## The NaCl library

Peter Schwabe



#### Joint work with Daniel J. Bernstein, Tanja Lange

July 11, 2012

Africacrypt 2012 Rump Session

 Flame used MD5 weakness to infect computers through Windows update

- Flame used MD5 weakness to infect computers through Windows update
- Padding oracles keep appearing (most recently: Bardou, Focardi, Kawamoto, Simionato, Steel, Tsay, 2012)

- Flame used MD5 weakness to infect computers through Windows update
- Padding oracles keep appearing (most recently: Bardou, Focardi, Kawamoto, Simionato, Steel, Tsay, 2012)
- Timing attack steals Linux harddisk encryption key in just 65ms (Osvik, Shamir, Tromer, 2006)

- Flame used MD5 weakness to infect computers through Windows update
- Padding oracles keep appearing (most recently: Bardou, Focardi, Kawamoto, Simionato, Steel, Tsay, 2012)
- Timing attack steals Linux harddisk encryption key in just 65ms (Osvik, Shamir, Tromer, 2006)
- Remote timing attack finds OpenSSL ECDSA key (Brumley, Tuveri, 2011)

- Flame used MD5 weakness to infect computers through Windows update
- Padding oracles keep appearing (most recently: Bardou, Focardi, Kawamoto, Simionato, Steel, Tsay, 2012)
- Timing attack steals Linux harddisk encryption key in just 65ms (Osvik, Shamir, Tromer, 2006)
- Remote timing attack finds OpenSSL ECDSA key (Brumley, Tuveri, 2011)
- For more than one year Debian GNU/Linux generated OpenSSL keys with just 15 bits of entropy (fixed 2008)

- Flame used MD5 weakness to infect computers through Windows update
- Padding oracles keep appearing (most recently: Bardou, Focardi, Kawamoto, Simionato, Steel, Tsay, 2012)
- Timing attack steals Linux harddisk encryption key in just 65ms (Osvik, Shamir, Tromer, 2006)
- Remote timing attack finds OpenSSL ECDSA key (Brumley, Tuveri, 2011)
- For more than one year Debian GNU/Linux generated OpenSSL keys with just 15 bits of entropy (fixed 2008)
- Sony PlayStation 3 security system ....

- Flame used MD5 weakness to infect computers through Windows update
- Padding oracles keep appearing (most recently: Bardou, Focardi, Kawamoto, Simionato, Steel, Tsay, 2012)
- Timing attack steals Linux harddisk encryption key in just 65ms (Osvik, Shamir, Tromer, 2006)
- Remote timing attack finds OpenSSL ECDSA key (Brumley, Tuveri, 2011)
- For more than one year Debian GNU/Linux generated OpenSSL keys with just 15 bits of entropy (fixed 2008)
- Sony PlayStation 3 security system ...
- Crypto is just very complicated to use, so many ways to get things wrong
- Proper crypto is just too slow to be used in practice

- Flame used MD5 weakness to infect computers through Windows update
- Padding oracles keep appearing (most recently: Bardou, Focardi, Kawamoto, Simionato, Steel, Tsay, 2012)
- Timing attack steals Linux harddisk encryption key in just 65ms (Osvik, Shamir, Tromer, 2006)
- Remote timing attack finds OpenSSL ECDSA key (Brumley, Tuveri, 2011)
- For more than one year Debian GNU/Linux generated OpenSSL keys with just 15 bits of entropy (fixed 2008)
- Sony PlayStation 3 security system ...
- Crypto is just very complicated to use, so many ways to get things wrong
- Proper crypto is just too slow to be used in practice

#### Let's fix this. Let's take a look at the NaCl library

### Usability of NaCl

#### Authenticated encryption

c = crypto\_box(m,n,pkR,skS)

#### Verification and decrypt

Before that: key generation on each side

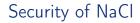
pk = crypto\_box\_keypair(&sk)

# Usability of NaCl

- All inputs and outputs are C++ std::string variables, sequences of bytes
- m: plaintext message (packet)
- n: 24-byte nonce
- skS/pkS: sender's secret key/public key (both 32 bytes)
- skR/pkR: recipient's secret key/public key (both 32 bytes)
- c: Authenticated ciphertext, 16 bytes longer than m

# Usability of NaCl

- All inputs and outputs are C++ std::string variables, sequences of bytes
- m: plaintext message (packet)
- n: 24-byte nonce
- skS/pkS: sender's secret key/public key (both 32 bytes)
- skR/pkR: recipient's secret key/public key (both 32 bytes)
- c: Authenticated ciphertext, 16 bytes longer than m
- Similarly simple API for cryptographic signatures



▶ No low-security primitives: 128-bit security level for all primitives

## Security of NaCl

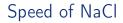
- ▶ No low-security primitives: 128-bit security level for all primitives
- Timing attacks are impossible: No data flow from secret data into addresses or branch conditions

## Security of NaCl

- ▶ No low-security primitives: 128-bit security level for all primitives
- Timing attacks are impossible: No data flow from secret data into addresses or branch conditions
- ▶ No padding oracles: Always authenticate, then decrypt

## Security of NaCl

- ▶ No low-security primitives: 128-bit security level for all primitives
- Timing attacks are impossible: No data flow from secret data into addresses or branch conditions
- ► No padding oracles: Always authenticate, then decrypt
- ▶ No randomness if unnecessary, e.g. deterministic signing
- Centralize randomness: use /dev/urandom



#### Wow, that has to be slow then!

It's not!

- ► It's not! For example on a single AMD Phenom II X6 1100T CPU:
- More than 80000 crypto\_box operations per second
- More than 80000 crypto\_box\_open operations per second
- More than 70000 crypto\_sign\_open per second
- More than 180000 crypto\_sign operations per second

- ► It's not! For example on a single AMD Phenom II X6 1100T CPU:
- More than 80000 crypto\_box operations per second
- More than 80000 crypto\_box\_open operations per second
- More than 70000 crypto\_sign\_open per second
- More than 180000 crypto\_sign operations per second
- Keeps up with the network
- ► Connection flooded with 50-byte packets: 32 Mbits per second

- ► It's not! For example on a single AMD Phenom II X6 1100T CPU:
- More than 80000 crypto\_box operations per second
- More than 80000 crypto\_box\_open operations per second
- More than 70000 crypto\_sign\_open per second
- More than 180000 crypto\_sign operations per second
- Keeps up with the network
- ► Connection flooded with 50-byte packets: 32 Mbits per second
- NaCl is much faster for larger packets

- ► It's not! For example on a single AMD Phenom II X6 1100T CPU:
- More than 80000 crypto\_box operations per second
- More than 80000 crypto\_box\_open operations per second
- More than 70000 crypto\_sign\_open per second
- More than 180000 crypto\_sign operations per second
- Keeps up with the network
- ► Connection flooded with 50-byte packets: 32 Mbits per second
- NaCl is much faster for larger packets
- NaCl is even faster for multiple packets sent to the same public key

- ► It's not! For example on a single AMD Phenom II X6 1100T CPU:
- More than 80000 crypto\_box operations per second
- More than 80000 crypto\_box\_open operations per second
- More than 70000 crypto\_sign\_open per second
- More than 180000 crypto\_sign operations per second
- Keeps up with the network
- ► Connection flooded with 50-byte packets: 32 Mbits per second
- NaCl is much faster for larger packets
- NaCl is even faster for multiple packets sent to the same public key
- NaCl is even faster when verifying signatures in batches

- ► It's not! For example on a single AMD Phenom II X6 1100T CPU:
- More than 80000 crypto\_box operations per second
- More than 80000 crypto\_box\_open operations per second
- More than 70000 crypto\_sign\_open per second
- More than 180000 crypto\_sign operations per second
- Keeps up with the network
- ► Connection flooded with 50-byte packets: 32 Mbits per second
- NaCl is much faster for larger packets
- NaCl is even faster for multiple packets sent to the same public key
- NaCl is even faster when verifying signatures in batches
- NaCl uses encrypt-then-MAC: Forged packets get dropped before decryption

### NaCl online

#### http://nacl.cr.yp.to

- ▶ NaCl is in the public domain
- NaCl steers clear of all patents that we have investigated and has not received any claims of patent infringement