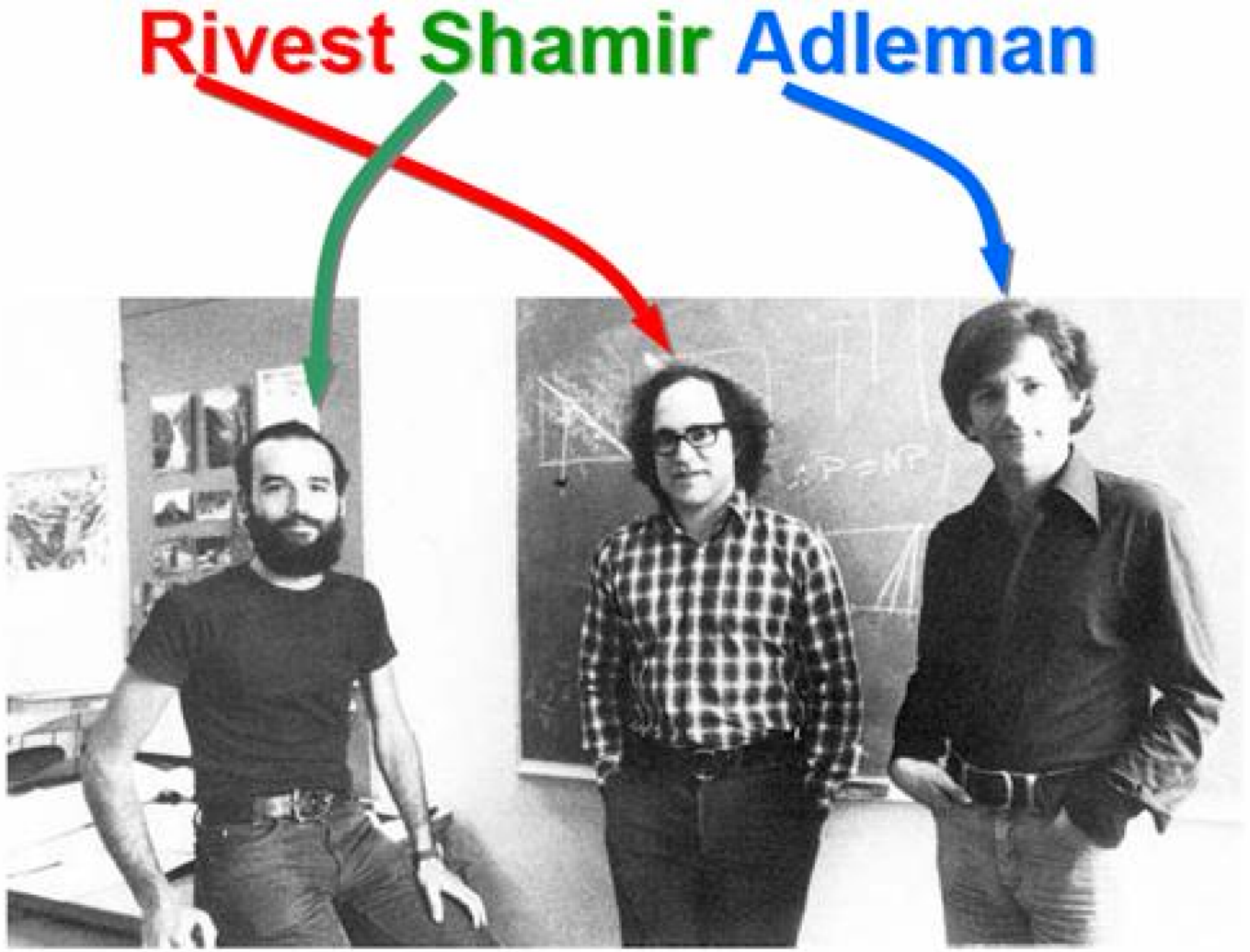
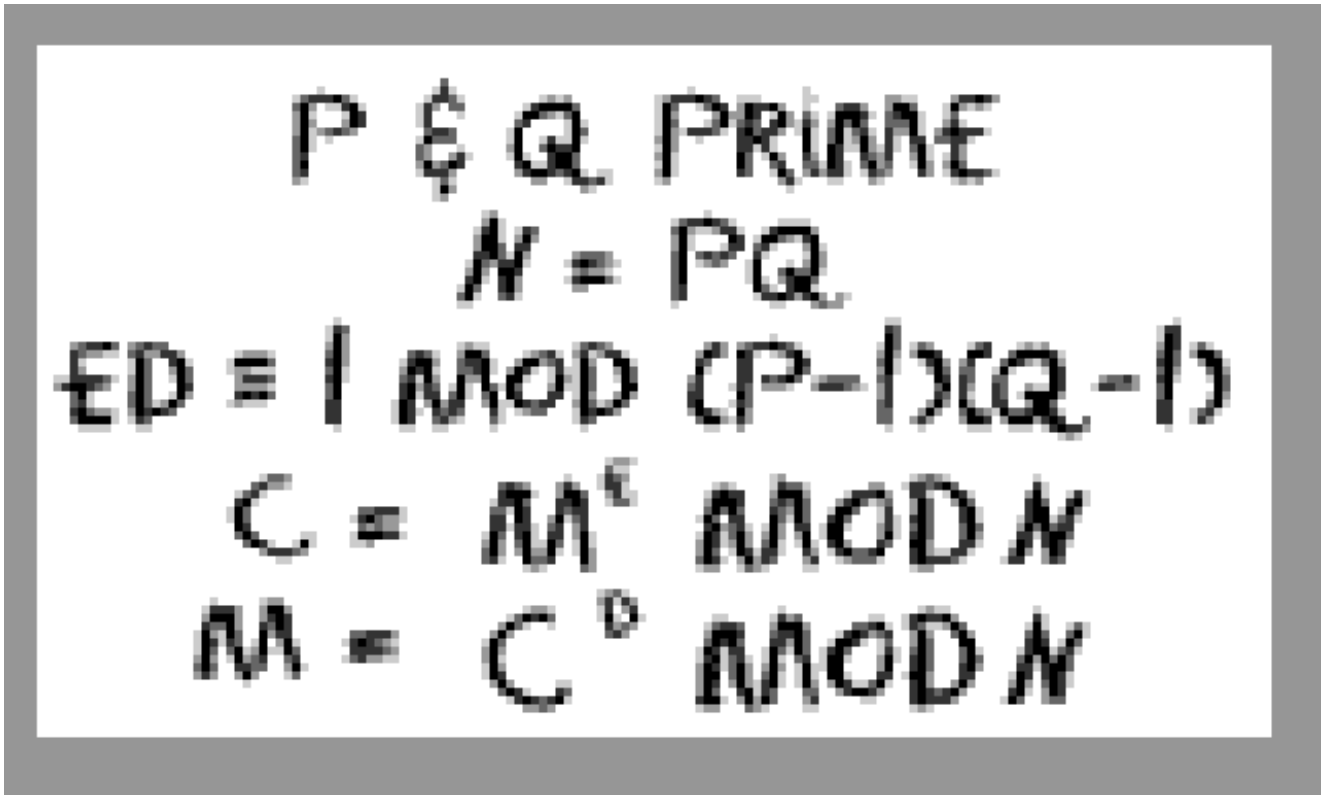
Public-Key Cryptography and RSA

Algebra I

The inventors of the RSA cryptosystem.



The original RSA logo.



Mister White wants to receive encrypted messages. He needs three numbers

$$N \text{ and } \quad E \qquad \text{public}$$

$$D \qquad \text{secret},$$

where $N = p \cdot q$, with p and q prime, E is the encryption key, D is the decryption key, and they satisfy the condition

$$ED \equiv 1 \pmod{(p-1)(q-1)}.$$

Let m be the message (converted into an integer).
We send him

$$C \equiv m^E \pmod{N}.$$

To read our message, mister White computes

$$C^D \pmod{N}.$$

Why does it work?

Since

$$ED = 1 + k(p-1)(q-1), \qquad \text{for some } k \in \mathbb{Z},$$

by Euler's theorem one has

$$C^D = (m^E)^D \equiv m^{ED} \equiv m^{1+k(p-1)(q-1)} \equiv m \pmod{N}.$$

Exercise

Mr. White has:

public keys $N = 391$ and $E = 223$
secret key $D = 191$.

- Check that these three numbers are OK.

(here N is small and you should be able to factor it!)

He received the encrypted message $c = 111$.

- What was the original message m?
- Send him the message $m = 77$ after encryption.

Example

Mr. White has:

public keys $N = 6077$ and $E = 113$
secret key $D = 2513$.

We send him the message $m = 11$ after encryption

$$C = m^E \pmod{N}$$

$$C = 11^{113} \equiv 2217 \pmod{N}$$

He reads the message by computing

$$C^D \pmod{N}$$

$$2217^{2513} \equiv 11 \pmod{6077}.$$

Digital signature

If Mr. White wants to be sure that the message was sent by us, then we also need a triple of integers

$$N_s,\ E_s,\ D_s.$$

We first encrypt the message with our secret key by computing

$$m_s \equiv m^{D_s} \pmod{N_s}$$

and then send him as usual

$$C_s \equiv m_s^E \pmod{N}.$$

To read it, mister White first uses his secret key D and obtains the message m_s "signed by us".

$$m_s \equiv C_s^D \pmod{N},$$

Next he uses our public keys E_s and N_s to obtain

$$m \equiv m_s^{E_s} \pmod{N_s}$$

Example

Mr. White has:

public keys $N = 6077$ and $E = 113$
secret key $D = 2513$.

We have:

public keys $N_s = 437$ and $E_s = 101$

secret key $D_s = 149$

Example: we sign the message, encrypt it and send it.

The message is $m = 11$.

The signed message is $m_s \equiv m^{D_s} \pmod{N_s}$

$$m_s \equiv 11^{149} \equiv 83 \pmod{437}$$

The encrypted signed message is $C_s \equiv m_s^E \pmod{N}$

$$C_s \equiv 83^{113} \equiv 4212 \pmod{6077}$$

Mr. White decrypts the message and checks the signature

To obtain the signed message he computes $m_s = C_s^D \pmod{N}$

$$m_s \equiv 4212^{2513} \equiv 83 \pmod{6077}$$

To obtain the original message he computes $m = m_s^{E_s} \pmod{N_s}$

$$m \equiv 83^{101} \equiv 11 \pmod{437}.$$

Example: Banca Intesa RSA certificate

The screenshot below shows the public keys of the italian bank BANCA INTESA.

The public exponent is the prime number $E = 65537$.

The modulus N is a number of 2048 bits. Here it is written in 512 hexadecimal characters, which correspond to 617 decimal digits.

The modulus N is the product of two primes p and q of approximately 300 decimal digits each.

Example: Banca Intesa RSA certificate

Subject Alt Names

DNS Name	www.intesasanpaolo.com
DNS Name	www.intesasanpaolo.it
DNS Name	intesasanpaolo.it
DNS Name	intesasanpaolo.com

Public Key Info

Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	B3:09:3B:09:CA:19:75:12:56:5A:25:E9:1C:36:3D:B3:71:0A:8E:64:5B:2A:25:58:80:77:58:D5:C4:A7:11:D5:C4:BC:84:A7:FA:69:17:9C:7D:75:B3:F4:04:72:B0:C1:F1:2F:16:7C:17:F0:91:34:B1:38:55:96:11:F1:C9:82:70:4F:62:80:95:EB:3B:1F:1B:8D:84:6F:C7:30:31:70:D8:32:94:16:15:DC:D5:24:E0:43:FA:DC:83:99:17:E3:01:A5:BA:B1:33:6C:65:8A:27:A5:B6:87:65:C5:8C:07:AF:EC:1C:E1:A9:F8:78:41:7C:7C:72:C8:85:2C:1C:9D:21:2F:8F:1A:16:52:64:86:45:DF:D0:C1:83:30:6B:E2:59:10:27:F8:BE:83:E3:E3:58:7A:9C:3A:E4:03:8B:D4:4A:64:F8:D2:4D:04:0B:8B:B8:73:94:B2:6C:73:F3:AA:7B:02:1C:6B:FD:00:84:8F:01:08:84:EF:3F:9D:47:AE:63:C1:51:2A:A6:FE:F3:4D:D3:B4:55:00:03:3F:6B:2A:F2:5F:85:AA:8B:2C:EC:73:84:2B:00:2C:EF:9D:E2:36:DB:E0:AD:AB:7F:A6:75:BC:AF:CC:A6:BE:5C:77:AE:AB:3A:9D:5B:28:53:8D:E0:DF:9B:B4:F5:EF:EB:81:55:F1

Security

Security of RSA depends on factorization hardness

Over the years the improvement of factoring algorithms and of computers required the adoption of larger and larger RSA public keys.

Or switching to different cryptosystems.

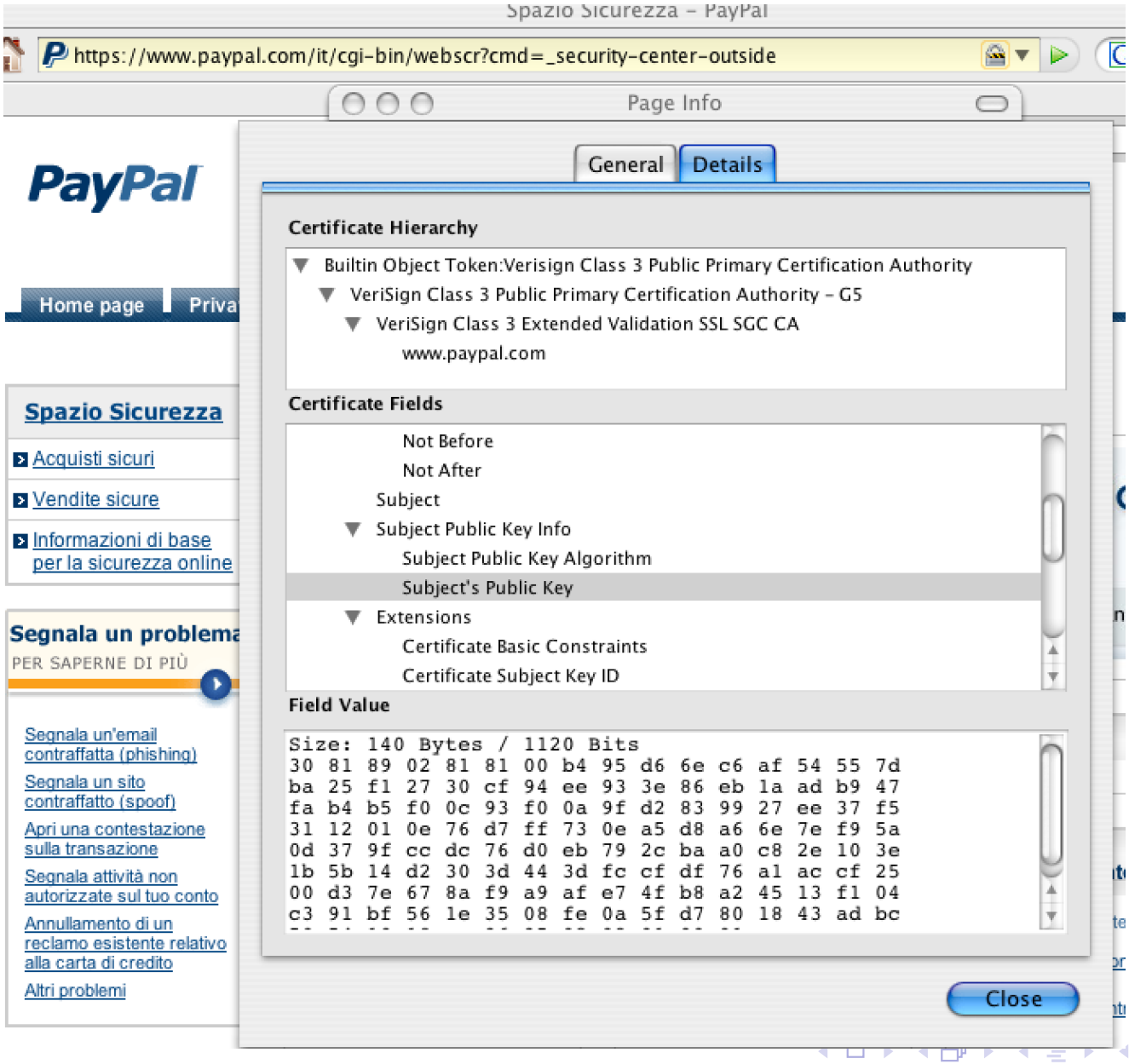
Security

Security of RSA also depends on some precautions in in the generation of the integers N, E and D

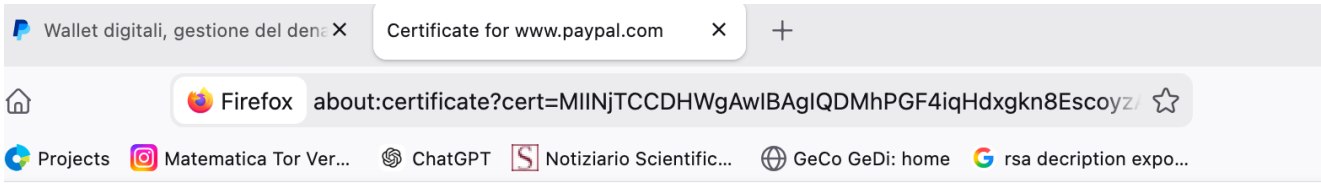
Different users should have coprime moduli N_1 and N_2!

What happens if two users have moduli N_1 and N_2 with a factor in common?

Paypal 2008: RSA modulus N of 1120 bits



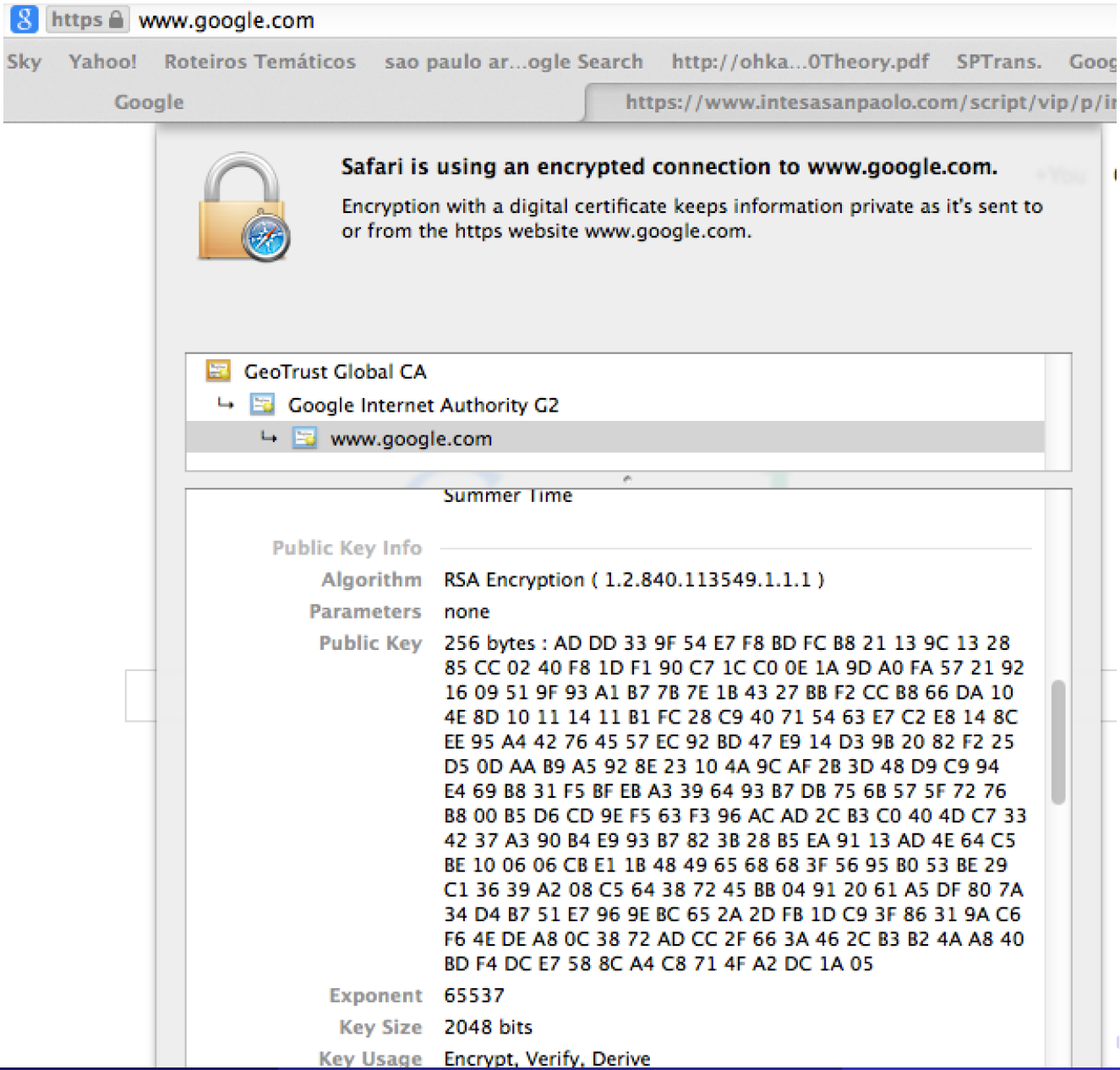
Paypal 2026: RSA modulus N of 2048 bits



Public Key Info

Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	9C:DA:71:68:AE:A1:BA:45:40:5C:80:FD:F1:CB:00:4D:24:3D:45:8A:6E:4E:E7:EC:EE:94:A6:E1:16:14:E6:A1:FC:67:01:B5:97:F8:24:0E:35:79:83:B5:C7:F4:3C:92:3D:12:4E:92:01:79:F2:D6:F1:71:1F:85:9E:8A:C6:D8:CE:D2:37:B7:01:12:85:F7:69:A4:90:0B:B0:9F:11:BC:DA:BB:47:22:6E:B0:2A:7F:7F:0B:8C:88:12:D7:F2:75:2E:C9:8B:F8:E6:02:68:EB:FD:97:5D:94:6E:9D:40:81:D6:F5:6C:04:3D:73:EA:7E:FD:AD:1E:B1:8A:4A:54:63:98:BB:EF:C7:E8:74:BD:1A:FE:51:F3:BC:92:DF:40:18:40:A6:3B:6B:6F:D2:D1:7F:3A:4A:71:B5:E6:9F:66:06:91:B7:98:E3:12:D2:D3:94:10:9E:6B:83:9E:17:C6:87:5B:DD:83:03:19:9F:DA:78:82:0A:E3:29:23:4A:14:27:E0:F3:FB:8B:03:5D:07:7E:7C:B6:12:2C:25:E0:29:61:6E:76:D3:DF:7E:72:AC:B5:43:15:17:E0:24:DD:45:7F:7E:5E:4F:75:E1:2B:F2:18:85:32:CD:D9:C5:15:4B:6C:C1:BC:2A:DD:51:E9:EA:0E:D2:A2:8C:EB:A8:68:2E:1D

Google 2014: RSA modulus N of 2048 bits



Google 2026: Elliptic Curve Discrete Logarithm

Subject Alt Names

DNS Name	mail.google.com
DNS Name	inbox.google.com

Public Key Info

Algorithm	Elliptic Curve
Key Size	256
Public Value	04:85:7B:9E:28:B4:56:A7:91:B1:5E:86:1F:E9:3D:0E:31:D4:0E:92:63:56:E8:F5:F7:5F:18:0E:67:CE:7A:21:3B:74:3C:68:4D:41:48:7C:3D:B6:4E:34:66:4D:56:66:03:55:FD:1B:A7:C7:E4:A3:44:22:EB:1A:66:12:45:1B:5D