.pem
```

- Certificate emission

```
openssl x509 -days 180 -CA rootcert.pem -CAkey
  dsarootkey.pem -req -CAcreateserial -CAserial
  ca.srl -in dsareq.pem -out newcert.pem
```

- Revieweing the certificate

```
openssl x509 -text -in newcert.pem | more
openssl asn1parse -in newcert.pem | more
```

- Certificate Verification

```
openssl verify -CAfile rootcert.pem newcert.pem
```

Apache configuration

- Copy the certificates files to the server
- Find the apache config file
- Identify the “VirtuaHost” block to configure

```
<VirtualHost 192.168.0.1:443>
DocumentRoot /var/www/html2
ServerName www.yourdomain.com
SSLEngine on
SSLCertificateFile /path/to/your_domain_name.crt
SSLCertificateKeyFile /path/to/your_private.key
SSLCertificateChainFile /path/to/DigiCertCA.crt
</VirtualHost>
```

Apache configuration

- Test your apache configuration
  - apachectl configtest
- Restart your apache server
  - apachectl stop
  - apachectl start

ngix configuration

- You need the CA's certificate
- Copy the certificates files to the server
- Concatenate the primary certificate and intermediate certificate
    - cat your_domain_name.crt rootcert.pem ¿¿ bundle.crt
- Edit ngix configuration file:

ngix configuration

```
server {
listen   443;
ssl    on;
ssl_certificate    /etc/ssl/your_domain_name.pem; (or bundle.c
ssl_certificate_key    /etc/ssl/your_domain_name.key;
server_name your.domain.com;
access_log /var/log/nginx/nginx.vhost.access.log;
error_log /var/log/nginx/nginx.vhost.error.log;
location / {
root   /home/www/public_html/your.domain.com/public/;
index  index.html;
}
}
```

- Restart the ngix server
  - /etc/init.d/nginx restart

```
nain_name.pem; (or bundle.c
r_domain_name.key;

st.access.log;
t.error.log;

omain.com/public/;
```

IIS configuration

```
pem; (or bundle.
ame.key;

og;
;

ublic/;
```

IIS configuration

SSL - Definiti

- SSL (Secure
standard to
a web server
the protocol

- This links en
server, and t
privacy, and

- This is the s

```
pem; (or bundle.c
ame.key;

log;
g;

public/;
```

IIS configuration

SSL - Definiti

- SSL (Secure
  standard to
  a web server
  the protocol

- This links en
  server, and t
  privacy, and

- This is the s

IIS configuration

e.

SSL - Definition

- SSL (Secure Sockets Layer) i
standard to establish an encry
a web server, and an internet
the protocol could be used fo

- This links ensures the data tr
server, and the client, and th
privacy, and integrity

- This is the standard for onlin

IIS configuration

SSL - Definition

- SSL (Secure Sockets Layer) is the security standard to establish an encrypted link between a web server, and an internet browser (perhaps, the protocol could be used for something else).
- This links ensures the data travels between the server, and the client, and that it mantains its privacy, and integrity
- This is the standard for online transactions.

guration

SSL - Definition

- SSL (Secure Sockets Layer) is the security standard to establish an encrypted link between a web server, and an internet browser (perhaps, the protocol could be used for something else).
- This links ensures the data travels between the server, and the client, and that it mantains its privacy, and integrity
- This is the standard for online transactions.

SSL - R

Ver

Generate 4

MS = (PM

Symmet

SSL - Definition

- SSL (Secure Sockets Layer) is the security standard to establish an encrypted link between a web server, and an internet browser (perhaps, the protocol could be used for something else).
- This links ensures the data travels between the server, and the client, and that it mantains its privacy, and integrity
- This is the standard for online transactions.

SSL - RSA communi

Alice

Verify Certificate
Generate 48-byte random PMS
MS = (PMS, CRnd,SRnd,etc.)

Symmetric key generation

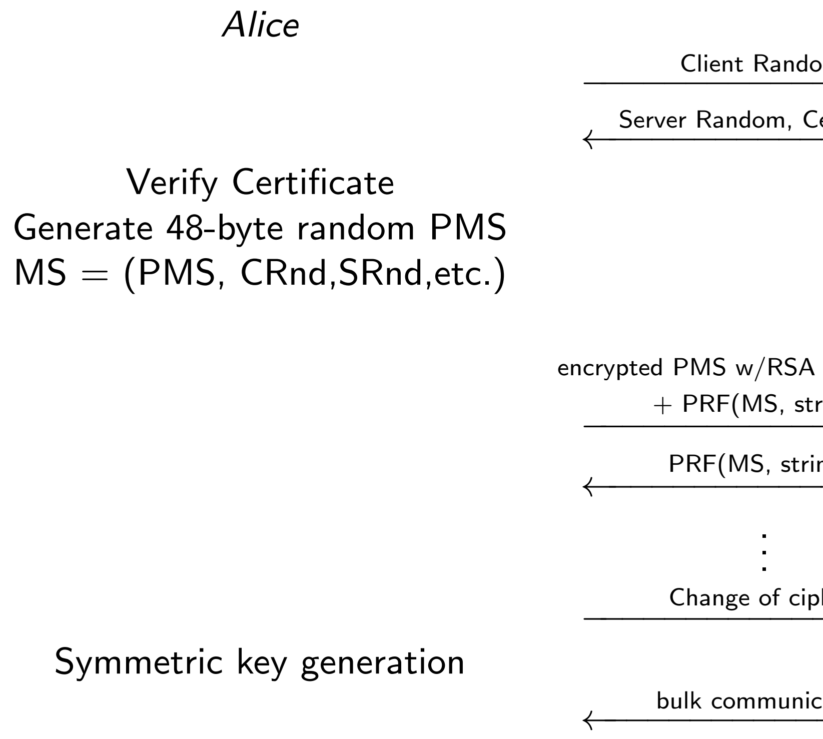
SSL - Definition

- SSL (Secure Sockets Layer) is the security standard to establish an encrypted link between a web server, and an internet browser (perhaps, the protocol could be used for something else).

- This links ensures the data travels between the server, and the client, and that it mantains its privacy, and integrity

- This is the standard for online transactions.

SSL - RSA communication



Alice

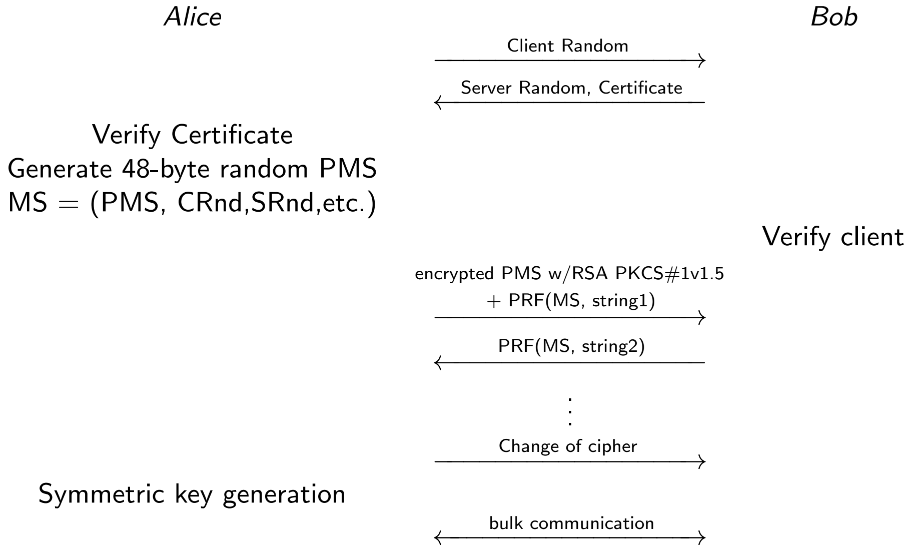
Verify Certificate
Generate 48-byte random PMS
MS = (PMS, CRnd,SRnd,etc.)

Symmetric key generation

SSL - Definition

- SSL (Secure Sockets Layer) is the security standard to establish an encrypted link between a web server, and an internet browser (perhaps, the protocol could be used for something else).
- This links ensures the data travels between the server, and the client, and that it mantains its privacy, and integrity
- This is the standard for online transactions.

SSL - RSA communication



Alice

Bob

Client Random

Server Random, Certificate

Verify Certificate
Generate 48-byte random PMS
MS = (PMS, CRnd,SRnd,etc.)

Verify client

encrypted PMS w/RSA PKCS#1v1.5
+ PRF(MS, string1)

PRF(MS, string2)

⋮

Change of cipher

Symmetric key generation

bulk communication

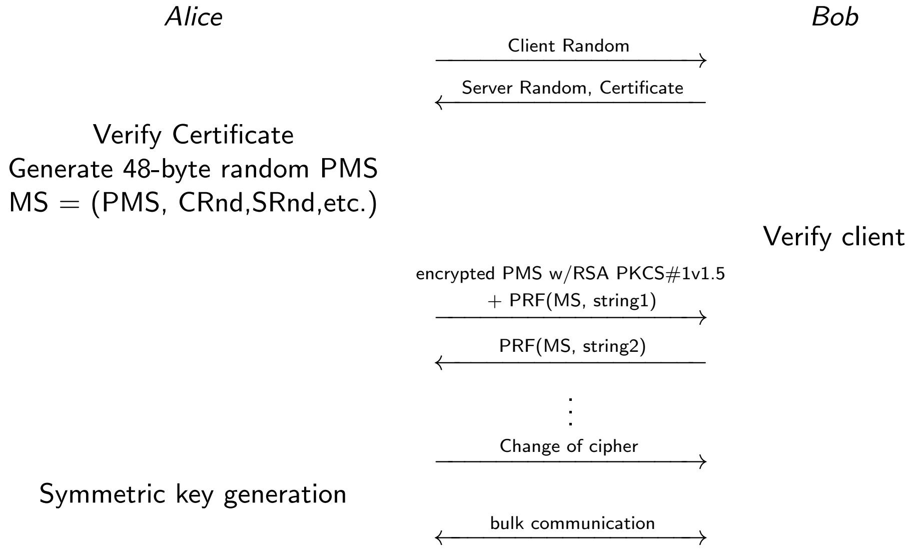
- Definition

SSL (Secure Sockets Layer) is the security tandard to establish an encrypted link between web server, and an internet browser (perhaps, he protocol could be used for something else).

This links ensures the data travels between the erver, and the client, and that it mantains its rivacy, and integrity

This is the standard for online transactions.

SSL - RSA communication



Alice

Bob

Client Random

Server Random, Certificate

Verify Certificate
Generate 48-byte random PMS
MS = (PMS, CRnd,SRnd,etc.)

Verify client

encrypted PMS w/RSA PKCS#1v1.5
+ PRF(MS, string1)

PRF(MS, string2)

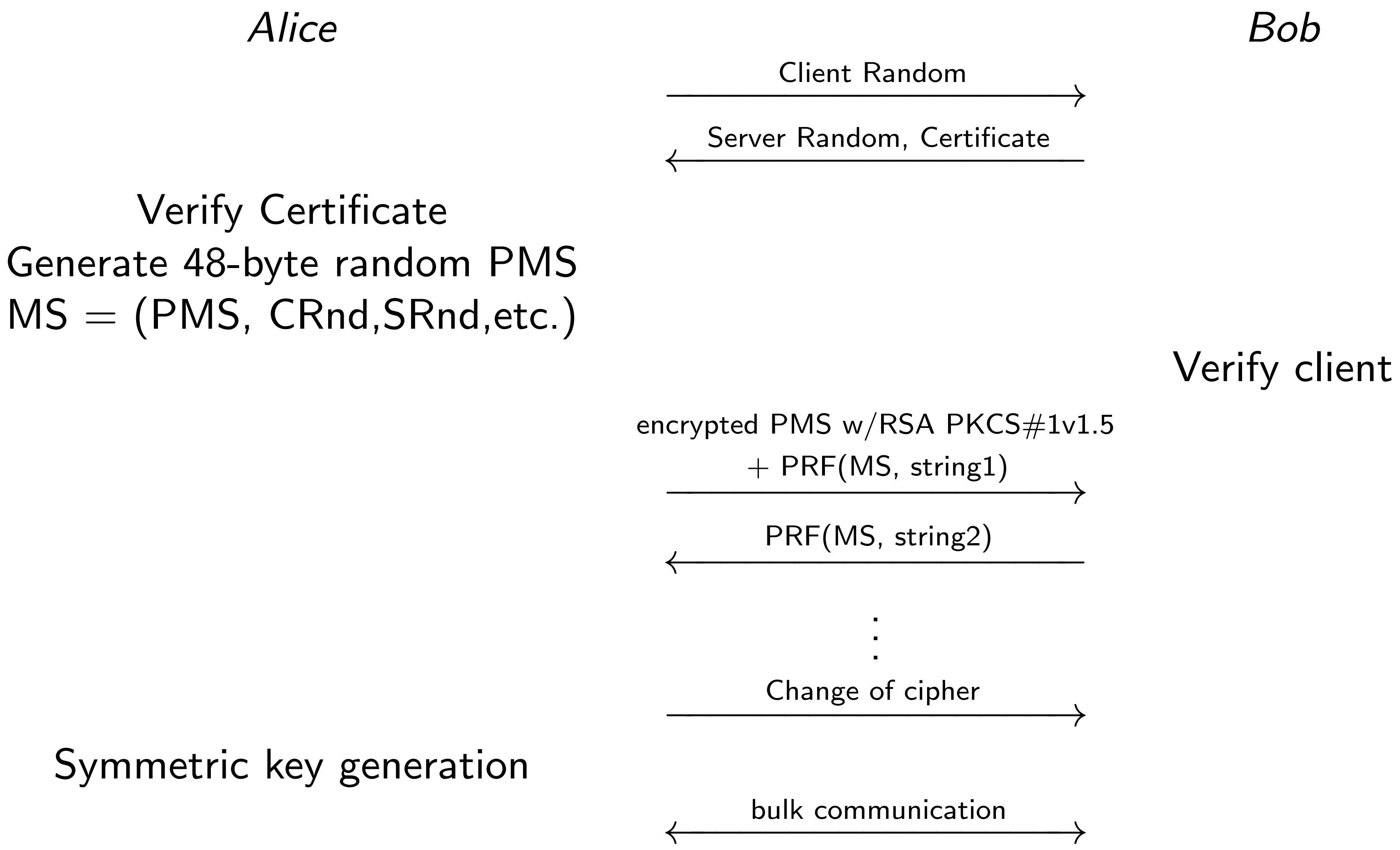
⋮

Change of cipher

Symmetric key generation

bulk communication

SSL - RSA communication



Alice

Bob

Client Random

Server Random, Certificate

Verify Certificate
Generate 48-byte random PMS
MS = (PMS, CRnd,SRnd,etc.)

Verify client

encrypted PMS w/RSA PKCS#1v1.5
+ PRF(MS, string1)

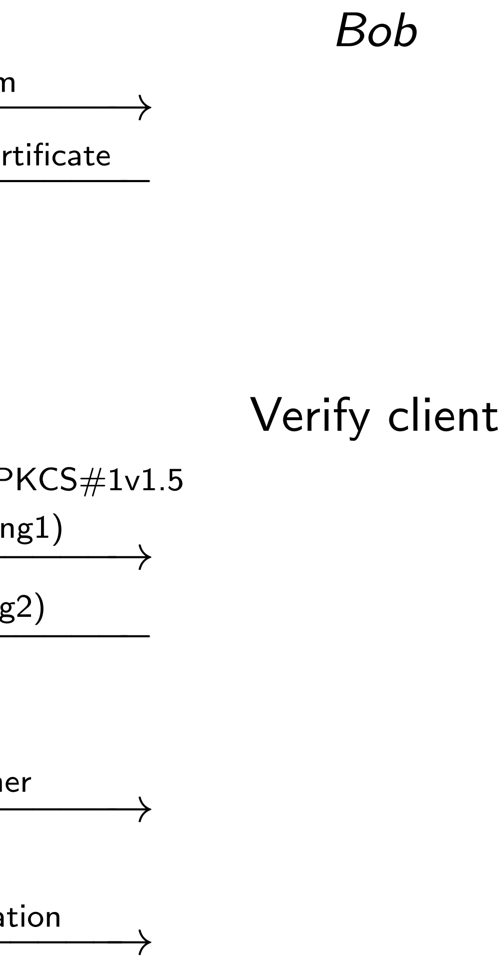
PRF(MS, string2)

⋮

Change of cipher

Symmetric key generation

bulk communication



SSL - overview

- We exchange the problem of verifying the public key of Bob (and everybody else), by the one of verifying the public of a Certificate Authority.

- Despite the are a lot of Certificate Authorities, the number of webserver is substantially larger. . .

Bob

rify client

SSL - overview

- We exchange the problem of verifying the public key of Bob (and everybody else), by the one of verifying the public of a Certificate Authority.
- Despite the are a lot of Certificate Authorities, the number of webserver is substantially larger. . .

SSL - overview

- We exchang
public key o
one of verify
Authority.
- Despite the
the number
larger. . .

- The solution
the user, sin
Certificate A
with the Op
example, we
Application

Bob

rify client

SSL - overview

- We exchange the problem of verifying the public key of Bob (and everybody else), by the one of verifying the public of a Certificate Authority.
- Despite the are a lot of Certificate Authorities, the number of webserver is substantially larger. . .

SSL - overview

- We exchang
public key o
one of verify
Authority.
- Despite the
the number
larger. . .

- The solution
the user, sin
Certificate A
with the Op
example, we
Application

SSL - overview

- We exchange the problem of verifying the public key of Bob (and everybody else), by the one of verifying the public of a Certificate Authority.

- Despite the are a lot of Certificate Authorities, the number of webserver is substantially larger. . .

SSL - overview

- We exchange the problem of
public key of Bob (and everyl
one of verifying the public of
Authority.

- Despite the are a lot of Certi
the number of webserver is su
larger. . .

- The solution to this problem
the user, since the Public Key
Certificate Authorities are ins
with the Operating System, c
example, we install an interne
Application provider does this

SSL - overview

- We exchange the problem of verifying the public key of Bob (and everybody else), by the one of verifying the public of a Certificate Authority.
- Despite the are a lot of Certificate Authorities, the number of webserver is substantially larger. . .

SSL - overview

- We exchange the problem of verifying the public key of Bob (and everybody else), by the one of verifying the public of a Certificate Authority.
- Despite the are a lot of Certificate Authorities, the number of webserver is substantially larger. . .

- The solution to this problem is transparent to the user, since the Public Keys of the Certificate Authorities are installed together with the Operating System, or when, for example, we install an internet browser. The Application provider does this for the user.

SSL - overview

- We exchange the problem of verifying the public key of Bob (and everybody else), by the one of verifying the public of a Certificate Authority.

- Despite the are a lot of Certificate Authorities, the number of webserver is substantially larger. . .

SSL - overview

- We exchange the problem of verifying the public key of Bob (and everybody else), by the one of verifying the public of a Certificate Authority.

- Despite the are a lot of Certificate Authorities, the number of webserver is substantially larger. . .

- The solution to this problem is transparent to the user, since the Public Keys of the Certificate Authorities are installed together with the Operating System, or when, for example, we install an internet browser. The Application provider does this for the user.

erview

xchange the problem of verifying the
c key of Bob (and everybody else), by the
f verifying the public of a Certificate
ority.

te the are a lot of Certificate Authorities,
umber of webserver is substantially
. . .

solution to this problem is transparent to
ser, since the Public Keys of the
ficate Authorities are installed together
the Operating System, or when, for
ple, we install an internet browser. The
cation provider does this for the user.

SSL - overview

- SSL/TLS is reduced to the problem of verifying the public keys of the other end-point (using the CA)

- It makes use of Revocation Lists to ensure no one is using a no longer valid certificate (perhaps, until recentrly, this was rarely done)

Email wi

There a
commu

- Secu
(S/M
- Prett

S/MIME
standar

PGP is
go for p

lem of verifying the
d everybody else), by the
ublic of a Certificate

f Certificate Authorities,
ver is substantially

roblem is transparent to
blic Keys of the
 are installed together
stem, or when, for
 internet browser. The
oes this for the user.

SSL - overview

- SSL/TLS is reduced to the problem of verifying the public keys of the other end-point (using the CA)
- It makes use of Revocation Lists to ensure no one is using a no longer valid certificate (perhaps, until recentrly, this was rarely done)

Email with PGP

There are two schemes to
communication:

- Secure/Multipurpose I
(S/MIME)
- Pretty Good Privacy (

S/MIME from RSA will e
standard for commercial

PGP is also on the stand
go for personal usage. I w

he
, by the
ate

norities,
y

rent to

ether
or
 The
ser.

SSL - overview

- SSL/TLS is reduced to the problem of verifying the public keys of the other end-point (using the CA)
- It makes use of Revocation Lists to ensure no one is using a no longer valid certificate (perhaps, until recentrly, this was rarely done)

Email with PGP

There are two schemes to protect email communication:

- Secure/Multipurpose Internet Mail Exte (S/MIME)
- Pretty Good Privacy (PGP)

S/MIME from RSA will emerge as the ind standard for commercial usage

PGP is also on the standardization track, go for personal usage. I will talk about PG

SSL - overview

- SSL/TLS is reduced to the problem of verifying the public keys of the other end-point (using the CA)
- It makes use of Revocation Lists to ensure no one is using a no longer valid certificate (perhaps, until recentrly, this was rarely done)

Email with PGP

There are two schemes to protect email communication:

- Secure/Multipurpose Internet Mail Extension (S/MIME)
- Pretty Good Privacy (PGP)

S/MIME from RSA will emerge as the industry standard for commercial usage

PGP is also on the standardization track, but will go for personal usage. I will talk about PGP

SSL - overview

- SSL/TLS is reduced to the problem of verifying the public keys of the other end-point (using the CA)
- It makes use of Revocation Lists to ensure no one is using a no longer valid certificate (perhaps, until recentrly, this was rarely done)

Email with PGP

There are two schemes to protect email communication:

- Secure/Multipurpose Internet Mail Extension (S/MIME)
- Pretty Good Privacy (PGP)

S/MIME from RSA will emerge as the industry standard for commercial usage

PGP is also on the standardization track, but will go for personal usage. I will talk about PGP

Email with PGP

There are two schemes to protect email communication:

- Secure/Multipurpose Internet Mail Extension (S/MIME)
- Pretty Good Privacy (PGP)

S/MIME from RSA will emerge as the industry standard for commercial usage

PGP is also on the standardization track, but will go for personal usage. I will talk about PGP

PGP

PGP consists of the following services:

- Authentication
- Confidentiality
- E-mail compatibility
- Segmentation

In a nutshell:

- Generates a session key, and encrypt it
- Signs the message
- Compress the message
- Encrypts the message
- (prepend the encrypted key to the message)

nsists of the following services:

entication

dentiality

il compatibility

entation

shell:

rates a session key, and encrypt it

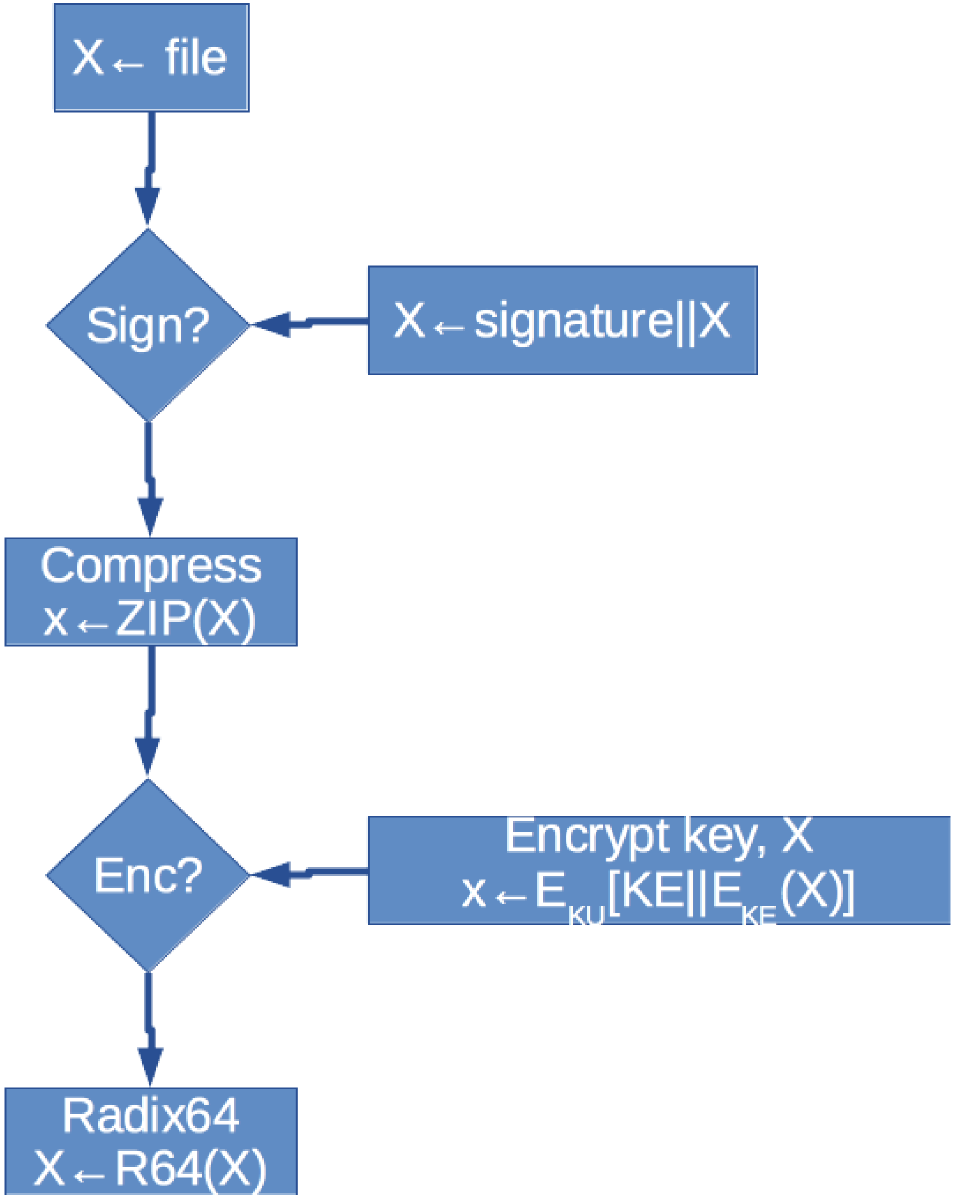
the message

press the message

ypts the message

end the encrypted key to the message)

PGP Diagram



Algoritm

- Digit
  Uses
  either
- Encry
  Diffie
  The r
  or Tr
  Diffie
  the re

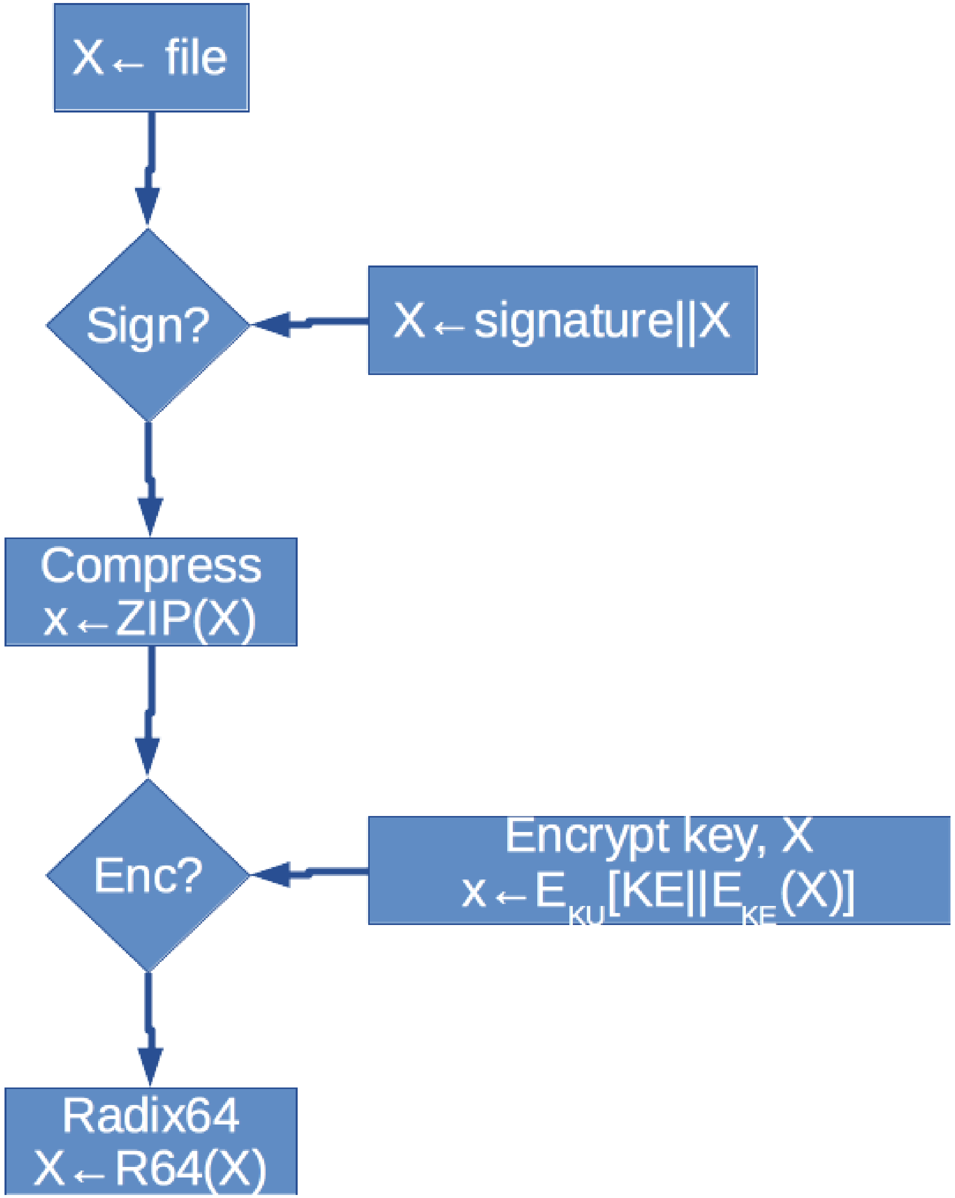
wing services:

y, and encrypt it

e

d key to the message)

PGP Diagram

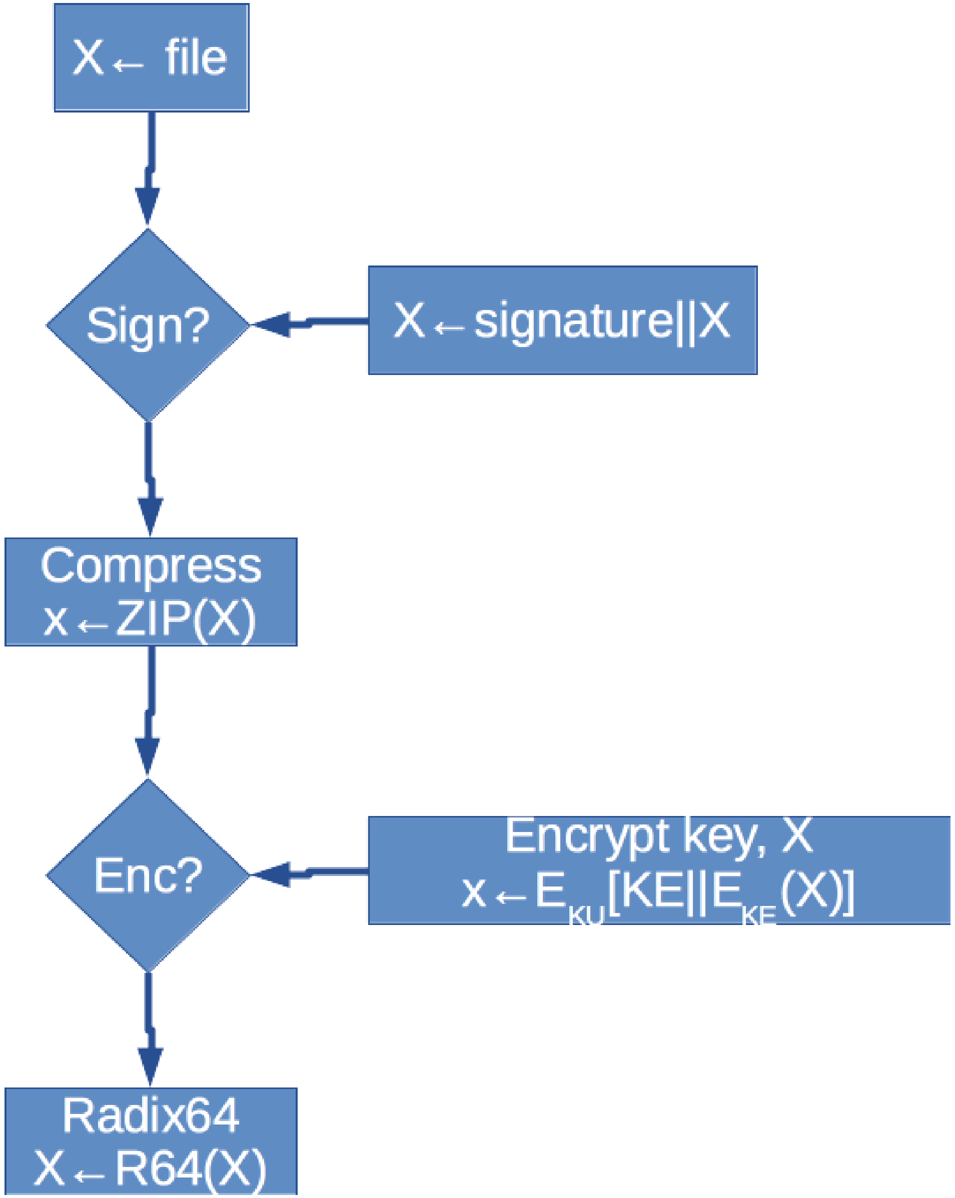


Algoritmhs used

- Digital Signature (DSS
  Uses SHA-1(!) for the
  either DSS, or RSA us
- Encryption (CAST, ID
  Diffie-Hellman, or RSA
  The message is encryp
  or Triple DES, with a
  Diffie-Hellman, or RSA
  the recipient

sage)

PGP Diagram



X ← file

Sign?

X ← signature||X

Compress
x ← ZIP(X)

Enc?

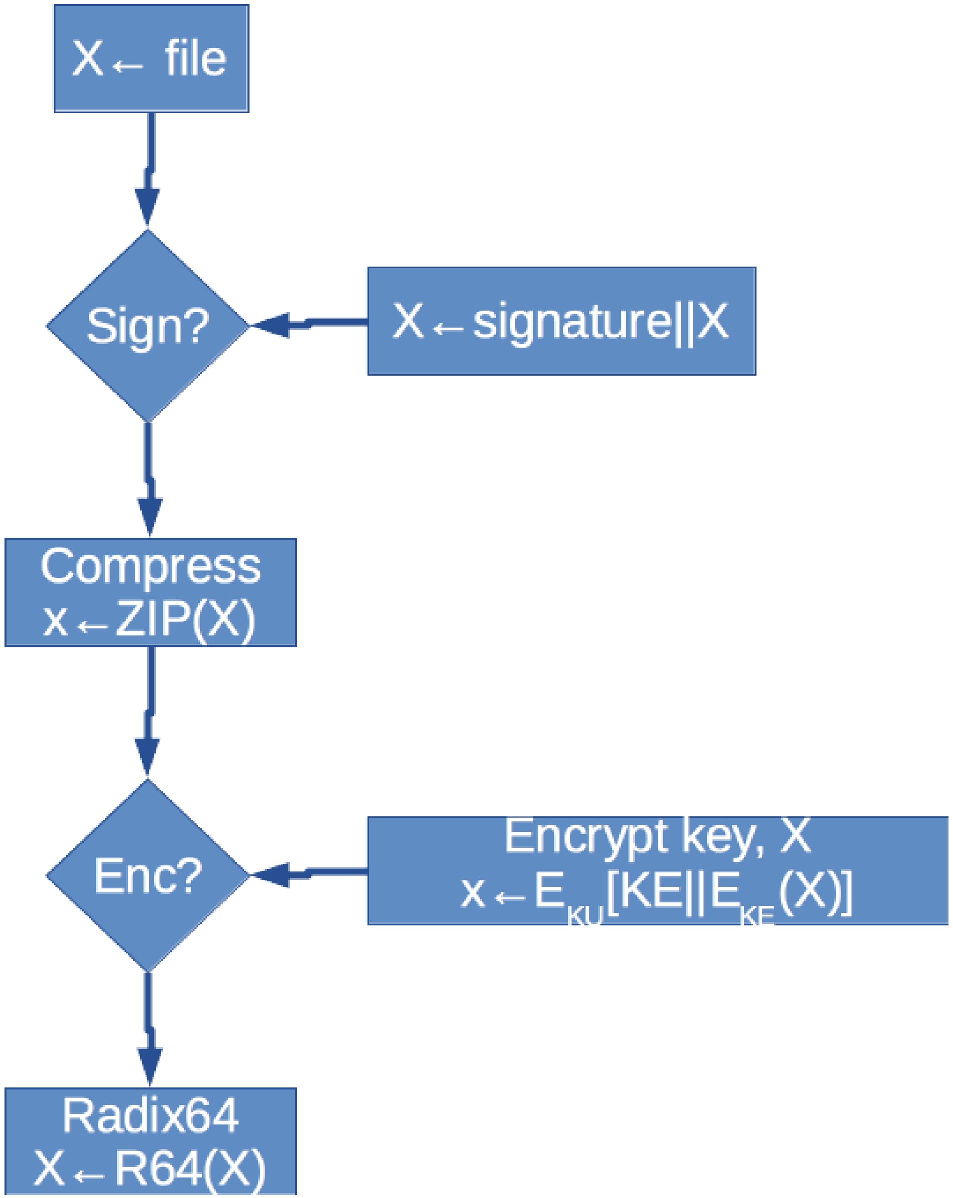
Encrypt key, X
$x \leftarrow E_{KU}[KE||E_{KE}(X)]$

Radix64
X ← R64(X)

Algoritmhs used

- Digital Signature (DSS/SHA or RSA/S
  Uses SHA-1(!) for the message, and us
  either DSS, or RSA using sender’s priva
- Encryption (CAST, IDEA, or Triple DE
  Diffie-Hellman, or RSA)
  The message is encrypted with CAST,
  or Triple DES, with a session key using
  Diffie-Hellman, or RSA with the public
  the recipient

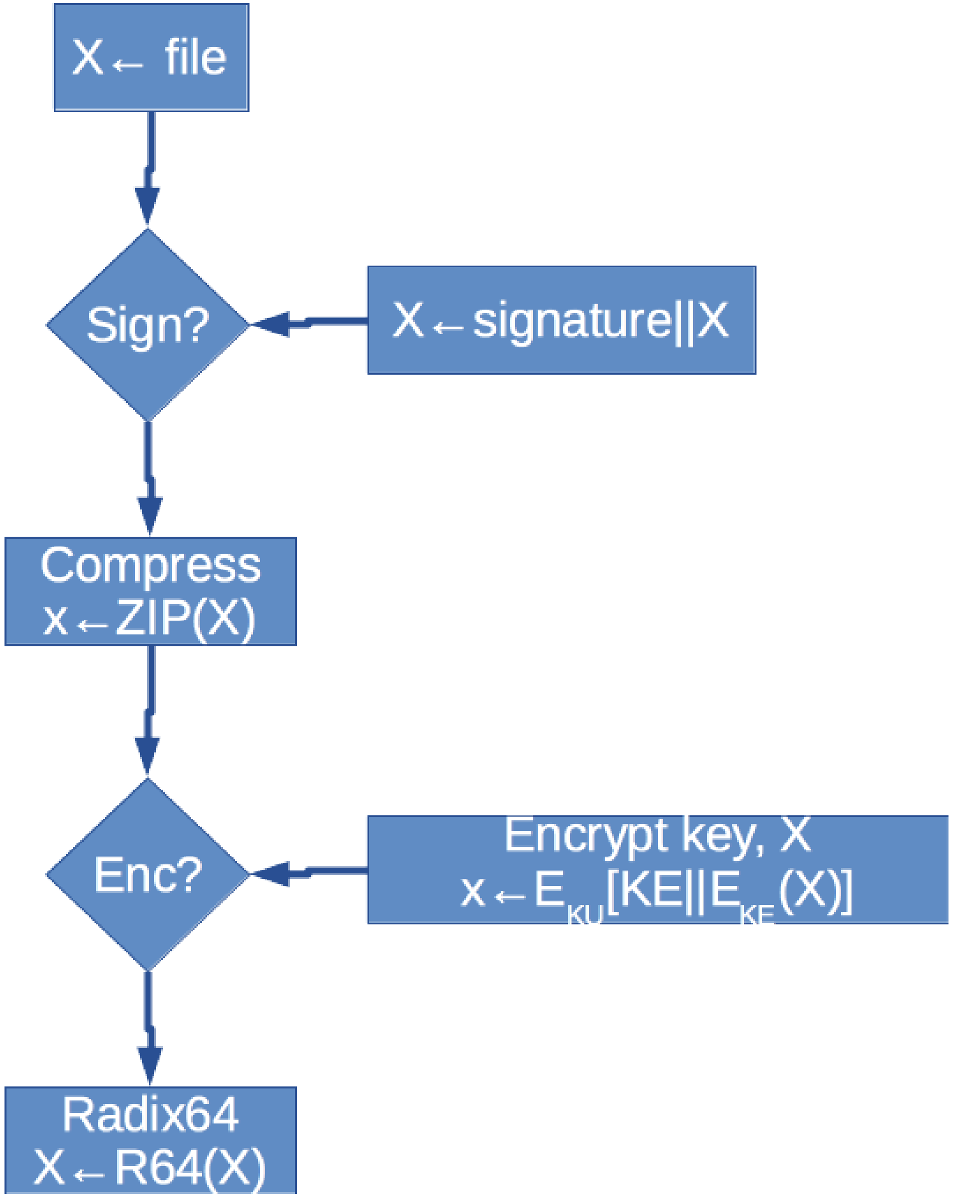
PGP Diagram



Algoritmhs used

- Digital Signature (DSS/SHA or RSA/SHA)
  Uses SHA-1(!) for the message, and uses either DSS, or RSA using sender's private key
- Encryption (CAST, IDEA, or Triple DES, with Diffie-Hellman, or RSA)
  The message is encrypted with CAST, IDEA, or Triple DES, with a session key using Diffie-Hellman, or RSA with the public key of the recipient

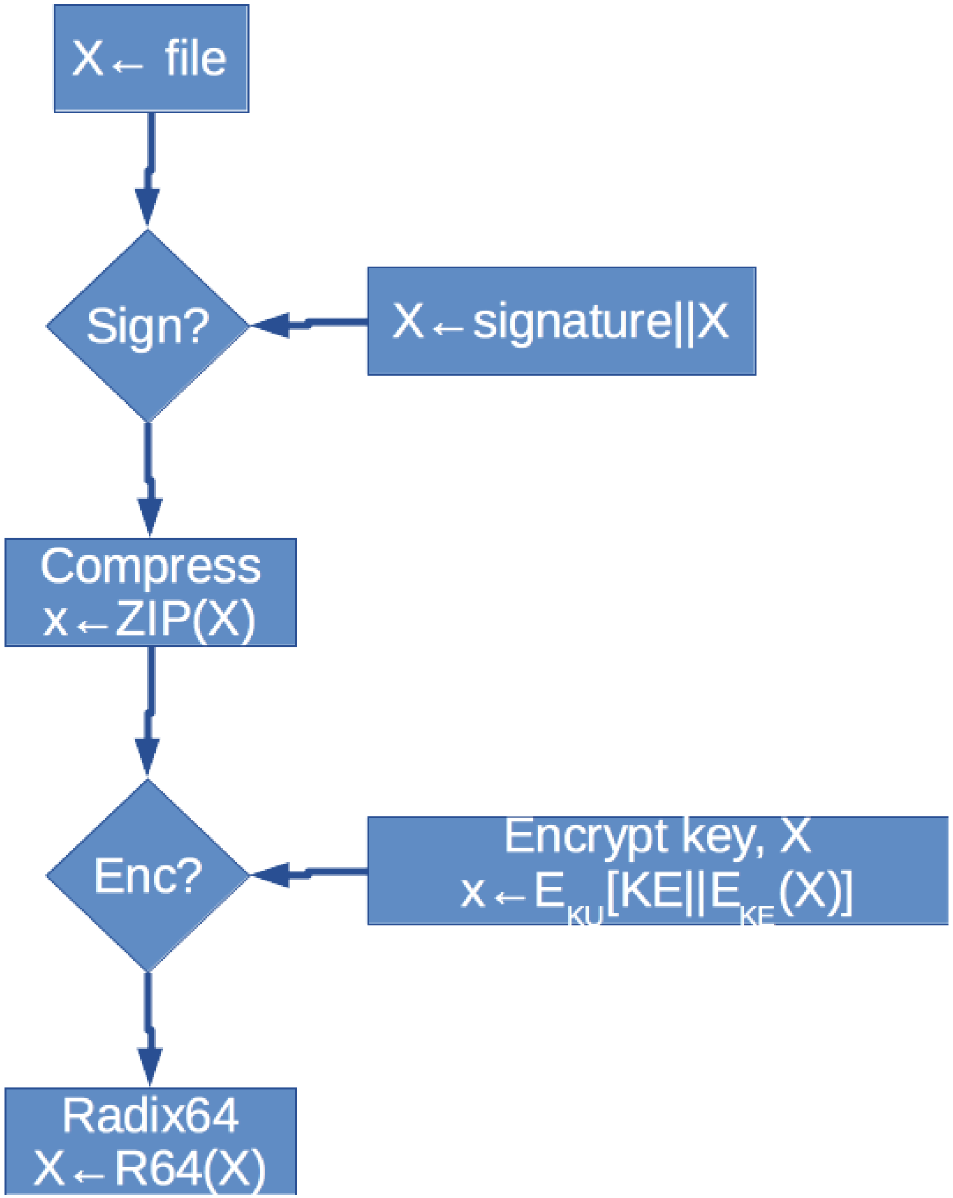
PGP Diagram



Algoritmhs used

- Digital Signature (DSS/SHA or RSA/SHA)
  Uses SHA-1(!) for the message, and uses either DSS, or RSA using sender's private key
- Encryption (CAST, IDEA, or Triple DES, with Diffie-Hellman, or RSA)
  The message is encrypted with CAST, IDEA, or Triple DES, with a session key using Diffie-Hellman, or RSA with the public key of the recipient
- Compression (ZIP)
  For transmission purposes, and for removing some of the statistic properties of the message (before encryption)

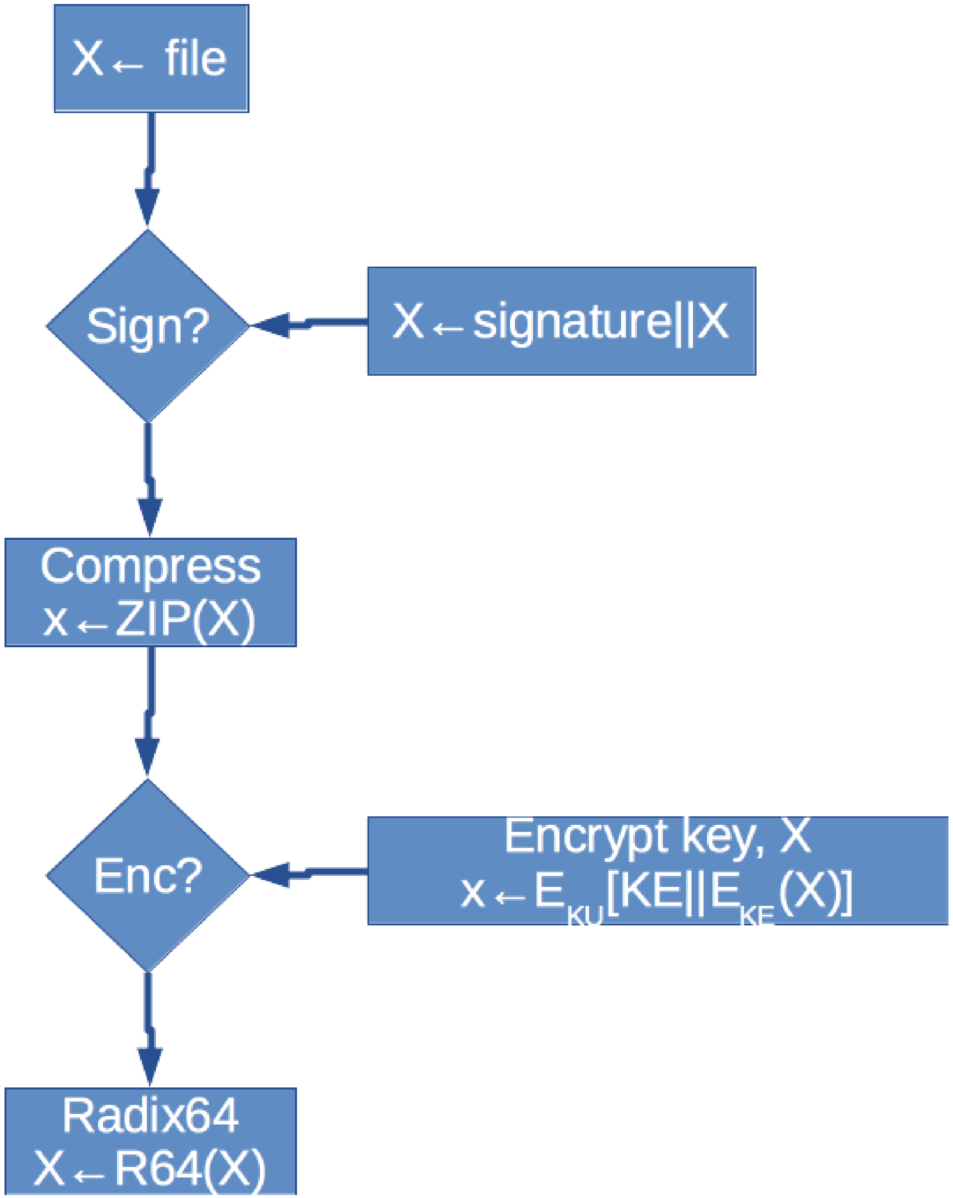
PGP Diagram



Algoritmhs used

- Digital Signature (DSS/SHA or RSA/SHA)
  Uses SHA-1(!) for the message, and uses either DSS, or RSA using sender's private key
- Encryption (CAST, IDEA, or Triple DES, with Diffie-Hellman, or RSA)
  The message is encrypted with CAST, IDEA, or Triple DES, with a session key using Diffie-Hellman, or RSA with the public key of the recipient
- Compression (ZIP)
  For transmission purposes, and for removing some of the statistic properties of the message (before encryption)
- Email compatibility (Radix 64 conversion)
  For compatibility purposes

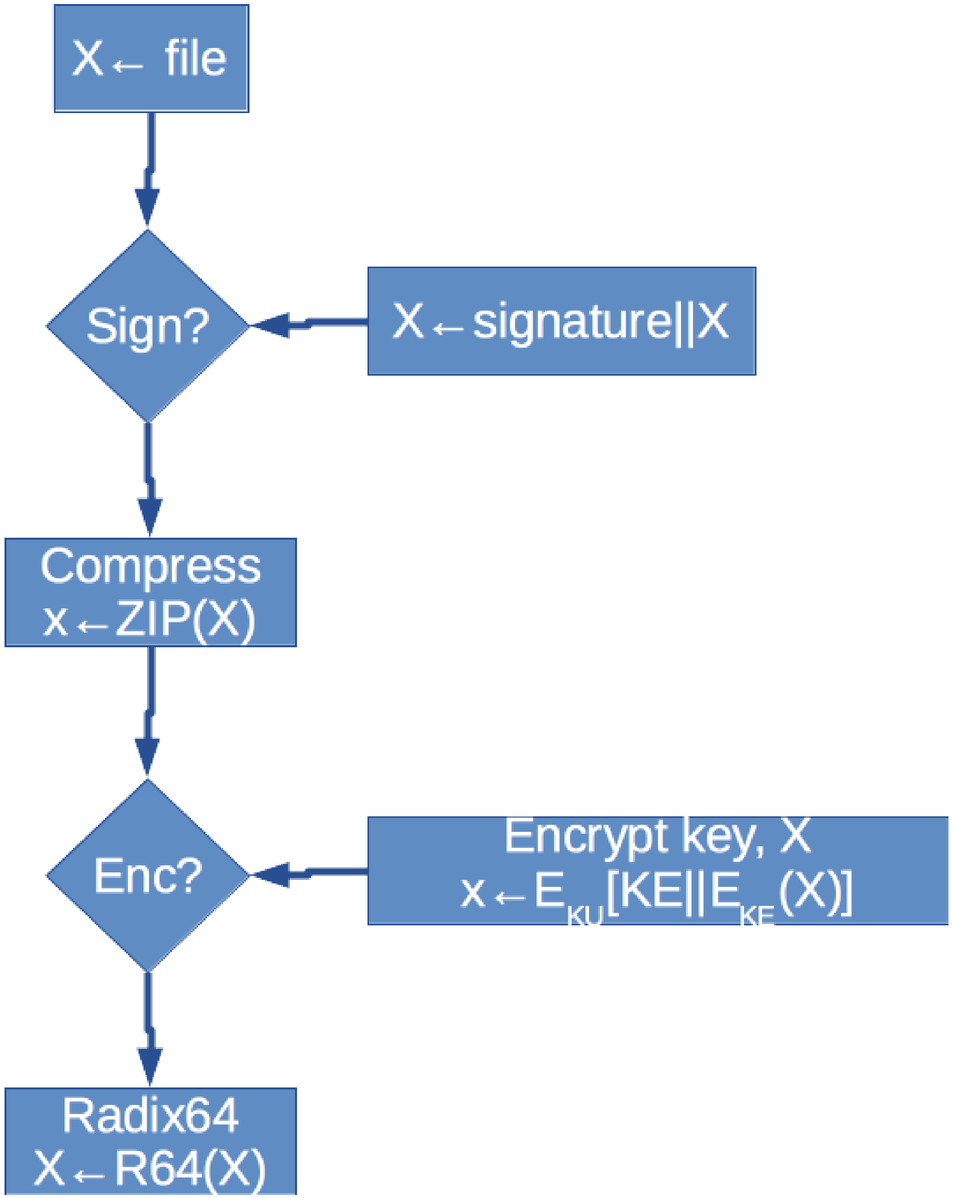
PGP Diagram



Algoritmhs used

- Digital Signature (DSS/SHA or RSA/SHA)
  Uses SHA-1(!) for the message, and uses either DSS, or RSA using sender's private key
- Encryption (CAST, IDEA, or Triple DES, with Diffie-Hellman, or RSA)
  The message is encrypted with CAST, IDEA, or Triple DES, with a session key using Diffie-Hellman, or RSA with the public key of the recipient
- Compression (ZIP)
  For transmission purposes, and for removing some of the statistic properties of the message (before encryption)
- Email compatibility (Radix 64 conversion)
  For compatibility purposes
- Segmentation
  Depending on the application, it may break the message as needed

PGP Diagram



X ← file

Sign?

X ← signature||X

Compress
x ← ZIP(X)

Enc?

Encrypt key, X
$x \leftarrow E_{KU}[KE||E_{KE}(X)]$

Radix64
X ← R64(X)

Algoritmhs used

- Digital Signature (DSS/SHA or RSA/SHA)
  Uses SHA-1(!) for the message, and uses either DSS, or RSA using sender's private key
- Encryption (CAST, IDEA, or Triple DES, with Diffie-Hellman, or RSA)
  The message is encrypted with CAST, IDEA, or Triple DES, with a session key using Diffie-Hellman, or RSA with the public key of the recipient
- Compression (ZIP)
  For transmission purposes, and for removing some of the statistic properties of the message (before encryption)
- Email compatibility (Radix 64 conversion)
  For compatibility purposes
- Segmentation
  Depending on the application, it may break the message as needed

hs used

al Signature (DSS/SHA or RSA/SHA)
SHA-1(!) for the message, and uses
r DSS, or RSA using sender’s private key

yption (CAST, IDEA, or Triple DES, with
-Hellman, or RSA)
message is encrypted with CAST, IDEA,
iple DES, with a session key using
-Hellman, or RSA with the public key of
ecipient

pression (ZIP)
ransmission purposes, and for removing
of the statistic properties of the message
re encryption)

l compatibility (Radix 64 conversion)
ompatibility purposes

entation
nding on the application, it may break the
age as needed

Usage

- Create Key
  - gpg –gen-key
  - gpg –armor –output pubkey.txt –export ’Your Name’
  - gpg –send-keys ’Your Name’ –keyserver hkp://subkeys.pgp.net
- Encrypting / Decrypting
  - gpg –encrypt –recipient ’Your Name’ foo.txt
  - gpg –output foo.txt –decrypt foo.txt.gpg
- Encrypting for Recipient
  - gpg –search-keys ’user1@example.org’ –keyserver hkp://subkeys.pgp.net
  - gpg –import key.asc
  - gpg –list-keys
  - gpg –encrypt –recipient ’user1@example.org’ foo.txt

Usage

- Decry
  - gp
- Signa
  - gp
  - gp

S/SHA or RSA/SHA)
message, and uses
ing sender's private key
EA, or Triple DES, with
A)
ted with CAST, IDEA,
session key using
with the public key of

ses, and for removing
roperties of the message

adix 64 conversion)
oses

lication, it may break the

Usage

- Create Key
  - gpg –gen-key
  - gpg –armor –output pubkey.txt –export 'Your Name'
  - gpg –send-keys 'Your Name' –keyserver hkp://subkeys.pgp.net
- Encrypting / Decrypting
  - gpg –encrypt –recipient 'Your Name' foo.txt
  - gpg –output foo.txt –decrypt foo.txt.gpg
- Encrypting for Recipient
  - gpg –search-keys 'user1@example.org' –keyserver hkp://subkeys.pgp.net
  - gpg –import key.asc
  - gpg –list-keys
  - gpg –encrypt –recipient 'user1@example.org' foo.txt

Usage

- Decrypting
  - gpg –output foo.txt –
- Signatures
  - gpg –verify crucial.ta
  - gpg –armor –detach-s

HA)
es
te key
S, with

IDEA,

key of

oving
nessage

on)

reak the

Usage

- Create Key
  - gpg –gen-key
  - gpg –armor –output pubkey.txt –export 'Your Name'
  - gpg –send-keys 'Your Name' –keyserver hkp://subkeys.pgp.net
- Encrypting / Decrypting
  - gpg –encrypt –recipient 'Your Name' foo.txt
  - gpg –output foo.txt –decrypt foo.txt.gpg
- Encrypting for Recipient
  - gpg –search-keys 'user1@example.org' –keyserver hkp://subkeys.pgp.net
  - gpg –import key.asc
  - gpg –list-keys
  - gpg –encrypt –recipient 'user1@example.org' foo.txt

Usage

- Decrypting
  - gpg –output foo.txt –decrypt foo.txt.gp
- Signatures
  - gpg –verify crucial.tar.gz.asc crucial.tar.
  - gpg –armor –detach-sign your-file.zip

Usage

- Create Key
  - gpg –gen-key
  - gpg –armor –output pubkey.txt –export 'Your Name'
  - gpg –send-keys 'Your Name' –keyserver hkp://subkeys.pgp.net
- Encrypting / Decrypting
  - gpg –encrypt –recipient 'Your Name' foo.txt
  - gpg –output foo.txt –decrypt foo.txt.gpg
- Encrypting for Recipient
  - gpg –search-keys 'user1@example.org' –keyserver hkp://subkeys.pgp.net
  - gpg –import key.asc
  - gpg –list-keys
  - gpg –encrypt –recipient 'user1@example.org' foo.txt

Usage

- Decrypting
  - gpg –output foo.txt –decrypt foo.txt.gpg
- Signatures
  - gpg –verify crucial.tar.gz.asc crucial.tar.gz
  - gpg –armor –detach-sign your-file.zip

e Key

g –gen-key

g –armor –output pubkey.txt –export 'Your me'

g –send-keys 'Your Name' –keyserver p://subkeys.pgp.net

pting / Decrypting

g –encrypt –recipient 'Your Name' foo.txt

g –output foo.txt –decrypt foo.txt.gpg

pting for Recipient

g –search-keys 'user1@example.org' –keyserver p://subkeys.pgp.net

g –import key.asc

g –list-keys

g –encrypt –recipient 'user1@example.org' o.txt

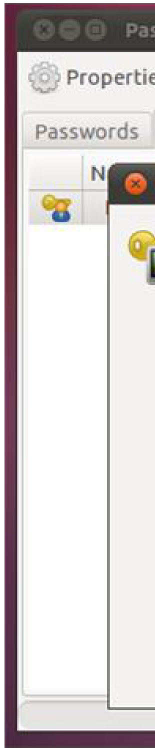
Usage

- Decrypting
  - gpg –output foo.txt –decrypt foo.txt.gpg
- Signatures
  - gpg –verify crucial.tar.gz.asc crucial.tar.gz
  - gpg –armor –detach-sign your-file.zip

Linux to

For Linu

- We u
- It has imple
- It pro
- There



pubkey.txt –export 'Your

Name' –keyserver
et

ng
nt 'Your Name' foo.txt
-decrypt foo.txt.gpg

nt
r1@example.org' –keyserver
et

nt 'user1@example.org'

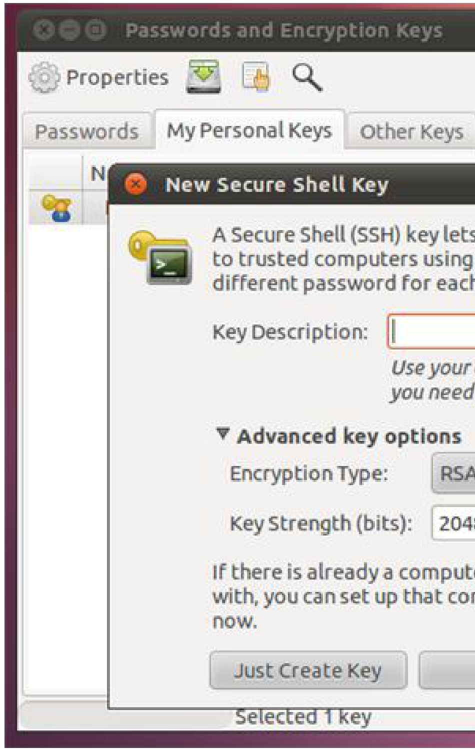
Usage

- Decrypting
  - gpg –output foo.txt –decrypt foo.txt.gpg
- Signatures
  - gpg –verify crucial.tar.gz.asc crucial.tar.gz
  - gpg –armor –detach-sign your-file.zip

Linux tools

For Linux

- We use GPG tools
- It has differences betw
  implementation
- It provides command l
- There are GUI tools, s



A Secure Shell (SSH) key lets
to trusted computers using
different password for each

Key Description:

Use your
you need

Encryption Type: RSA

Key Strength (bits): 204

If there is already a comput
with, you can set up that co
now.

'Your

o.txt
g

keyserver

.org'

Usage

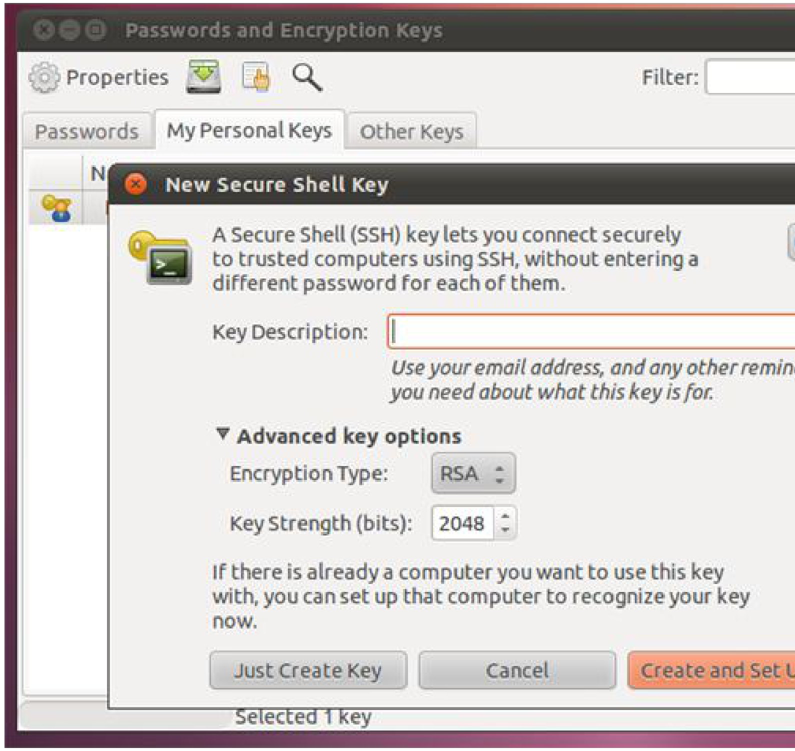
- Decrypting
  - gpg –output foo.txt –decrypt foo.txt.gpg

- Signatures
  - gpg –verify crucial.tar.gz.asc crucial.tar.gz
  - gpg –armor –detach-sign your-file.zip

Linux tools

For Linux

- We use GPG tools
- It has differences between the PGP offi implementation
- It provides command line interface
- There are GUI tools, such as Seahorse



A Secure Shell (SSH) key lets you connect securely to trusted computers using SSH, without entering a different password for each of them.

Key Description:

Use your email address, and any other remin you need about what this key is for.

▼ **Advanced key options**

Encryption Type: RSA

Key Strength (bits): 2048

If there is already a computer you want to use this key with, you can set up that computer to recognize your key now.

Selected 1 key

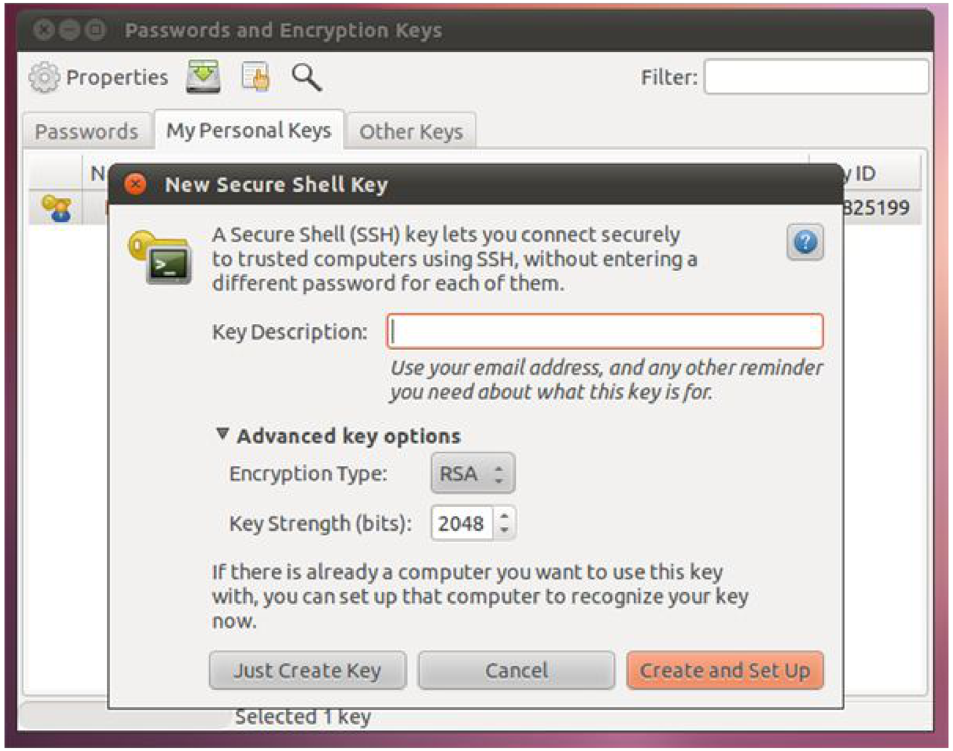
Usage

- Decrypting
  - gpg –output foo.txt –decrypt foo.txt.gpg
- Signatures
  - gpg –verify crucial.tar.gz.asc crucial.tar.gz
  - gpg –armor –detach-sign your-file.zip

Linux tools

For Linux

- We use GPG tools
- It has differences between the PGP official implementation
- It provides command line interface
- There are GUI tools, such as Seahorse



A Secure Shell (SSH) key lets you connect securely to trusted computers using SSH, without entering a different password for each of them.

Key Description:

Use your email address, and any other reminder you need about what this key is for.

▼ **Advanced key options**

Encryption Type: RSA

Key Strength (bits): 2048

If there is already a computer you want to use this key with, you can set up that computer to recognize your key now.

ypting

g –output foo.txt –decrypt foo.txt.gpg

tures

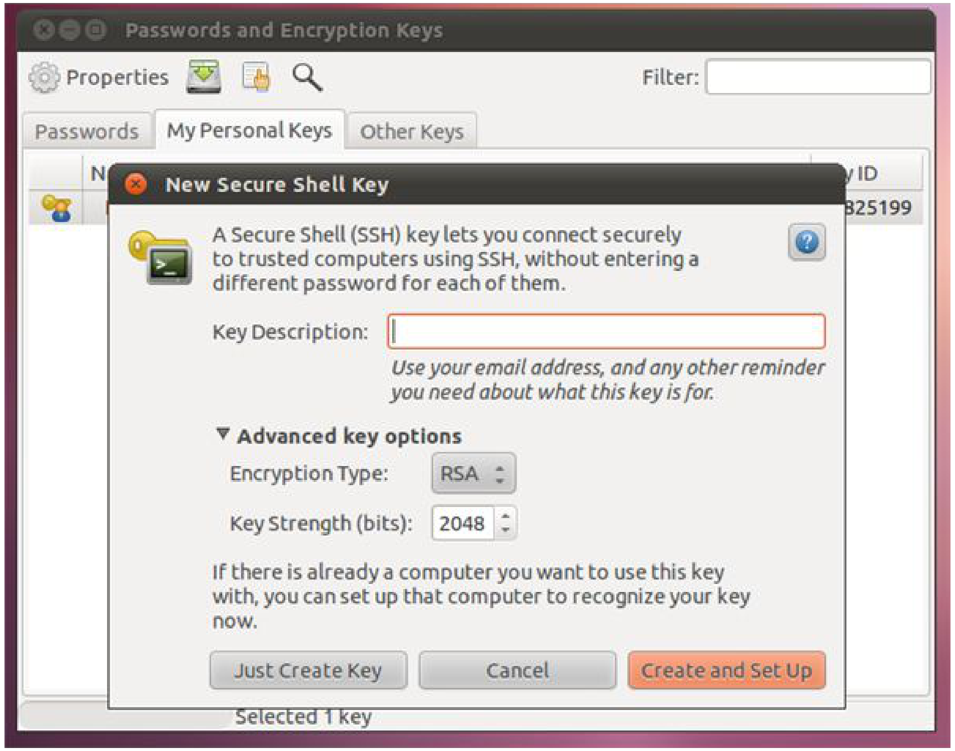
g –verify crucial.tar.gz.asc crucial.tar.gz

g –armor –detach-sign your-file.zip

Linux tools

For Linux

- We use GPG tools
- It has differences between the PGP official implementation
- It provides command line interface
- There are GUI tools, such as Seahorse



A Secure Shell (SSH) key lets you connect securely to trusted computers using SSH, without entering a different password for each of them.

Key Description:

Use your email address, and any other reminder you need about what this key is for.

▼ Advanced key options

Encryption Type: RSA

Key Strength (bits): 2048

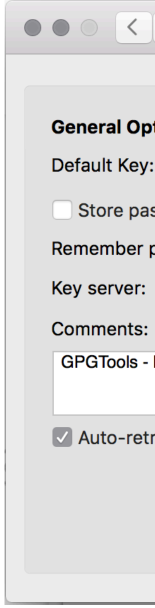
If there is already a computer you want to use this key with, you can set up that computer to recognize your key now.

Selected 1 key

OSX Too

GNU to

- Work
- GPG
- GPG
- GPG
- MacG



General Opt

Default Key:

□ Store pas

Remember p

Key server:

Comments:

GPGTools -

☑ Auto-retr

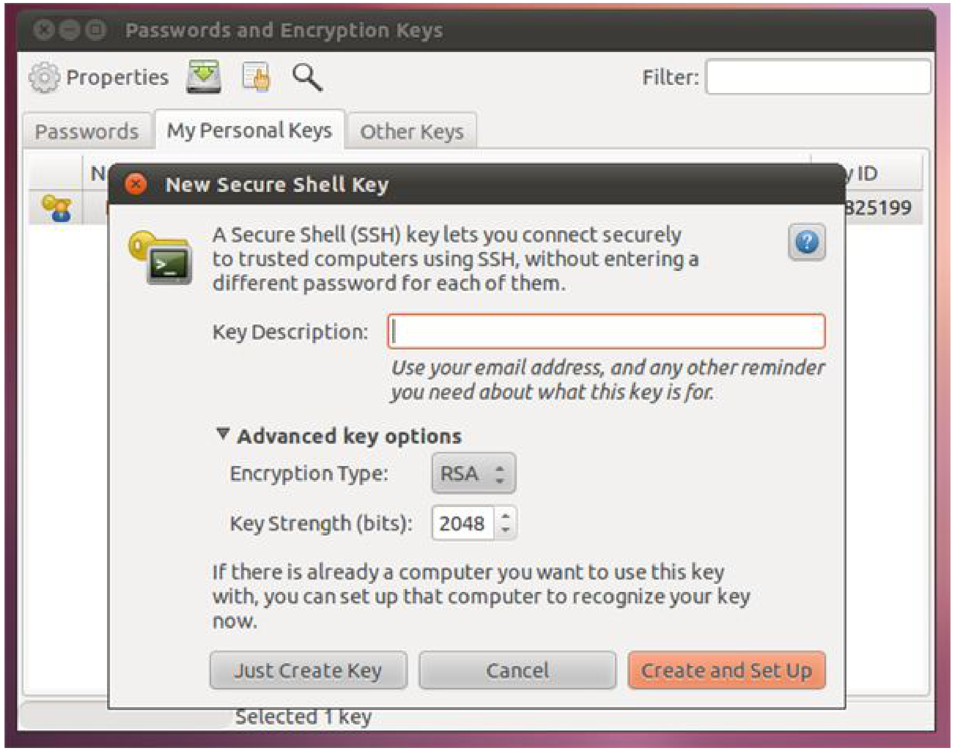
-decrypt foo.txt.gpg

r.gz.asc crucial.tar.gz
sign your-file.zip

Linux tools

For Linux

- We use GPG tools
- It has differences between the PGP official implementation
- It provides command line interface
- There are GUI tools, such as Seahorse


Passwords and Encryption Keys

Properties Filter:

Passwords | My Personal Keys | Other Keys

N ... y ID

325199

New Secure Shell Key

A Secure Shell (SSH) key lets you connect securely to trusted computers using SSH, without entering a different password for each of them.

Key Description:

Use your email address, and any other reminder you need about what this key is for.

▼ Advanced key options

Encryption Type: RSA

Key Strength (bits): 2048

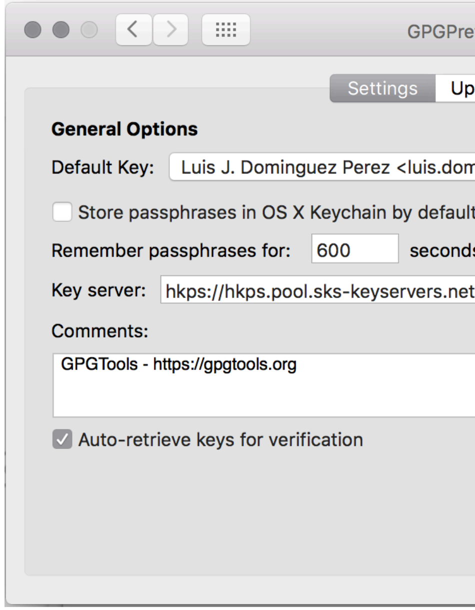
If there is already a computer you want to use this key with, you can set up that computer to recognize your key now.

Selected 1 key

OSX Tools

GNU tools

- Works for El Capitan (
- GPG for Mail (Apple e
- GPG keychain
- GPG services
- MacGPG


GPGPre

General Options

Default Key: Luis J. Dominguez Perez <luis.dom

□ Store passphrases in OS X Keychain by default

Remember passphrases for: 600 seconds

Key server: hkps://hkps.pool.sks-keyservers.net

Comments:

GPGTools - https://gpgtools.org

☑ Auto-retrieve keys for verification

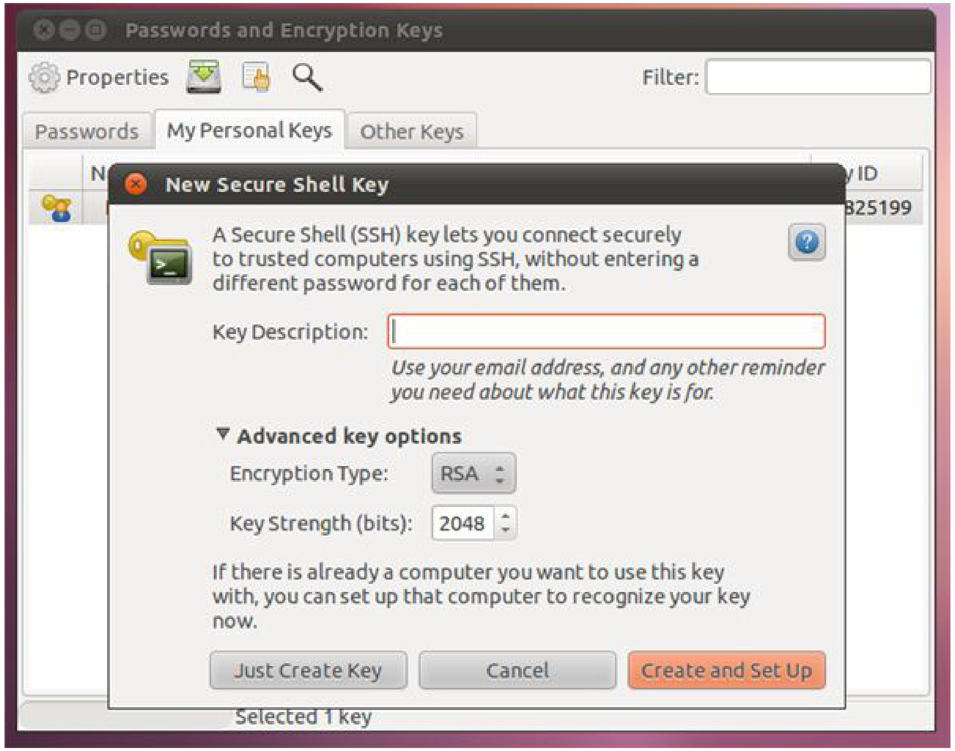
g

gz

Linux tools

For Linux

- We use GPG tools
- It has differences between the PGP official implementation
- It provides command line interface
- There are GUI tools, such as Seahorse



New Secure Shell Key

A Secure Shell (SSH) key lets you connect securely to trusted computers using SSH, without entering a different password for each of them.

Key Description:

Use your email address, and any other reminder you need about what this key is for.

▼ Advanced key options

Encryption Type: RSA

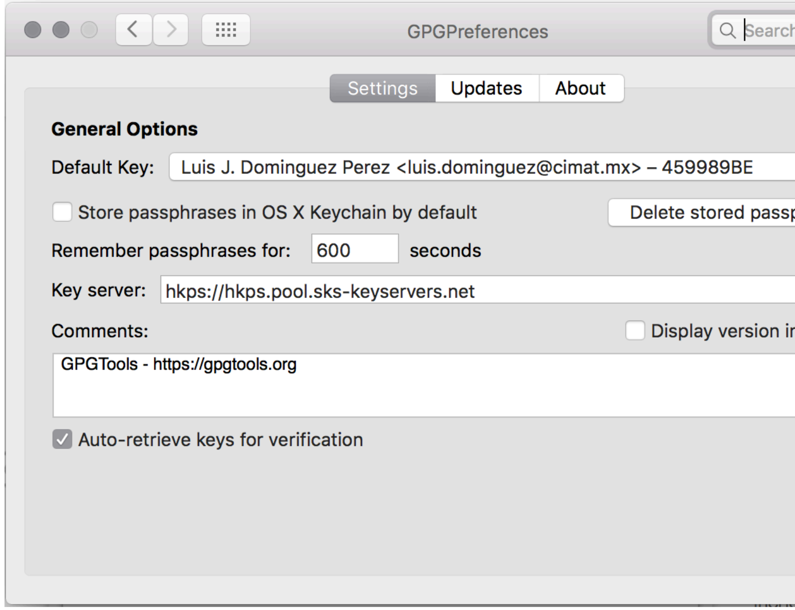
Key Strength (bits): 2048

If there is already a computer you want to use this key with, you can set up that computer to recognize your key now.

OSX Tools

GNU tools

- Works for El Capitan (OS X 10.11
- GPG for Mail (Apple email)
- GPG keychain
- GPG services
- MacGPG



General Options

Default Key: Luis J. Dominguez Perez <luis.dominguez@cimat.mx> – 459989BE

□ Store passphrases in OS X Keychain by default

Remember passphrases for: 600 seconds

Key server: hkps://hkps.pool.sks-keyservers.net

Comments:

□ Display version ir

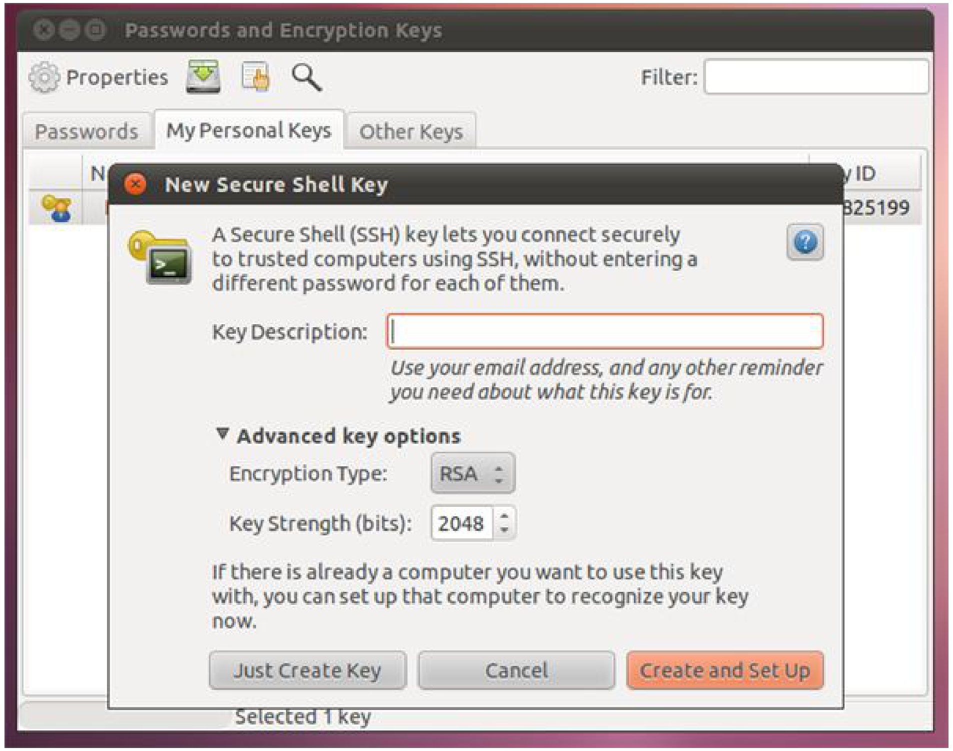
GPGTools - https://gpgtools.org

☑ Auto-retrieve keys for verification

Linux tools

For Linux

- We use GPG tools
- It has differences between the PGP official implementation
- It provides command line interface
- There are GUI tools, such as Seahorse



A Secure Shell (SSH) key lets you connect securely to trusted computers using SSH, without entering a different password for each of them.

Key Description:

Use your email address, and any other reminder you need about what this key is for.

▼ **Advanced key options**

Encryption Type: RSA

Key Strength (bits): 2048

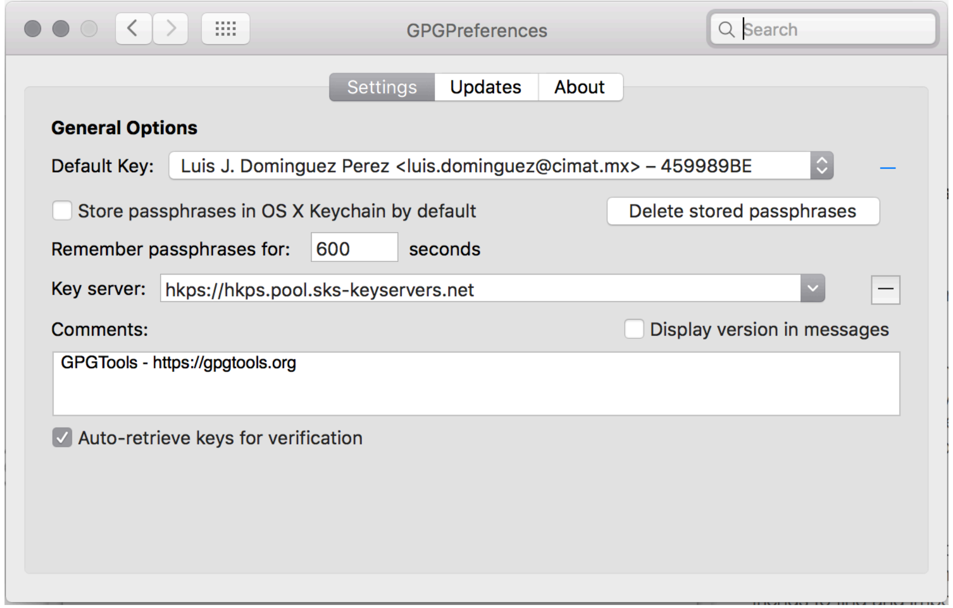
If there is already a computer you want to use this key with, you can set up that computer to recognize your key now.

Selected 1 key

OSX Tools

GNU tools

- Works for El Capitan (OS X 10.11
- GPG for Mail (Apple email)
- GPG keychain
- GPG services
- MacGPG



General Options

Default Key: Luis J. Dominguez Perez <luis.dominguez@cimat.mx> – 459989BE

□ Store passphrases in OS X Keychain by default

Remember passphrases for: 600 seconds

Key server: hkps://hkps.pool.sks-keyservers.net

Comments: □ Display version in messages

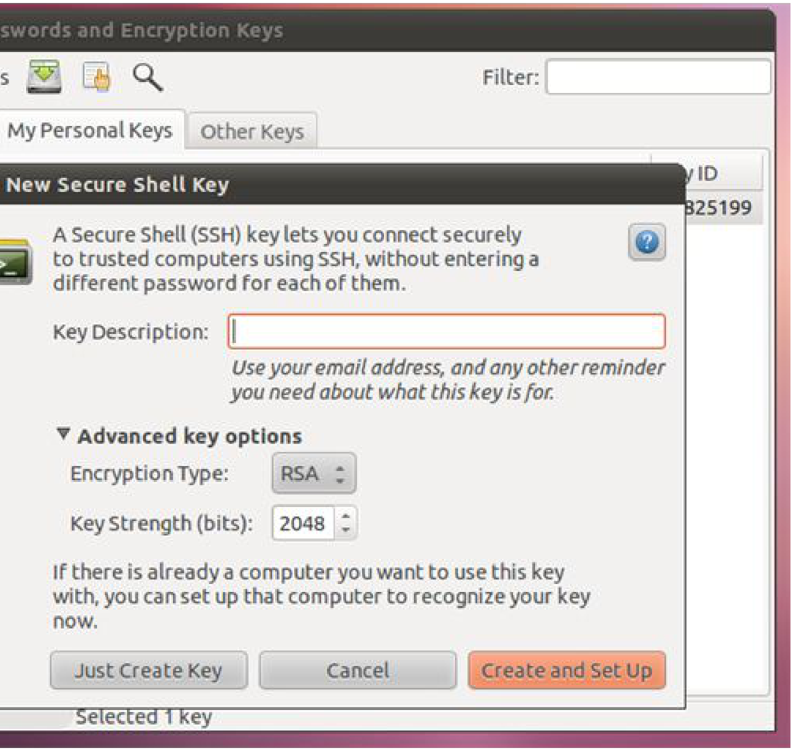
GPGTools - https://gpgtools.org

☑ Auto-retrieve keys for verification

ols

x

- se GPG tools
- s differences between the PGP official mentation
- vides command line interface
- e are GUI tools, such as Seahorse



swords and Encryption Keys

s Filter:

My Personal Keys Other Keys

y ID

825199

New Secure Shell Key

A Secure Shell (SSH) key lets you connect securely to trusted computers using SSH, without entering a different password for each of them.

Key Description:

Use your email address, and any other reminder you need about what this key is for.

▼ **Advanced key options**

Encryption Type: RSA

Key Strength (bits): 2048

If there is already a computer you want to use this key with, you can set up that computer to recognize your key now.

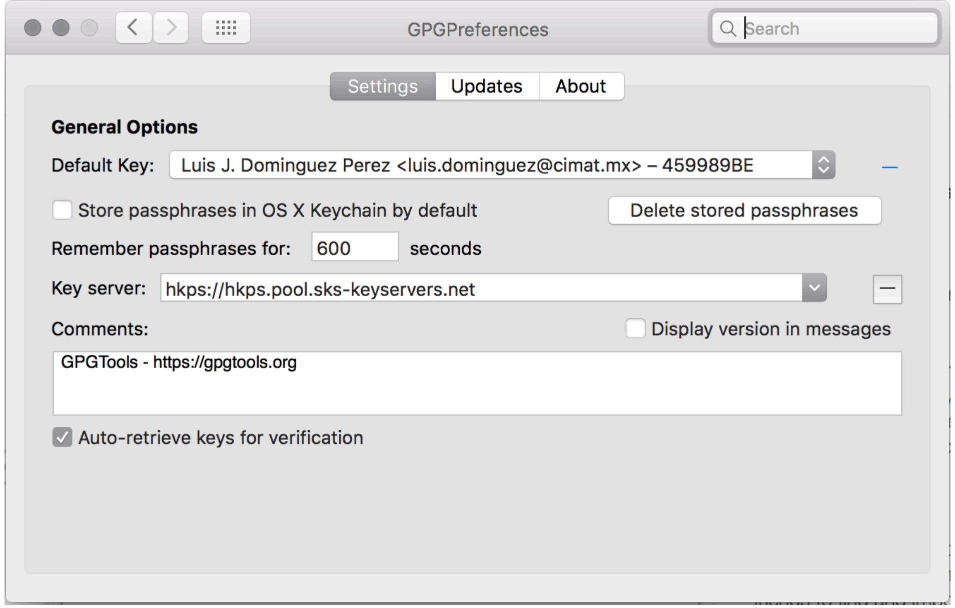
Just Create Key | Cancel | Create and Set Up

Selected 1 key

OSX Tools

GNU tools

- ▶ Works for El Capitan (OS X 10.11
- ▶ GPG for Mail (Apple email)
- ▶ GPG keychain
- ▶ GPG services
- ▶ MacGPG



GPGPreferences

Search

Settings | Updates | About

General Options

Default Key: Luis J. Dominguez Perez <luis.dominguez@cimat.mx> – 459989BE

□ Store passphrases in OS X Keychain by default

Delete stored passphrases

Remember passphrases for: 600 seconds

Key server: hkps://hkps.pool.sks-keyservers.net

Comments: □ Display version in messages

GPGTools - https://gpgtools.org

☑ Auto-retrieve keys for verification

Windows

For Win

- ▶ PGP
- ▶ GPG
- ▶ GNU

P [redacted]

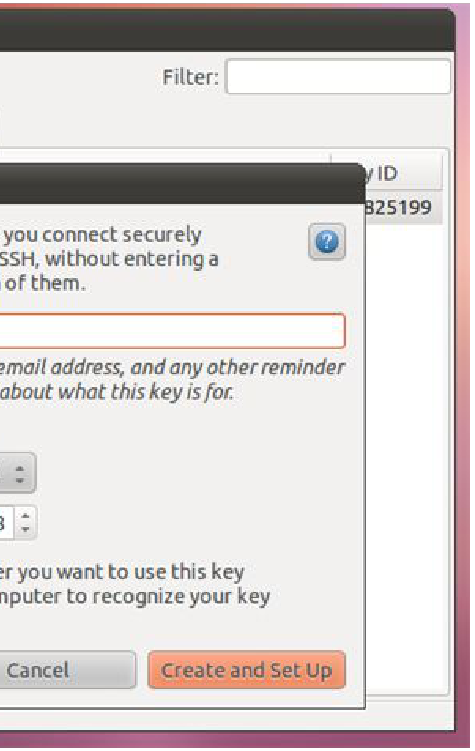
P [redacted]

P E8918B35

een the PGP official

ine interface

uch as Seahorse



you connect securely
SSH, without entering a
of them.

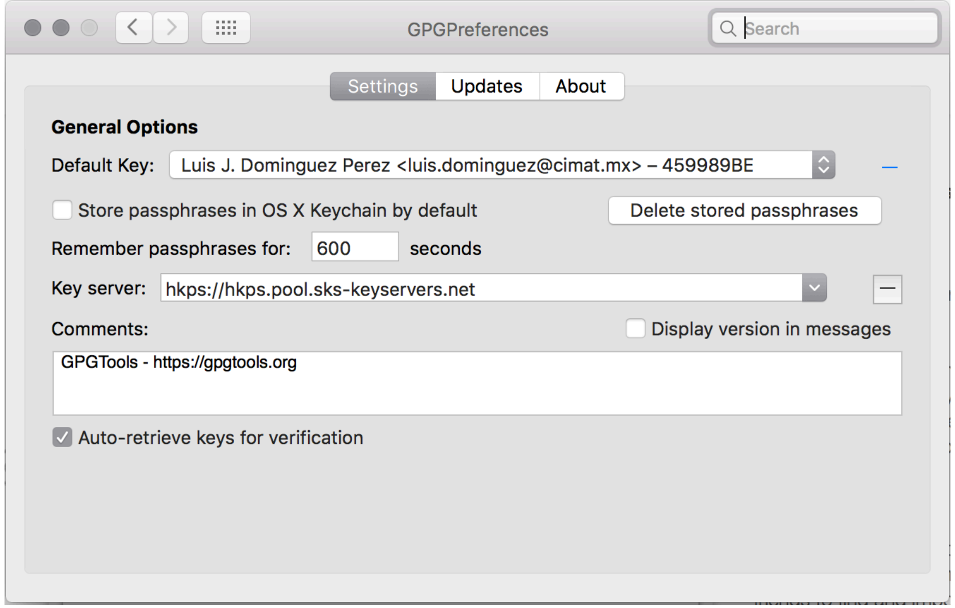
email address, and any other reminder
about what this key is for.

er you want to use this key
nputer to recognize your key

OSX Tools

GNU tools

- Works for El Capitan (OS X 10.11
- GPG for Mail (Apple email)
- GPG keychain
- GPG services
- MacGPG



General Options

Default Key: Luis J. Dominguez Perez <luis.dominguez@cimat.mx> – 459989BE

□ Store passphrases in OS X Keychain by default

Remember passphrases for: 600 seconds

Key server: hkps://hkps.pool.sks-keyservers.net

Comments: □ Display version in messages

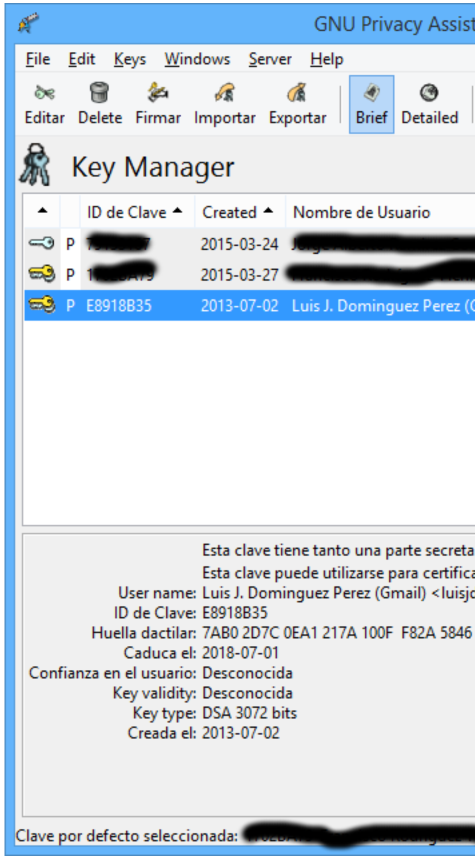
GPGTools - https://gpgtools.org

☑ Auto-retrieve keys for verification

Windows Tools, etc.

For Windows, you have

- PGP by Symantec
- GPG 4 Win (for Outlo
- GNU tools with Cygw

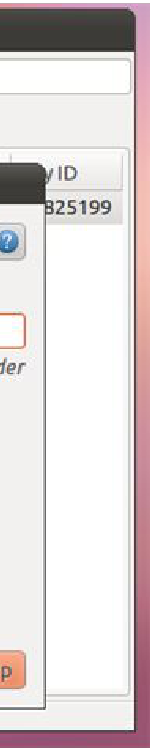


	ID de Clave	Created	Nombre de Usuario
P	[redacted]	2015-03-24	[redacted]
P	[redacted]	2015-03-27	[redacted]
P	E8918B35	2013-07-02	Luis J. Dominguez Perez ((

Esta clave tiene tanto una parte secreta
Esta clave puede utilizarse para certifica
User name: Luis J. Dominguez Perez (Gmail) <luisjo
ID de Clave: E8918B35
Huella dactilar: 7AB0 2D7C 0EA1 217A 100F F82A 5846
Caduca el: 2018-07-01
Confianza en el usuario: Desconocida
Key validity: Desconocida
Key type: DSA 3072 bits
Creada el: 2013-07-02

Clave por defecto seleccionada: [redacted]

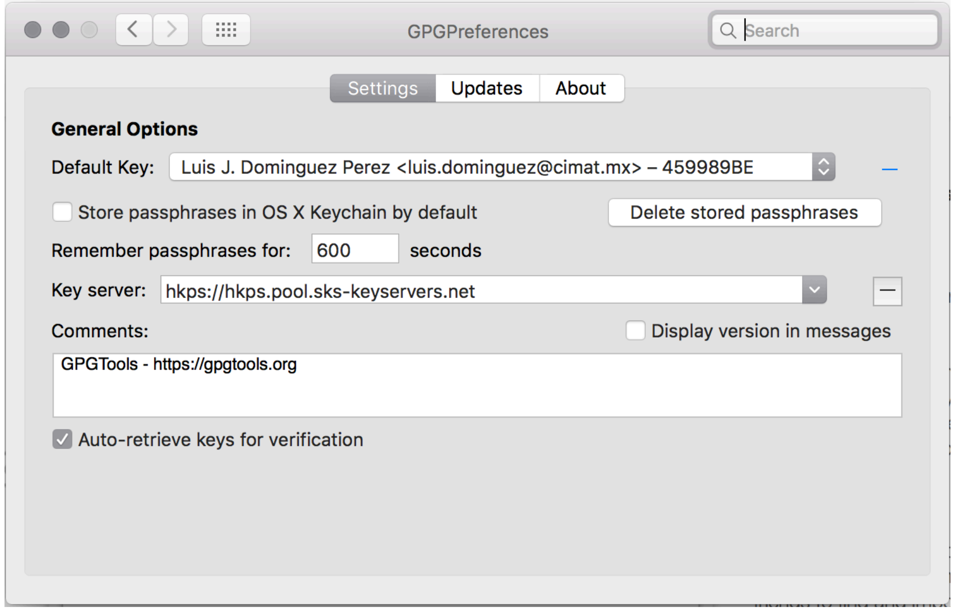
cial



OSX Tools

GNU tools

- Works for El Capitan (OS X 10.11
- GPG for Mail (Apple email)
- GPG keychain
- GPG services
- MacGPG



General Options

Default Key: Luis J. Dominguez Perez <luis.dominguez@cimat.mx> – 459989BE

□ Store passphrases in OS X Keychain by default

Delete stored passphrases

Remember passphrases for: 600 seconds

Key server: hkps://hkps.pool.sks-keyservers.net

Comments: □ Display version in messages

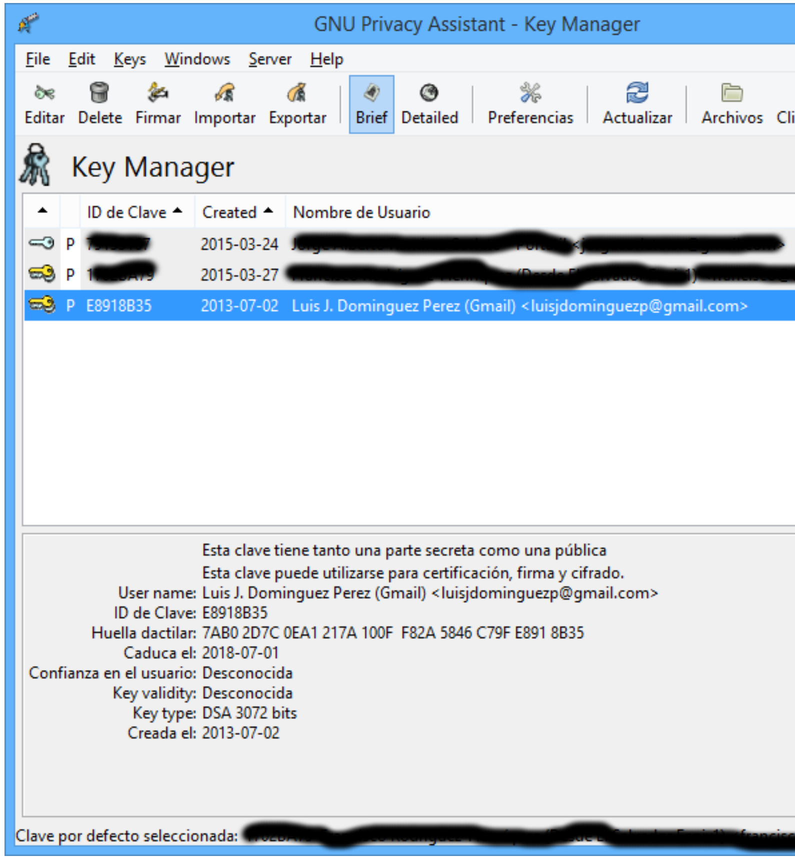
GPGTools - https://gpgtools.org

☑ Auto-retrieve keys for verification

Windows Tools, etc.

For Windows, you have

- PGP by Symantec
- GPG 4 Win (for Outlook)
- GNU tools with Cygwin



		ID de Clave	Created	Nombre de Usuario
	P	[redacted]	2015-03-24	[redacted]
	P	[redacted]	2015-03-27	[redacted]
	P	E8918B35	2013-07-02	Luis J. Dominguez Perez (Gmail) <luisjdominguezp@gmail.com>

Esta clave tiene tanto una parte secreta como una pública

Esta clave puede utilizarse para certificación, firma y cifrado.

User name: Luis J. Dominguez Perez (Gmail) <luisjdominguezp@gmail.com>

ID de Clave: E8918B35

Huella dactilar: 7AB0 2D7C 0EA1 217A 100F F82A 5846 C79F E891 8B35

Caduca el: 2018-07-01

Confianza en el usuario: Desconocida

Key validity: Desconocida

Key type: DSA 3072 bits

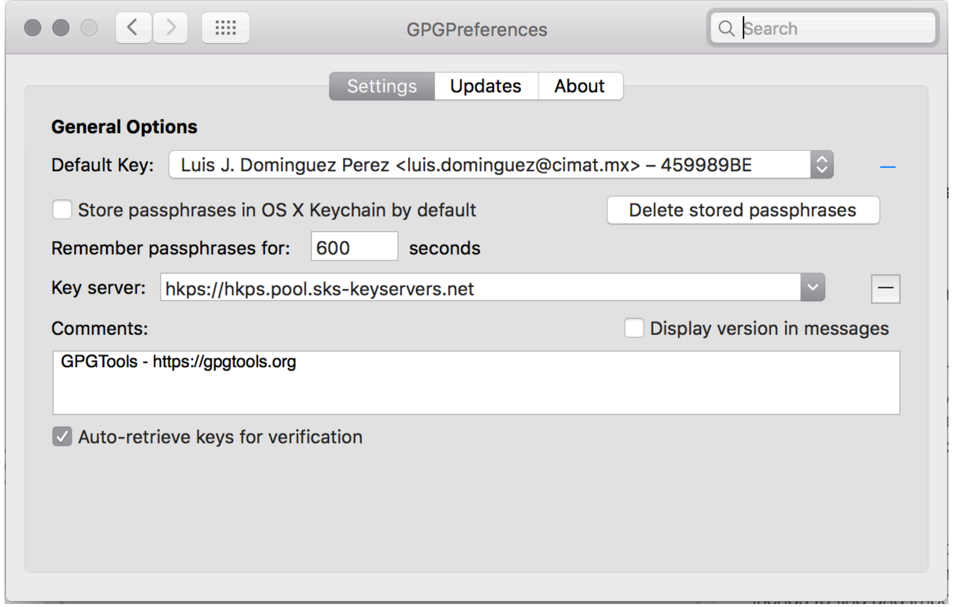
Creada el: 2013-07-02

Clave por defecto seleccionada: [redacted]

OSX Tools

GNU tools

- Works for El Capitan (OS X 10.11
- GPG for Mail (Apple email)
- GPG keychain
- GPG services
- MacGPG



General Options

Default Key: Luis J. Dominguez Perez <luis.dominguez@cimat.mx> – 459989BE

□ Store passphrases in OS X Keychain by default

Delete stored passphrases

Remember passphrases for: 600 seconds

Key server: hkps://hkps.pool.sks-keyservers.net

Comments: □ Display version in messages

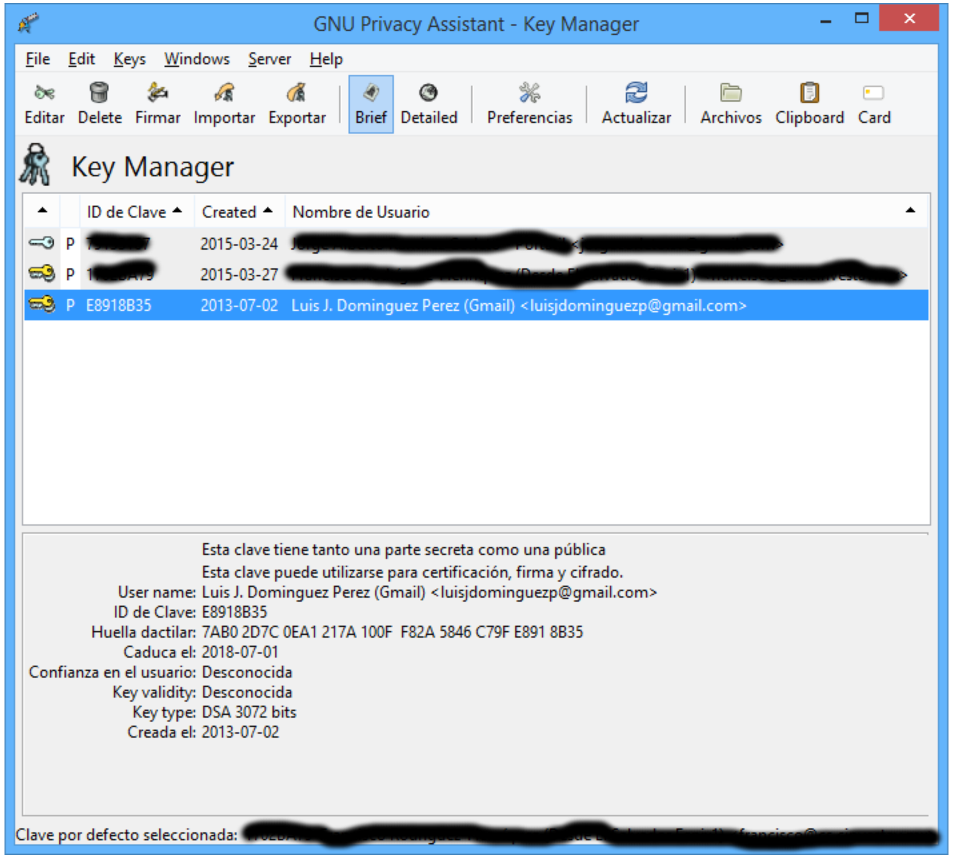
GPGTools - https://gpgtools.org

☑ Auto-retrieve keys for verification

Windows Tools, etc.

For Windows, you have

- PGP by Symantec
- GPG 4 Win (for Outlook)
- GNU tools with Cygwin



	ID de Clave	Created	Nombre de Usuario
P	[redacted]	2015-03-24	[redacted]
P	[redacted]	2015-03-27	[redacted]
P	E8918B35	2013-07-02	Luis J. Dominguez Perez (Gmail) <luisjdominguezp@gmail.com>

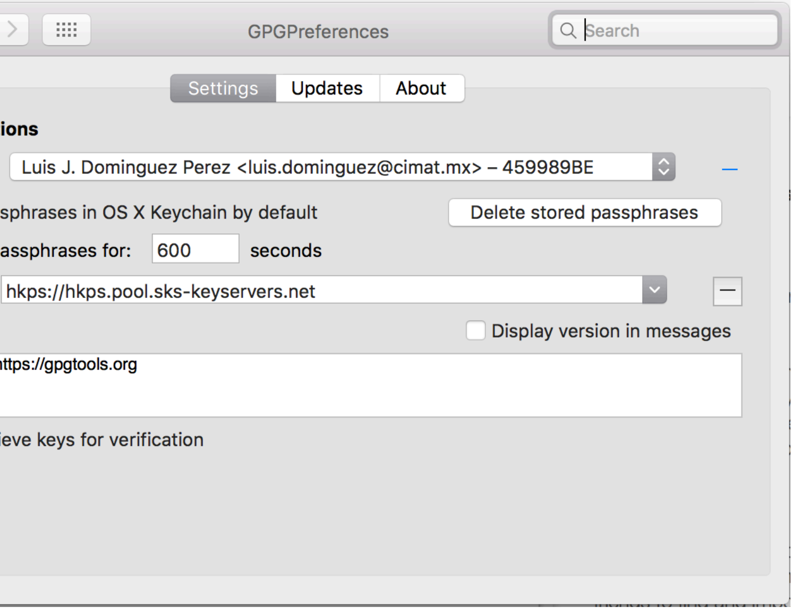
Esta clave tiene tanto una parte secreta como una pública
Esta clave puede utilizarse para certificación, firma y cifrado.

```
             User name: Luis J. Dominguez Perez (Gmail) <luisjdominguezp@gmail.com>
           ID de Clave: E8918B35
       Huella dactilar: 7AB0 2D7C 0EA1 217A 100F  F82A 5846 C79F E891 8B35
             Caduca el: 2018-07-01
Confianza en el usuario: Desconocida
          Key validity: Desconocida
              Key type: DSA 3072 bits
             Creada el: 2013-07-02
```

Clave por defecto seleccionada: [redacted]

ols

- ols
- s for El Capitan (OS X 10.11
- for Mail (Apple email)
- keychain
- services
- GPG



ions

Luis J. Dominguez Perez <luis.dominguez@cimat.mx> – 459989BE

sphrases in OS X Keychain by default

Delete stored passphrases

assphrases for: 600 seconds

hkps://hkps.pool.sks-keyservers.net

□ Display version in messages

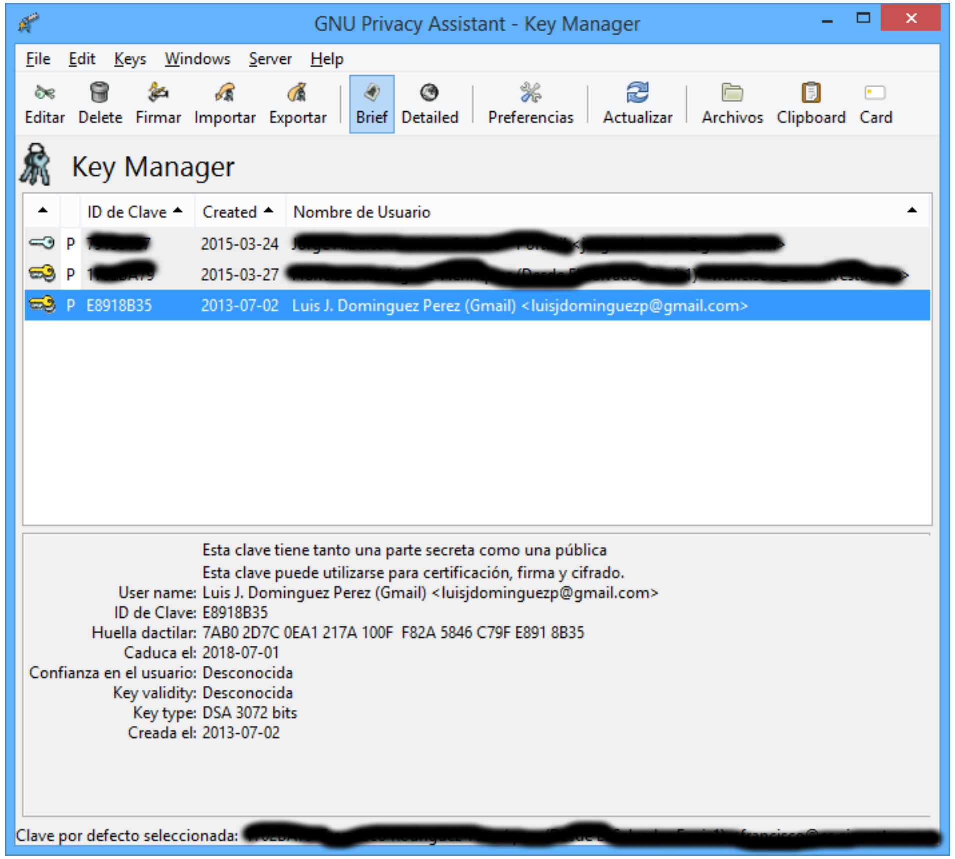
ttps://gpgtools.org

ieve keys for verification

Windows Tools, etc.

For Windows, you have

- PGP by Symantec
- GPG 4 Win (for Outlook)
- GNU tools with Cygwin



GNU Privacy Assistant - Key Manager
File Edit Keys Windows Server Help
Editar Delete Firmar Importar Exportar Brief Detailed Preferencias Actualizar Archivos Clipboard Card
Key Manager

	ID de Clave	Created	Nombre de Usuario
P	[redacted]	2015-03-24	[redacted]
P	[redacted]	2015-03-27	[redacted]
P	E8918B35	2013-07-02	Luis J. Dominguez Perez (Gmail) <luisjdominguezp@gmail.com>

Esta clave tiene tanto una parte secreta como una pública
Esta clave puede utilizarse para certificación, firma y cifrado.
User name: Luis J. Dominguez Perez (Gmail) <luisjdominguezp@gmail.com>
ID de Clave: E8918B35
Huella dactilar: 7AB0 2D7C 0EA1 217A 100F F82A 5846 C79F E891 8B35
Caduca el: 2018-07-01
Confianza en el usuario: Desconocida
Key validity: Desconocida
Key type: DSA 3072 bits
Creada el: 2013-07-02

Clave por defecto seleccionada: [redacted]

Mailvelo

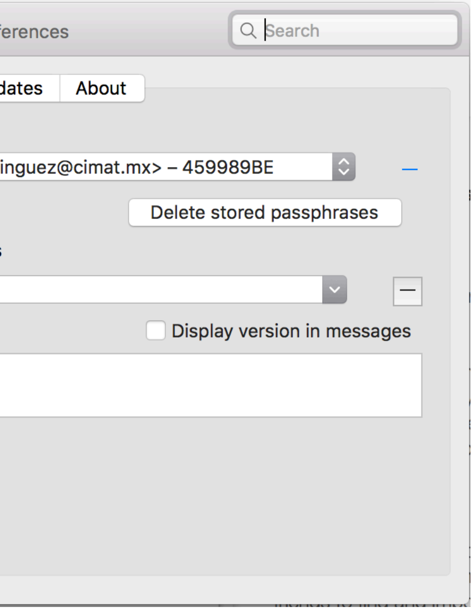
- Is a C
  encry
- Uses:
  Boots

Support

- Key I
- Key C
- Key I
- Encry

The key
provides
am not

(OS X 10.11
email)



inguez@cimat.mx> – 459989BE

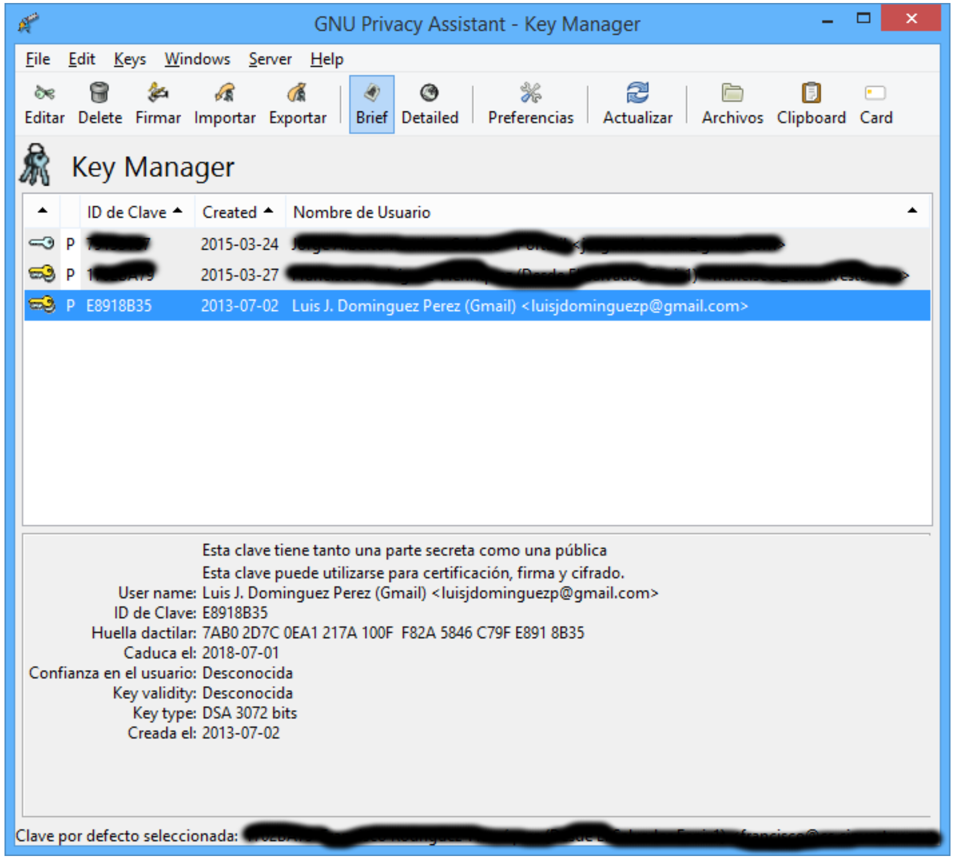
Delete stored passphrases

□ Display version in messages

Windows Tools, etc.

For Windows, you have

- PGP by Symantec
- GPG 4 Win (for Outlook)
- GNU tools with Cygwin


GNU Privacy Assistant - Key Manager
File Edit Keys Windows Server Help
Editar Delete Firmar Importar Exportar Brief Detailed Preferencias Actualizar Archivos Clipboard Card
Key Manager

		ID de Clave	Created	Nombre de Usuario
	P	[redacted]	2015-03-24	[redacted]
	P	[redacted]	2015-03-27	[redacted]
	P	E8918B35	2013-07-02	Luis J. Dominguez Perez (Gmail) <luisjdominguezp@gmail.com>

Esta clave tiene tanto una parte secreta como una pública
Esta clave puede utilizarse para certificación, firma y cifrado.
User name: Luis J. Dominguez Perez (Gmail) <luisjdominguezp@gmail.com>
ID de Clave: E8918B35
Huella dactilar: 7AB0 2D7C 0EA1 217A 100F F82A 5846 C79F E891 8B35
Caduca el: 2018-07-01
Confianza en el usuario: Desconocida
Key validity: Desconocida
Key type: DSA 3072 bits
Creada el: 2013-07-02

Clave por defecto seleccionada: [redacted]

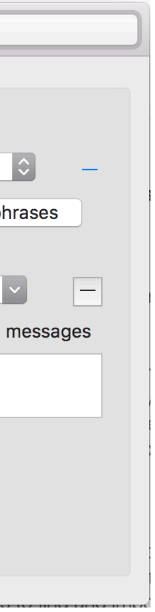
Mailvelope

- Is a Chrome, and Firef
encryption in your web
- Uses: OpenPGP.js, em
Bootstrap, jQuery, Ox

Supports:

- Key Management
- Key Generation
- Key Import/Export
- Encrypt/Decrypt, and

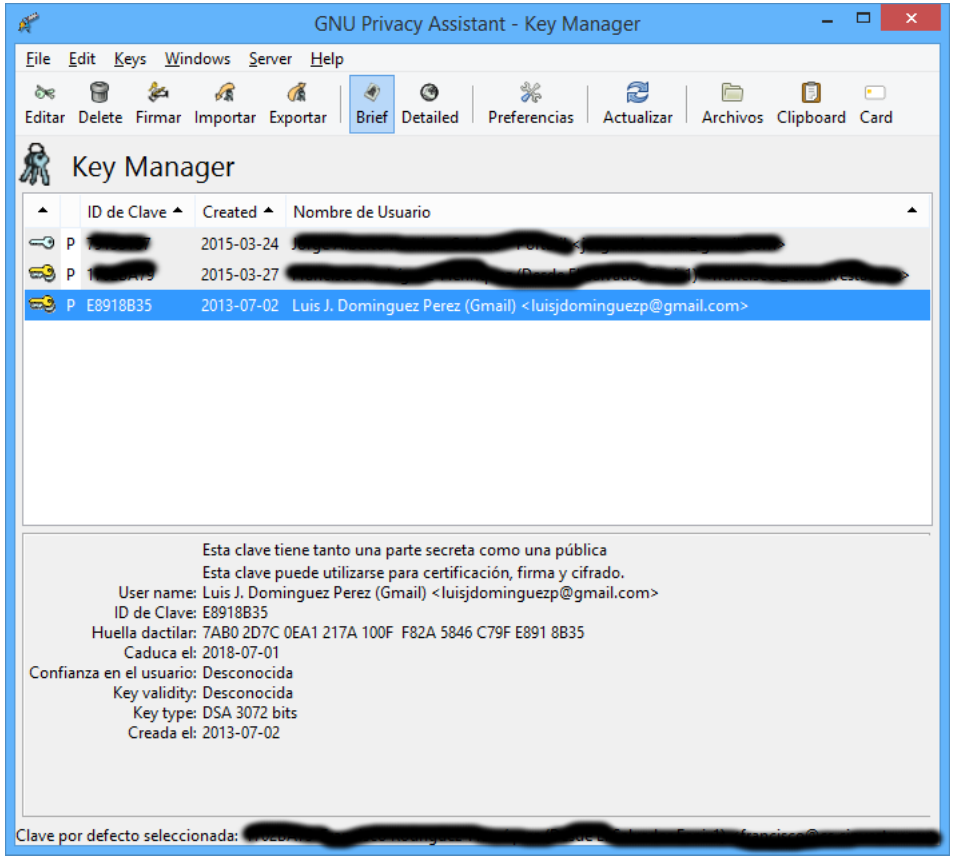
The keys are stored in yo
provides a strong encrypt
am not sure if also for th



Windows Tools, etc.

For Windows, you have

- PGP by Symantec
- GPG 4 Win (for Outlook)
- GNU tools with Cygwin


GNU Privacy Assistant - Key Manager
File Edit Keys Windows Server Help
Editar Delete Firmar Importar Exportar Brief Detailed Preferencias Actualizar Archivos Clipboard Card
Key Manager

	ID de Clave	Created	Nombre de Usuario
P	[redacted]	2015-03-24	[redacted]
P	[redacted]	2015-03-27	[redacted]
P	E8918B35	2013-07-02	Luis J. Dominguez Perez (Gmail) <luisjdominguezp@gmail.com>

Esta clave tiene tanto una parte secreta como una pública
Esta clave puede utilizarse para certificación, firma y cifrado.
User name: Luis J. Dominguez Perez (Gmail) <luisjdominguezp@gmail.com>
ID de Clave: E8918B35
Huella dactilar: 7AB0 2D7C 0EA1 217A 100F F82A 5846 C79F E891 8B35
Caduca el: 2018-07-01
Confianza en el usuario: Desconocida
Key validity: Desconocida
Key type: DSA 3072 bits
Creada el: 2013-07-02

Clave por defecto seleccionada: [redacted]

Mailvelope

- Is a Chrome, and Firefox plug-in for usi encryption in your webmail clients.
- Uses: OpenPGP.js, email.js, DOMpurif Bootstrap, jQuery, Oxygen icons

Supports:

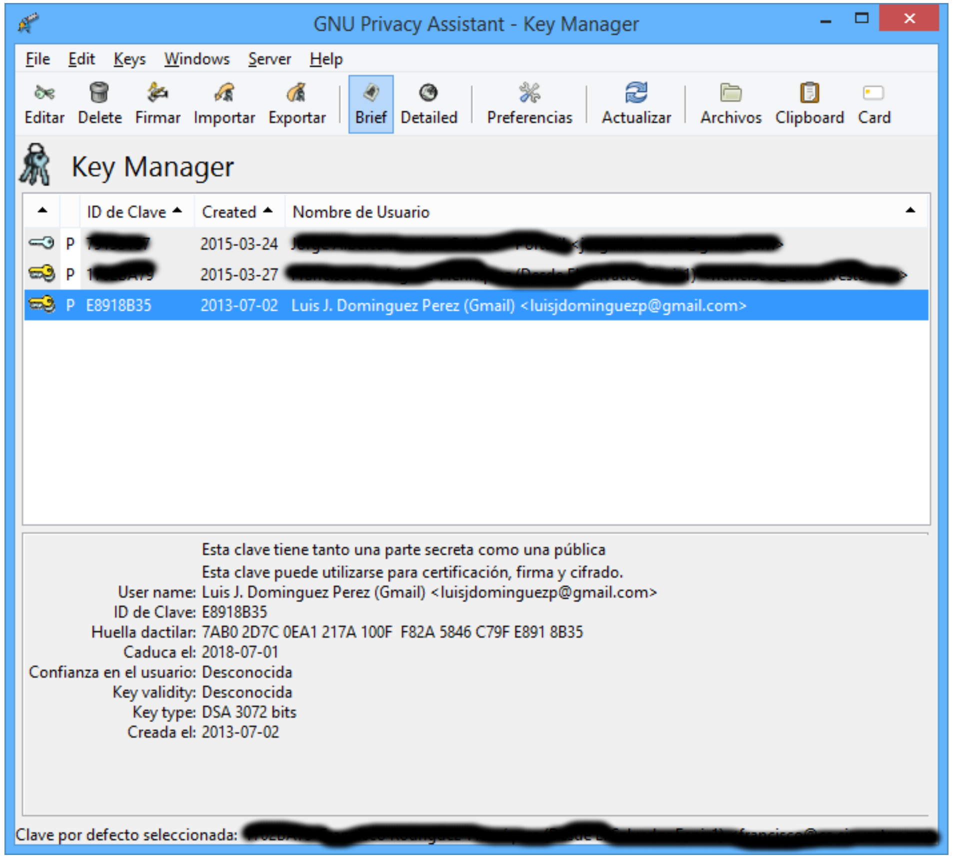
- Key Management
- Key Generation
- Key Import/Export
- Encrypt/Decrypt, and signing

The keys are stored in your profile. Only F provides a strong encryption for password, am not sure if also for the whole profile.

Windows Tools, etc.

For Windows, you have

- PGP by Symantec
- GPG 4 Win (for Outlook)
- GNU tools with Cygwin


GNU Privacy Assistant - Key Manager
File Edit Keys Windows Server Help
Editar Delete Firmar Importar Exportar Brief Detailed Preferencias Actualizar Archivos Clipboard Card
Key Manager

	ID de Clave	Created	Nombre de Usuario
P	[redacted]	2015-03-24	[redacted]
P	[redacted]	2015-03-27	[redacted]
P	E8918B35	2013-07-02	Luis J. Dominguez Perez (Gmail) <luisjdominguezp@gmail.com>

Esta clave tiene tanto una parte secreta como una pública
Esta clave puede utilizarse para certificación, firma y cifrado.
User name: Luis J. Dominguez Perez (Gmail) <luisjdominguezp@gmail.com>
ID de Clave: E8918B35
Huella dactilar: 7AB0 2D7C 0EA1 217A 100F F82A 5846 C79F E891 8B35
Caduca el: 2018-07-01
Confianza en el usuario: Desconocida
Key validity: Desconocida
Key type: DSA 3072 bits
Creada el: 2013-07-02

Clave por defecto seleccionada: [redacted]

Mailvelope

- Is a Chrome, and Firefox plug-in for using PGP encryption in your webmail clients.
- Uses: OpenPGP.js, email.js, DOMpurify, Bootstrap, jQuery, Oxygen icons

Supports:

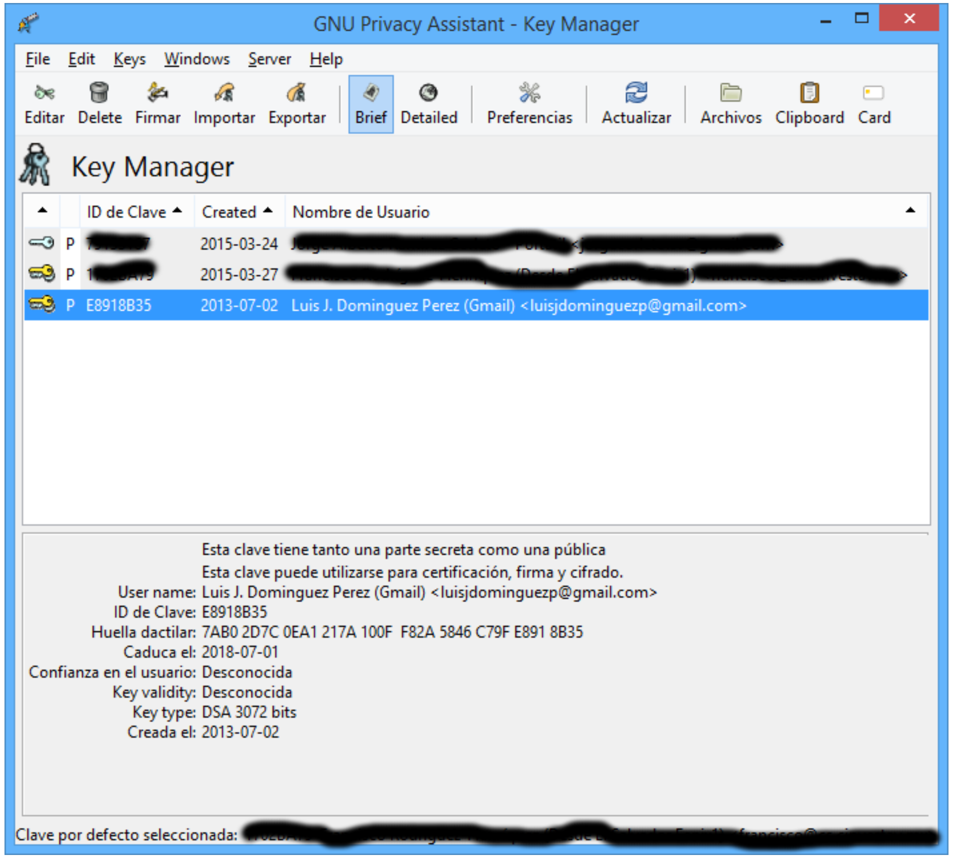
- Key Management
- Key Generation
- Key Import/Export
- Encrypt/Decrypt, and signing

The keys are stored in your profile. Only Firefox provides a strong encryption for password, but I am not sure if also for the whole profile.

Windows Tools, etc.

For Windows, you have

- PGP by Symantec
- GPG 4 Win (for Outlook)
- GNU tools with Cygwin


GNU Privacy Assistant - Key Manager
File Edit Keys Windows Server Help
Editar Delete Firmar Importar Exportar Brief Detailed Preferencias Actualizar Archivos Clipboard Card
Key Manager

		ID de Clave	Created	Nombre de Usuario
	P	[redacted]	2015-03-24	[redacted]
	P	[redacted]	2015-03-27	[redacted]
	P	E8918B35	2013-07-02	Luis J. Dominguez Perez (Gmail) <luisjdominguezp@gmail.com>

Esta clave tiene tanto una parte secreta como una pública
Esta clave puede utilizarse para certificación, firma y cifrado.
User name: Luis J. Dominguez Perez (Gmail) <luisjdominguezp@gmail.com>
ID de Clave: E8918B35
Huella dactilar: 7AB0 2D7C 0EA1 217A 100F F82A 5846 C79F E891 8B35
Caduca el: 2018-07-01
Confianza en el usuario: Desconocida
Key validity: Desconocida
Key type: DSA 3072 bits
Creada el: 2013-07-02

Clave por defecto seleccionada: [redacted]

Mailvelope

- Is a Chrome, and Firefox plug-in for using PGP encryption in your webmail clients.
- Uses: OpenPGP.js, email.js, DOMpurify, Bootstrap, jQuery, Oxygen icons

Supports:

- Key Management
- Key Generation
- Key Import/Export
- Encrypt/Decrypt, and signing

The keys are stored in your profile. Only Firefox provides a strong encryption for password, but I am not sure if also for the whole profile.

pe

Chrome, and Firefox plug-in for using PGP
ption in your webmail clients.

OpenPGP.js, email.js, DOMpurify,
strap, jQuery, Oxygen icons

s:

Management

Generation

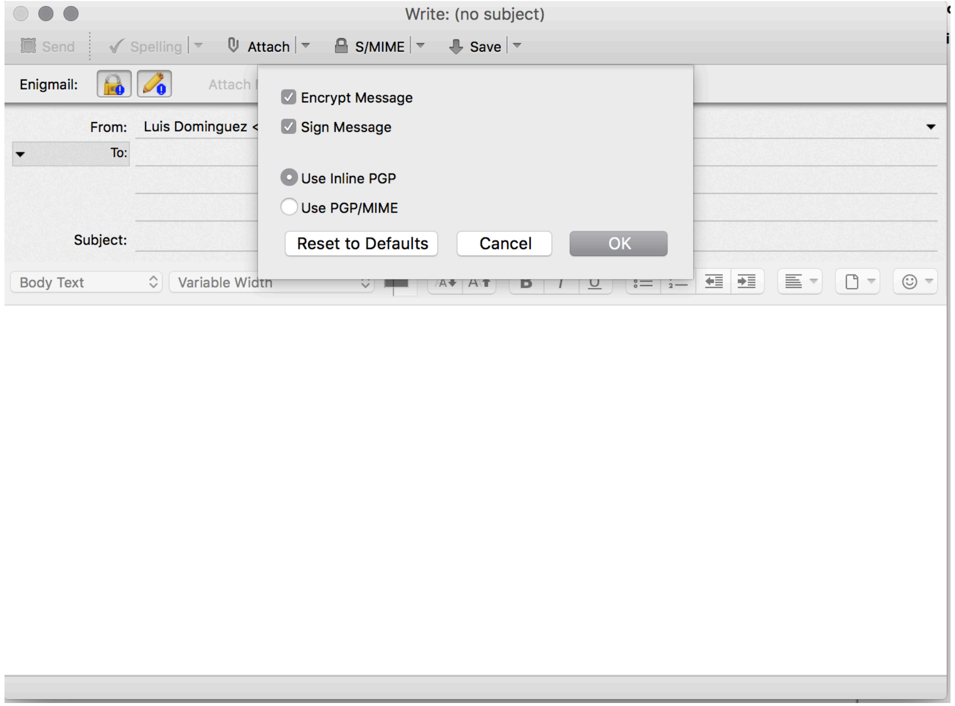
mport/Export

pt/Decrypt, and signing

s are stored in your profile. Only Firefox
a strong encryption for password, but I
sure if also for the whole profile.

Multiplataform

- A cross platform solution is Enigmail, a plug-in for Thunderbird
- It connects to your GPG installation



Write: (no subject)

From: Luis Dominguez <

To:

Subject:

☑ Encrypt Message

☑ Sign Message

● Use Inline PGP

○ Use PGP/MIME

Reset to Defaults | Cancel | OK

Browsing

- Priva
  on m
  - No
  - No
  - No
  - No
  - No
  - Ye
  - Ye
  - Ye
    bro
  - Ye
    ac

That's r

fox plug-in for using PGP
mail clients.

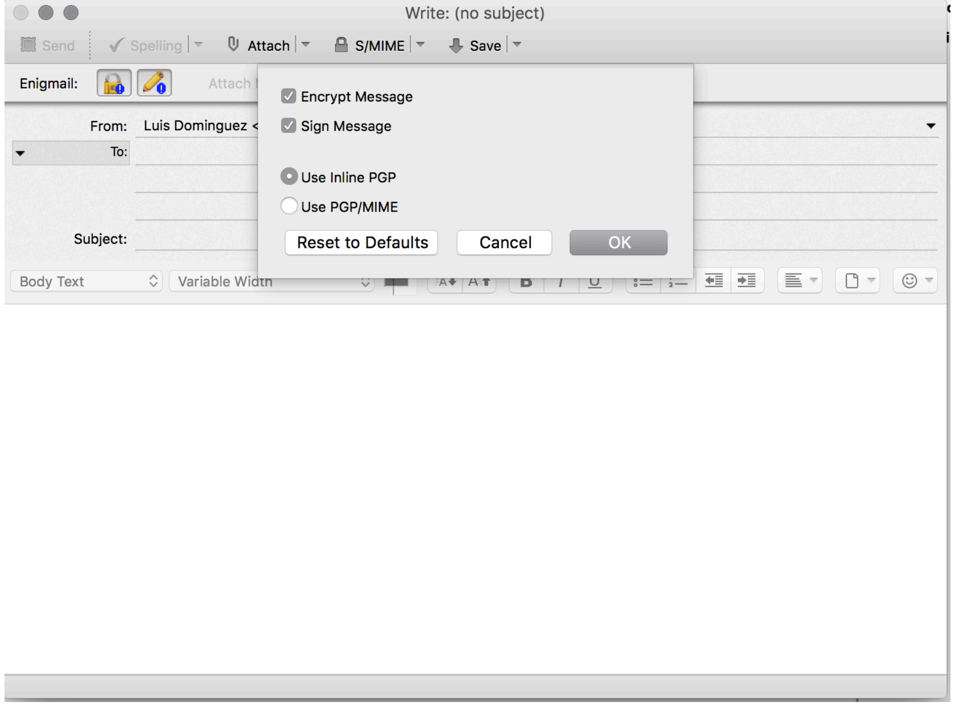
ail.js, DOMpurify,
ygen icons

signing

ur profile. Only Firefox
ion for password, but I
e whole profile.

Multiplataform

- A cross platform solution is Enigmail, a plug-in for Thunderbird
- It connects to your GPG installation



☑ Encrypt Message
☑ Sign Message
● Use Inline PGP
○ Use PGP/MIME

Browsing

- Private mode browsing on most browsers:
  - No saving cookies
  - No tracking
  - No history
  - No new passwords
  - No cache
  - Yes you can save boo
  - Yes you can undo clo
  - Yes, everybody could browsing
  - Yes, you ID can be li account

That’s nice, but not enou

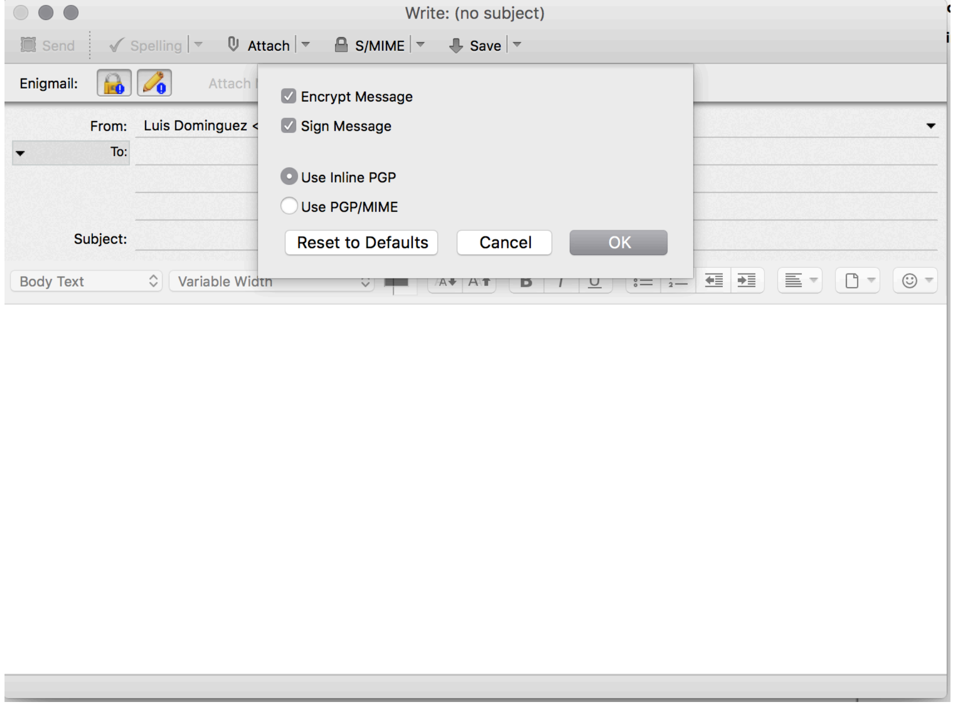
ng PGP

y,

Firefox
but I

Multiplataform

- A cross platform solution is Enigmail, a plug-in for Thunderbird
- It connects to your GPG installation


Write: (no subject)

Enigmail:

From: Luis Dominguez <

To:

Subject:

☑ Encrypt Message
☑ Sign Message
● Use Inline PGP
○ Use PGP/MIME

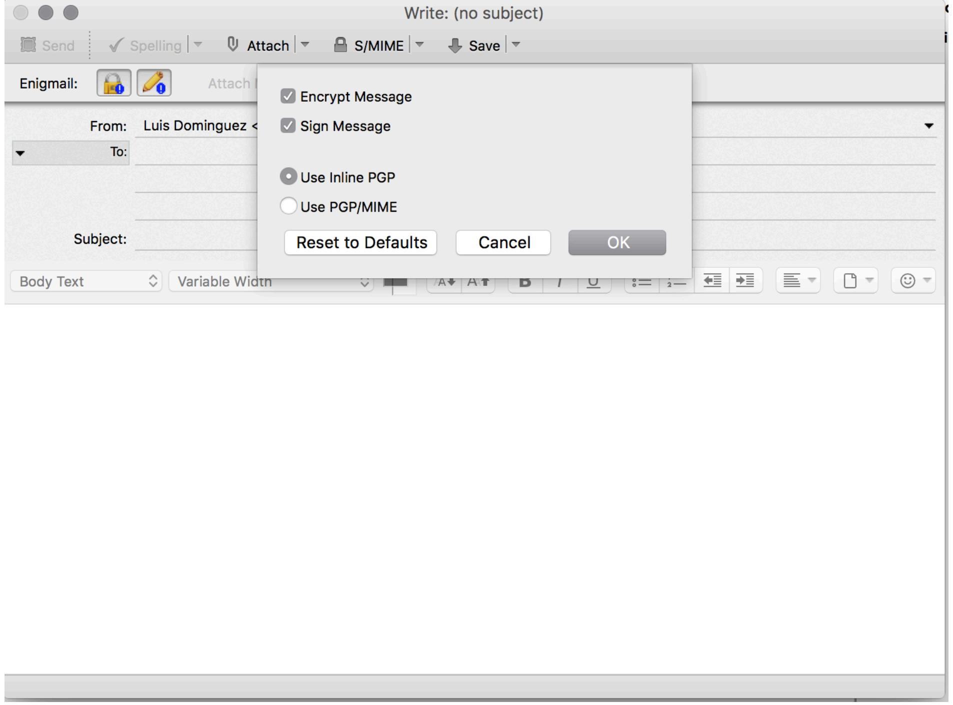
Browsing

- Private mode browsing is now easy to a on most browsers:
  - No saving cookies
  - No tracking
  - No history
  - No new passwords
  - No cache
  - Yes you can save bookmarks
  - Yes you can undo closing tabs
  - Yes, everybody could detect where you browsing
  - Yes, you ID can be linked if you login in account

That’s nice, but not enough for many. . .

Multiplataform

- A cross platform solution is Enigmail, a plug-in for Thunderbird
- It connects to your GPG installation



Write: (no subject)

From: Luis Dominguez <

To:

Subject:

☑ Encrypt Message

☑ Sign Message

● Use Inline PGP

○ Use PGP/MIME

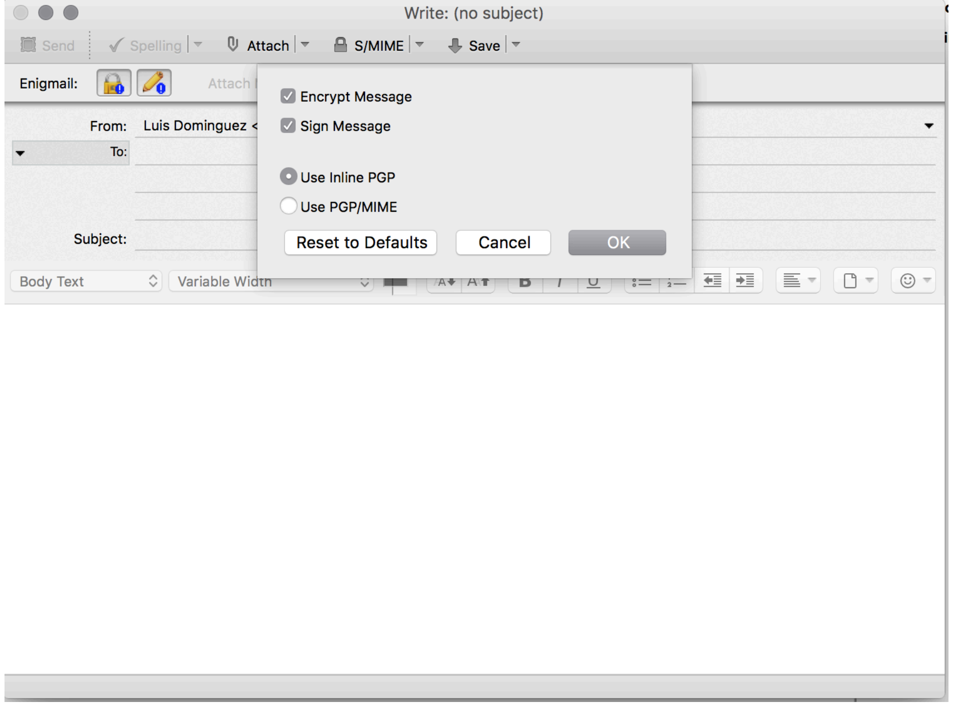
Browsing

- Private mode browsing is now easy to activate on most browsers:
  - No saving cookies
  - No tracking
  - No history
  - No new passwords
  - No cache
  - Yes you can save bookmarks
  - Yes you can undo closing tabs
  - Yes, everybody could detect where you are browsing
  - Yes, you ID can be linked if you login into your account

That's nice, but not enough for many. . .

Multiplataform

- A cross platform solution is Enigmail, a plug-in for Thunderbird
- It connects to your GPG installation



From: Luis Dominguez <

To:

Subject:

☑ Encrypt Message

☑ Sign Message

● Use Inline PGP

○ Use PGP/MIME

Browsing

- Private mode browsing is now easy to activate on most browsers:
  - No saving cookies
  - No tracking
  - No history
  - No new passwords
  - No cache
  - Yes you can save bookmarks
  - Yes you can undo closing tabs
  - Yes, everybody could detect where you are browsing
  - Yes, you ID can be linked if you login into your account

That's nice, but not enough for many. . .

Browsing

- Private mode browsing is now easy to activate on most browsers:
  - No saving cookies
  - No tracking
  - No history
  - No new passwords
  - No cache
  - Yes you can save bookmarks
  - Yes you can undo closing tabs
  - Yes, everybody could detect where you are browsing
  - Yes, you ID can be linked if you login into your account

That's nice, but not enough for many. . .

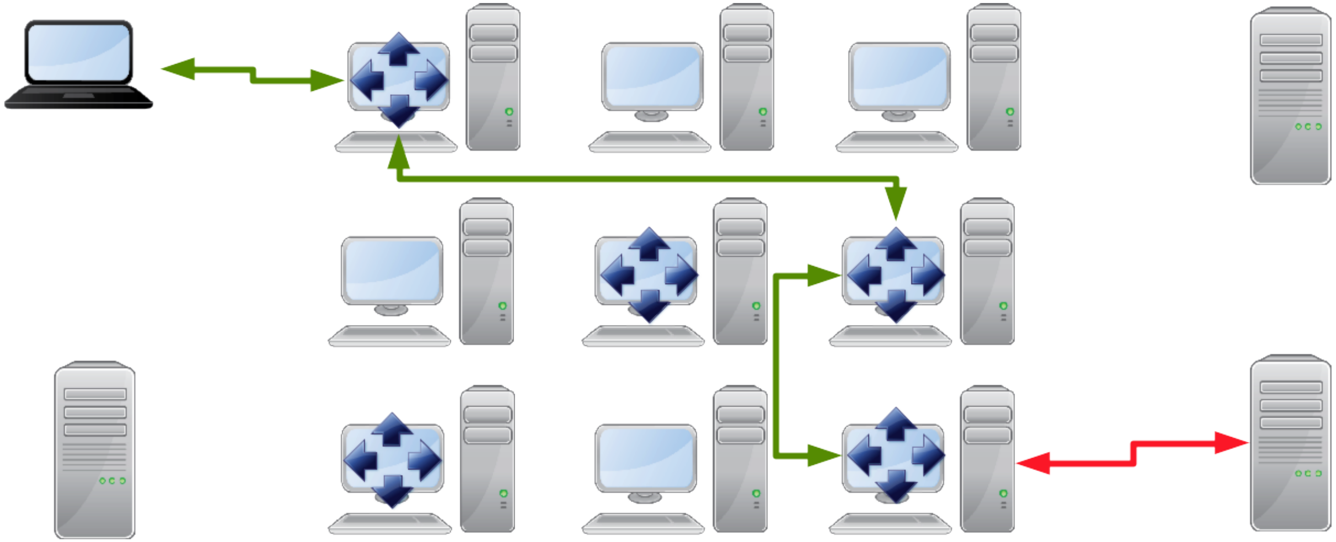
Tor network

"Tor is free software and an open network that helps you defend against traffic analysis, a form of network surveillance that threatens personal freedom and privacy, confidential business activities and relationships, and state security."

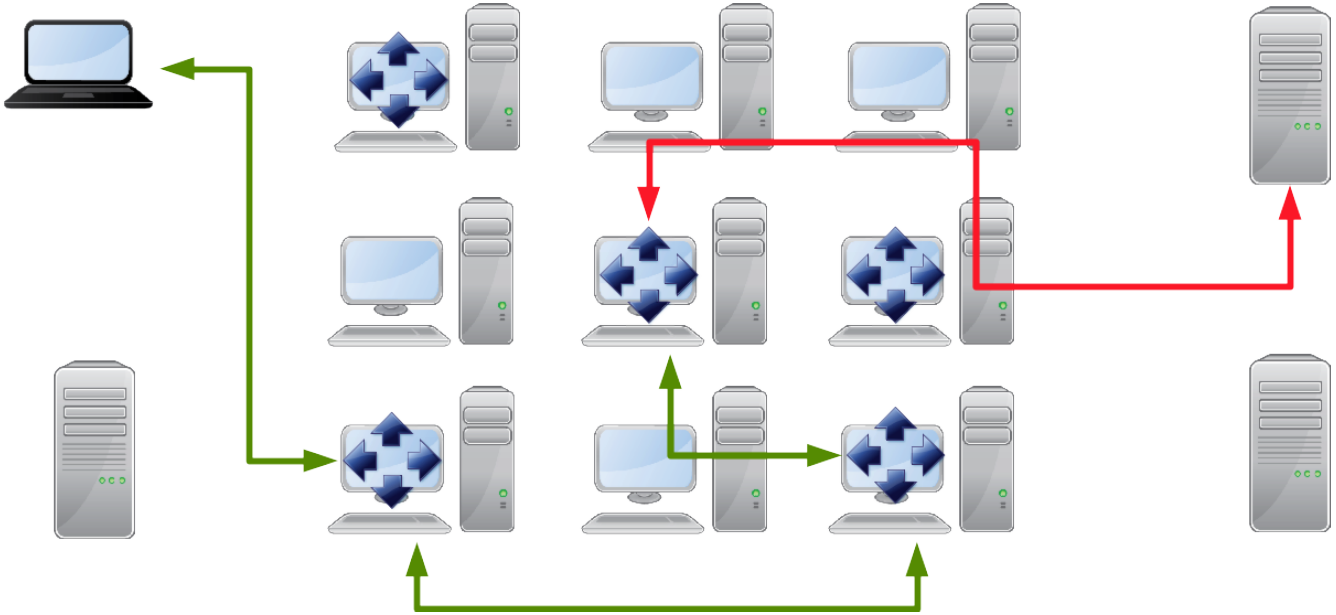
- It helps to anonymize your browsing experience from application providers
- Used by journalists, and people in general for free-speech
- Also used for illegal traffic, and terrorists
- Helps on testing network issues
- Military uses it for information gathering

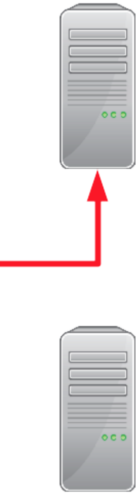
Does not forbids you to leak your information (if you login into facebook, you are no longer anonymous)

Tor diagram



Tor diagram





Chat

Instant message conversations are also susceptible to be read on the internet

For mobile, we have very few solutions:

- Telegram - multiplataform, uses “perfect forward secrecy” (100 times, or weekly) for its secure chat mode… but it’s an *in-house* protocol
- BBM using a BES uses Triple DES encryption, as it is the recommended standard in FIPS (they would change it accordingly)
- iMessage uses some sort of encryption, but it has a bad record… perhaps it uses AES-128 until you download your messages
- Another good alternative is Cryptocat, which also works on iPhone… they use OTR, with all the flashes

Chat

Instant message conversations are also susceptible to be read on the internet

For mobile, we have very few solutions:

- Telegram - multiplataform, uses “perfect forward secrecy” (100 times, or weekly) for its secure chat mode. . . but it’s an *in-house* protocol
- BBM using a BES uses Triple DES encryption, as it is the recommended standard in FIPS (they would change it accordingly)
- iMessage uses some sort of encryption, but it has a bad record. . . perhaps it uses AES-128 until you download your messages
- Another good alternative is Cryptocat, which also works on iPhone. . . they use OTR, with all the flashes

Chat

For servers we

- An XMPP s
  PGP for the
- Microsoft ha
  chatting, an
  applications.
  am not sure
  you receive t

Chat

Instant message conversations are also susceptible to be read on the internet

For mobile, we have very few solutions:

- Telegram - multiplataform, uses “perfect forward secrecy” (100 times, or weekly) for its secure chat mode… but it’s an *in-house* protocol
- BBM using a BES uses Triple DES encryption, as it is the recommended standard in FIPS (they would change it accordingly)
- iMessage uses some sort of encryption, but it has a bad record… perhaps it uses AES-128 until you download your messages
- Another good alternative is Cryptocat, which also works on iPhone… they use OTR, with all the flashes

Chat

For servers we

- An XMPP s PGP for the
- Microsoft ha chatting, an applications. am not sure you receive t

Chat

Instant message conversations are also susceptible to be read on the internet

For mobile, we have very few solutions:

- Telegram - multiplataform, uses "perfect forward secrecy" (100 times, or weekly) for its secure chat mode... but it's an *in-house* protocol
- BBM using a BES uses Triple DES encryption, as it is the recommended standard in FIPS (they would change it accordingly)
- iMessage uses some sort of encryption, but it has a bad record... perhaps it uses AES-128 until you download your messages
- Another good alternative is Cryptocat, which also works on iPhone... they use OTR, with all the flashes

Chat

For servers we have:

- An XMPP server could be co
PGP for the messages
- Microsoft has an enterprise s
chatting, and there are a few
applications... protects comr
am not sure how they store t
you receive them.

Chat

Instant message conversations are also susceptible to be read on the internet

For mobile, we have very few solutions:

- Telegram - multiplataform, uses “perfect forward secrecy” (100 times, or weekly) for its secure chat mode. . . but it’s an *in-house* protocol
- BBM using a BES uses Triple DES encryption, as it is the recommended standard in FIPS (they would change it accordingly)
- iMessage uses some sort of encryption, but it has a bad record. . . perhaps it uses AES-128 until you download your messages
- Another good alternative is Cryptocat, which also works on iPhone. . . they use OTR, with all the flashes

Chat

For servers we have:

- An XMPP server could be configured to use PGP for the messages
- Microsoft has an enterprise solution for chatting, and there are a few mobile applications. . . protects communication, but I am not sure how they store the messages until you receive them.

message conversations are also susceptible
ad on the internet

ile, we have very few solutions:

ram - multiplataform, uses “perfect
rd secrecy” (100 times, or weekly) for its
e chat mode. . . but it’s an *in-house*
col

using a BES uses Triple DES encryption,
is the recommended standard in FIPS
would change it accordingly)

sage uses some sort of encryption, but it
bad record. . . perhaps it uses AES-128
you download your messages

her good alternative is Cryptocat, which
vorks on iPhone. . . they use OTR, with all
ashes

Chat

For servers we have:

- An XMPP server could be configured to use PGP for the messages
- Microsoft has an enterprise solution for chatting, and there are a few mobile applications. . . protects communication, but I am not sure how they store the messages until you receive them.

Future

We need

but also

tions are also susceptible
t

few solutions:

orm, uses “perfect
times, or weekly) for its
ut it’s an *in-house*

s Triple DES encryption,
ded standard in FIPS
accordingly)

rt of encryption, but it
erhaps it uses AES-128
ur messages

ive is Cryptocat, which
. . they use OTR, with all

Chat

For servers we have:

- An XMPP server could be configured to use PGP for the messages
- Microsoft has an enterprise solution for chatting, and there are a few mobile applications. . . protects communication, but I am not sure how they store the messages until you receive them.

Future

We need to work on usab

but also more protocols,

ceptible

ct
for its
se

ryption,
IPS

but it
S-128

which
with all

Chat

For servers we have:

- An XMPP server could be configured to use PGP for the messages
- Microsoft has an enterprise solution for chatting, and there are a few mobile applications. . . protects communication, but I am not sure how they store the messages until you receive them.

Future

We need to work on usability

but also more protocols, and primitives are

Chat

For servers we have:

- An XMPP server could be configured to use PGP for the messages
- Microsoft has an enterprise solution for chatting, and there are a few mobile applications… protects communication, but I am not sure how they store the messages until you receive them.

Future

We need to work on usability

but also more protocols, and primitives are needed

Chat

For servers we have:

- An XMPP server could be configured to use PGP for the messages
- Microsoft has an enterprise solution for chatting, and there are a few mobile applications... protects communication, but I am not sure how they store the messages until you receive them.

Future

We need to work on usability

but also more protocols, and primitives are needed

“Understanding tools for a more secure internet” 2nd cyber-security week @ CIC-IPN.

Luis J. Dominguez Perez
CONACyT. CIMAT

luis.dominguez@cimat.mx

CONACYT

Centro de Investigación
en Computación
Instituto Politécnico Nacional

CIMAT