OpenSSL

CLUG Talk

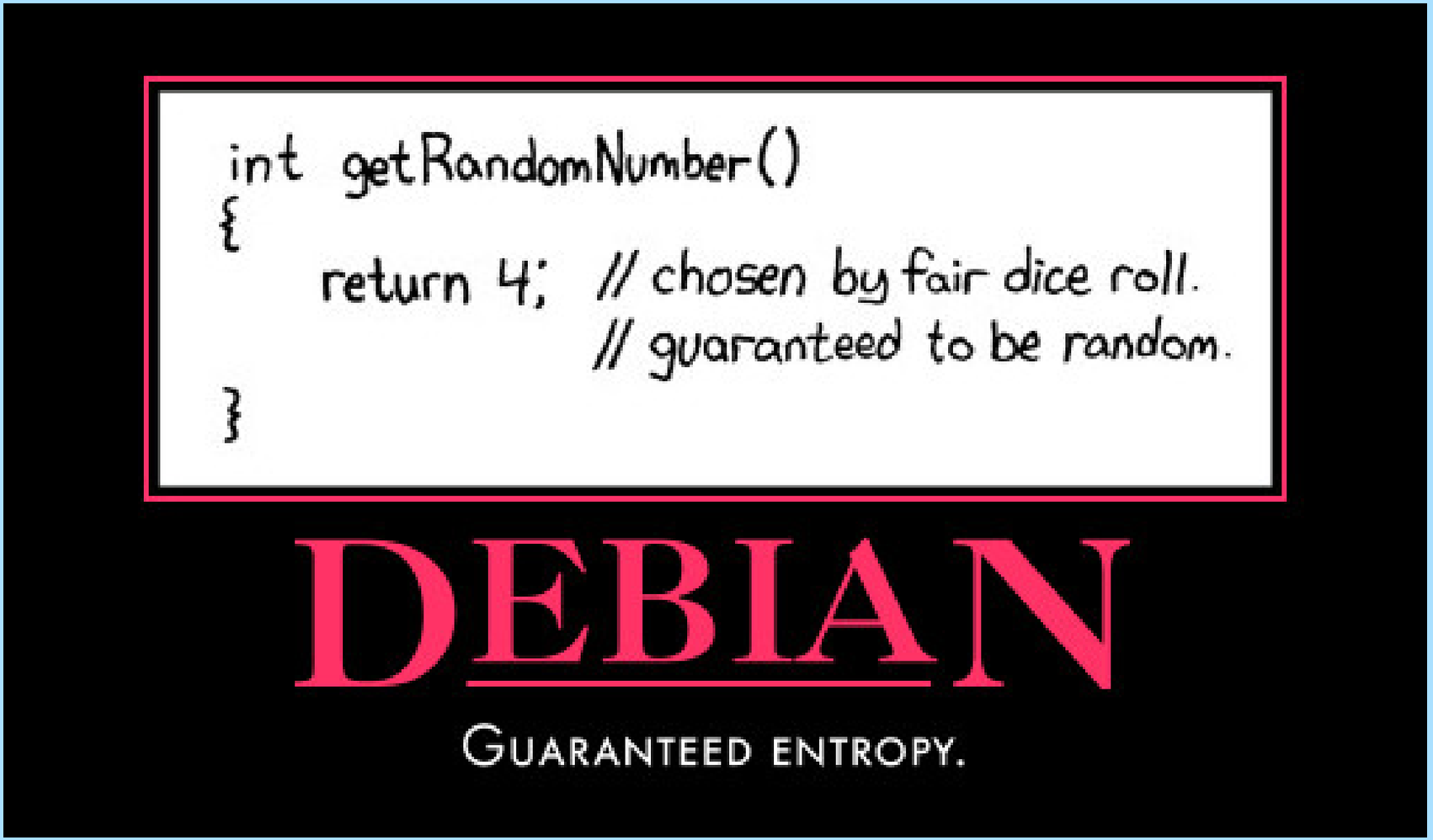
Stefano Rivera

27 May 2008

Prologue


DILBERT By Scott Adams
TOUR OF ACCOUNTING
OVER HERE WE HAVE OUR RANDOM NUMBER GENERATOR.
NINE NINE NINE NINE NINE NINE
ARE YOU SURE THAT'S RANDOM?
THAT'S THE PROBLEM WITH RANDOMNESS YOU CAN NEVER BE SURE.
DEBIAN
You can never be sure.

Prologue



Cape Linux User Group
www.clug.org.za

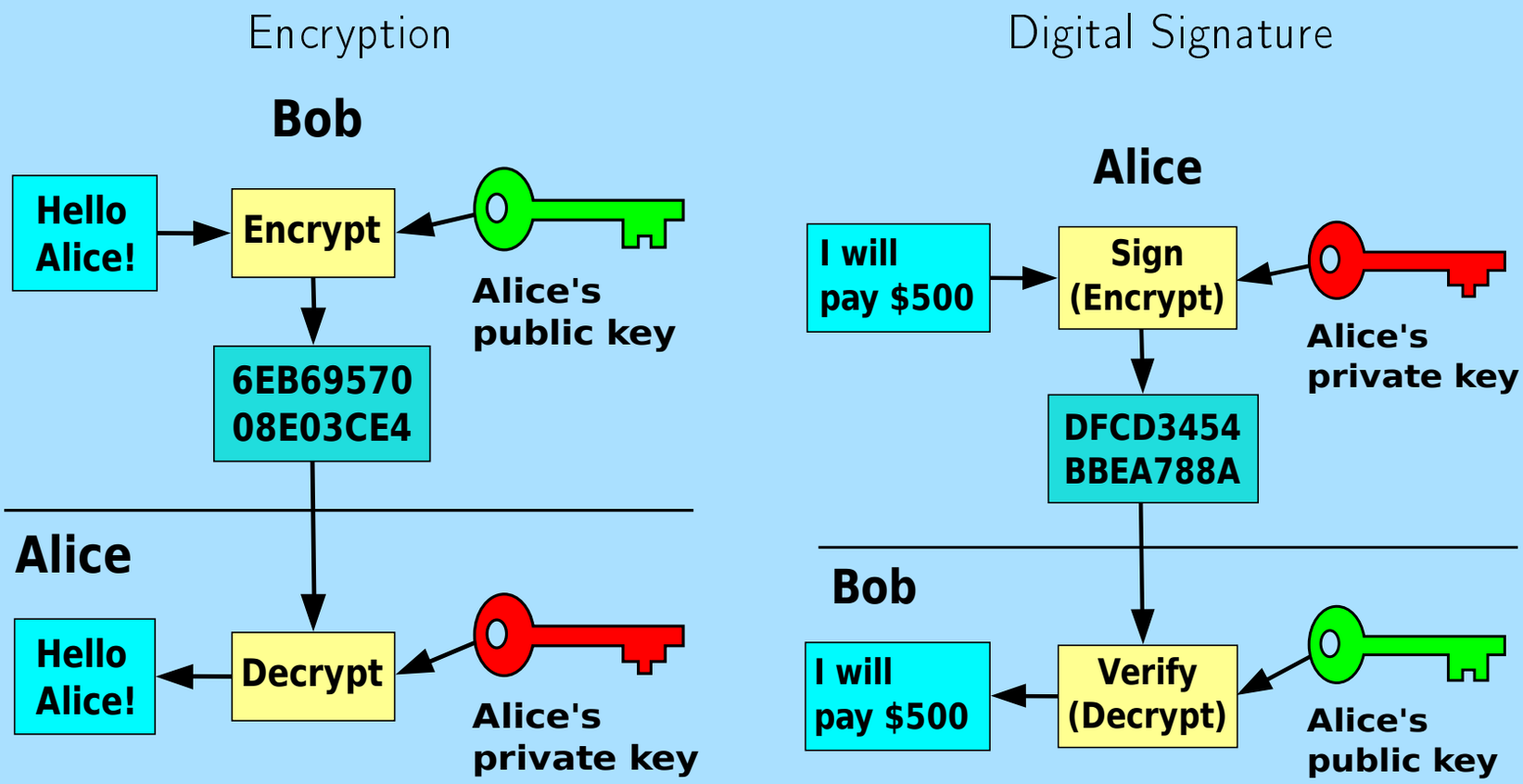
Contents

- The What and Why OpenSSL
- Necessary Background
- How to generate SSL keys
- How to run your own CA
- Q&A

What and Why

1. how

What is Public Key Crypto



Public Key example: RSA

1. Choose two prime numbers

Public Key example: RSA

1. Choose two prime numbers: $p = 61$ and $q = 53$

Public Key example: RSA

1. Choose two prime numbers: $p = 61$ and $q = 53$
2. Compute $n = pq$

Public Key example: RSA

1. Choose two prime numbers: $p = 61$ and $q = 53$
2. Compute $n = pq = (61)(53)$

Public Key example: RSA

1. Choose two prime numbers: $p = 61$ and $q = 53$
2. Compute $n = pq = (61)(53) = 3233$

Public Key example: RSA

1. Choose two prime numbers: $p = 61$ and $q = 53$
2. Compute $n = pq = (61)(53) = 3233$
3. Compute tortient: $\varphi(n) = (p-1)(q-1)$

Public Key example: RSA

1. Choose two prime numbers: $p = 61$ and $q = 53$
2. Compute $n = pq = (61)(53) = 3233$
3. Compute tortient: $\varphi(n) = (p-1)(q-1) = (61-1)(53-1)$

Public Key example: RSA

1. Choose two prime numbers: $p = 61$ and $q = 53$
2. Compute $n = pq = (61)(53) = 3233$
3. Compute tortient: $\varphi(n) = (p-1)(q-1) = (61-1)(53-1) = 3120$

Public Key example: RSA

1. Choose two prime numbers: $p = 61$ and $q = 53$
2. Compute $n = pq = (61)(53) = 3233$
3. Compute tortient: $\varphi(n) = (p-1)(q-1) = (61-1)(53-1) = 3120$
4. Choose $e > 1$ comprime to $\varphi(n)$

Cape Linux User Group
www.clug.org.za

Public Key example: RSA

1. Choose two prime numbers: $p = 61$ and $q = 53$
2. Compute $n = pq = (61)(53) = 3233$
3. Compute tortient: $\varphi(n) = (p-1)(q-1) = (61-1)(53-1) = 3120$
4. Choose $e > 1$ comprime to $\varphi(n)$: $e = 17$

Public Key example: RSA

1. Choose two prime numbers: $p = 61$ and $q = 53$
2. Compute $n = pq = (61)(53) = 3233$
3. Compute tortient: $\varphi(n) = (p-1)(q-1) = (61-1)(53-1) = 3120$
4. Choose $e > 1$ comprime to $\varphi(n)$: $e = 17$
5. Compute d such that $de \equiv 1 \pmod{\varphi(n)}$ i.e. $de = 1 + k\varphi(n)$

Cape Linux User Group
www.clug.org.za

Public Key example: RSA

1. Choose two prime numbers: $p = 61$ and $q = 53$
2. Compute $n = pq = (61)(53) = 3233$
3. Compute tortient: $\varphi(n) = (p-1)(q-1) = (61-1)(53-1) = 3120$
4. Choose $e > 1$ comprime to $\varphi(n)$: $e = 17$
5. Compute d such that $de \equiv 1 \pmod{\varphi(n)}$ i.e. $de = 1 + k\varphi(n) : d = 2763$

Cape Linux User Group
www.clug.org.za

Public Key example: RSA

1. Choose two prime numbers: $p = 61$ and $q = 53$
2. Compute $n = pq = (61)(53) = 3233$
3. Compute tortient: $\varphi(n) = (p-1)(q-1) = (61-1)(53-1) = 3120$
4. Choose $e > 1$ comprime to $\varphi(n)$: $e = 17$
5. Compute d such that $de \equiv 1 \pmod{\varphi(n)}$ i.e. $de = 1 + k\varphi(n)$: $d = 2763$

   $17 * 2753 = 46801 = 1 + 15 * 3120$

Public Key example: RSA

1. Choose two prime numbers: $p = 61$ and $q = 53$
2. Compute $n = pq = (61)(53) = 3233$
3. Compute tortient: $\varphi(n) = (p-1)(q-1) = (61-1)(53-1) = 3120$
4. Choose $e > 1$ comprime to $\varphi(n)$: $e = 17$
5. Compute d such that $de \equiv 1 \pmod{\varphi(n)}$ i.e. $de = 1 + k\varphi(n)$: $d = 2763$

   $17 * 2753 = 46801 = 1 + 15 * 3120$

Private Key: $(n = 3233,\ e = 17)$

Public_Key: $(n = 3233,\ d = 2753)$

Cape Linux User Group
www.clug.org.za

Encryption & Signing with RSA Key

Encryption: $c = m^e \mod n = m^{17} \mod 3233$

Decryption: $m = c^d \mod n = c^{2753} \mod 3233$

Cape Linux User Group
www.clug.org.za

Encryption & Signing with RSA Key

Encryption: $c = m^e \mod n = m^{17} \mod 3233$

Decryption: $m = c^d \mod n = c^{2753} \mod 3233$

Example, encrypt $m = 123$:

1. $c = 123^{17} \mod 3233 = 855$

Cape Linux User Group
www.clug.org.za

Encryption & Signing with RSA Key

Encryption: $c = m^e \mod n = m^{17} \mod 3233$

Decryption: $m = c^d \mod n = c^{2753} \mod 3233$

Example, encrypt $m = 123$:

1. $c = 123^{17} \mod 3233 = 855$

2. $m = 855^{2753} \mod 3233 = 123$

Encryption & Signing with RSA Key

Encryption: $c = m^e \mod n = m^{17} \mod 3233$

Decryption: $m = c^d \mod n = c^{2753} \mod 3233$

Example, encrypt $m = 123$:

1. $c = 123^{17} \mod 3233 = 855$

2. $m = 855^{2753} \mod 3233 = 123$

To Sign, reverse. Hash is c, signature m.

Encryption & Signing with RSA Key

Encryption: $c = m^e \mod n = m^{17} \mod 3233$

Decryption: $m = c^d \mod n = c^{2753} \mod 3233$

Example, encrypt $m = 123$:

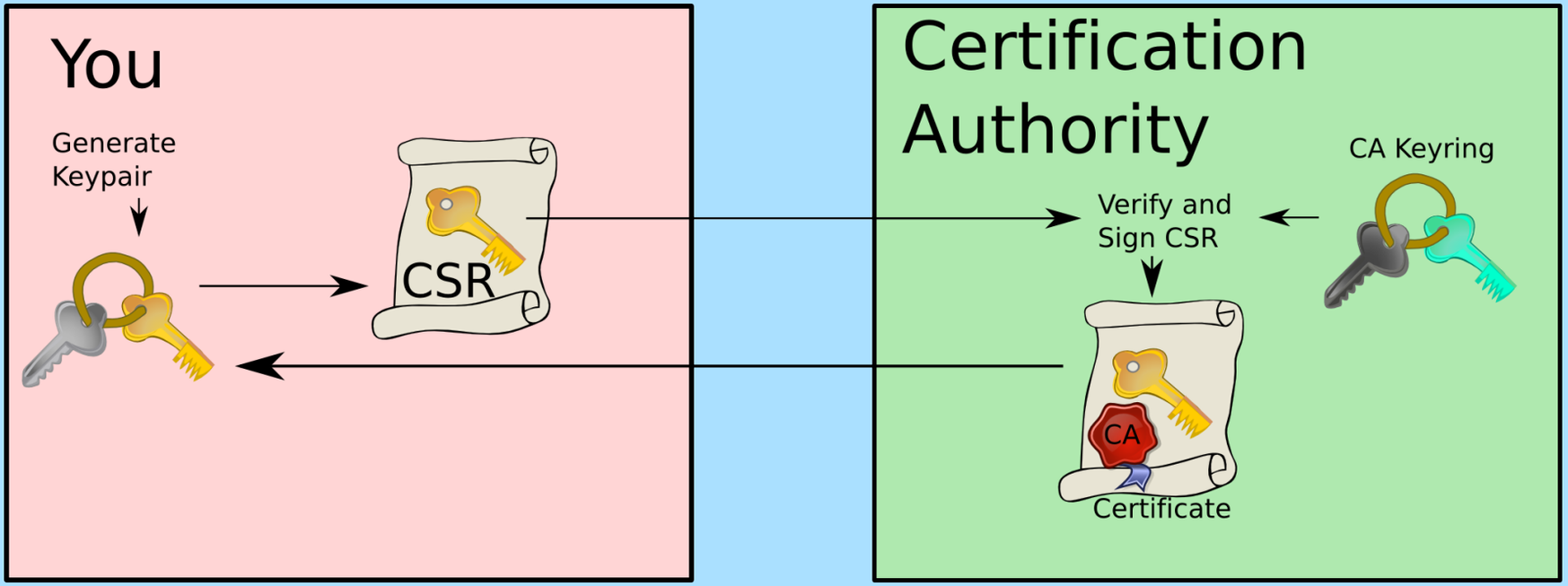
1. $c = 123^{17} \mod 3233 = 855$

2. $m = 855^{2753} \mod 3233 = 123$

To Sign, reverse. Hash is c, signature m.

Show real example

PKI


You
Generate Keypair
CSR
Certification Authority
CA Keyring
Verify and Sign CSR
CA
Certificate

In the Real World

- Look at a certificate
  - Certificate types
  - Issuee
  - Issuer (Signature)
  - Validity
  - Key (CSR)

In the Real World

- Look at a certificate
  - Certificate types
  - Issuee
  - Issuer (Signature)
  - Validity
  - Key (CSR)
- Disadvantages:
  - Money = Trust
  - "Windows" model:
    - Web Browsers
    - Java

Generating SSL keys

1. Set the correct certificate type in `openssl.conf`

2. `openssl req -newkey rsa:2048 -nodes`
   - CSR outputted
   - Private key in `privkey.pem`
     - → Move to `private/`*`foo`*`.pem`

3. Give CSR to CA

4. Install Certificate and Key
   - Install to certs/foo.pem
   - Might require chain cert

Setting up a CA

1. Create directory structure
2. Create CA key
3. Sign requests, revoke, etc.

➢ Cacert.org?

Q&A?