Asymmetric encryption with 2 XOR's: the cipher **TCHo**

Jean-Philippe Aumasson

University of Applied Sciences Northwestern Switzerland
School of Engineering

Most of public-key schemes reduce security to integer factorization, discrete log, lattice problems, etc.

But:

- on **quantum** computers, RSA, ECC, ElGamal, etc. are broken
- in **light hardware**, complex and often slow implementation

On the other hand, LFSR-based (symmetric) ciphers fit well lightweight environments...

TCHo

- encrypts with only a LFSR and pseudorandom bits
- decrypts with simple linear algebra over GF(2)
- reduces semantic security to a hard problem,
- is not broken by quantum computers

TCHo AND RSA

Public key:

- **TCHo**: **irreducible** polynomial P
- RSA: **composite** integer $n = pq$

Private key:

- **TCHo**: a sparse **multiple** of P
- RSA: a prime **factor** of n

Hard problem:

- **TCHo**: finding a sparse **multiple** (polynomial)
- RSA: finding a prime **factor** (integer)

Encryption:

- **TCHo**: encryption is **probabilistic**
- RSA: encryption is **deterministic**

DESCRIPTION OF **TCHo**

HISTORY

Original **TCHo** in [Finiasz-Vaudenay 06]

Improvement in [Aumasson-Finiasz-Meier-Vaudenay 07], with

- **faster** encryption
- more **security** arguments
- performance **benchmarks**

Here we present the **new TCHo**.

ENCRYPTION

$10101001 \ldots 10101001$	**repetition** of $m \| m \| \ldots \| m$
$\oplus$	
$00100100 \ldots 00100010$	**random bits** with bias $\gamma = \Pr(0) - \Pr(1)$
$\oplus$	
$01110110 \ldots 01101110$	LFSR_P(**random state**)

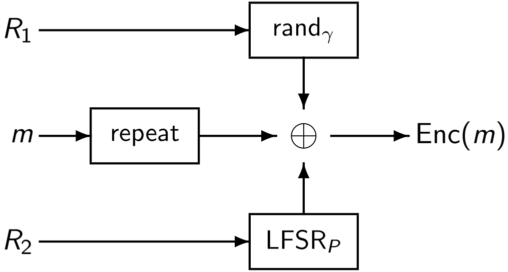
such that

- P is the **public key**
- $\gamma > 0$ (more zeros than ones)

$$\mathsf{Enc}(m) = \left(m || \ldots || m\right) \oplus \mathsf{rand}_\gamma(R_1) \oplus \mathsf{LFSR}_P(R_2)$$

ENCRYPTION

Implementation built on three **independent components**, fed with two random (unbiased) samples R_1 and R_2
$\Rightarrow$ parallelizable



LFSR_P plaintext-independent $\Rightarrow$ can be precomputed

DECRYPTION

$$\begin{array}{cl} K & \textbf{private key}\text{, sparse multiple of } P \\ \otimes & \\ 10011011\ldots10101011 & \big(m||\ldots||m\big) \oplus \mathsf{rand}_\gamma \oplus \mathsf{LFSR}_P \\ = 0100\ldots1101 & \big(\tilde{m}||\ldots||\tilde{m}\big) \oplus \mathsf{rand}_{\gamma^{\mathsf{w}(K)}} \end{array}$$

$\Rightarrow$ can compute $\tilde{m}$ (count majority), and recover m:

$$m \leftarrow \psi(\tilde{m})$$

with ψ a **linear map** defined by K

PRODUCT POLYNOMIAL $\otimes$ BITSTRING

Let $K = \sum k_i x^i$, and a bitstring $u = (u_0, \ldots, u_{\ell-1})$, then

$$K \otimes u = v,$$

with $|v| = \ell - \deg(K)$ bits, and

$$v_i = u_i k_0 + \cdots + u_{i+\deg(K)} k_{\deg(K)}$$

$\approx$ **sequence of dot products**

Properties exploited in decryption (recall $K = P \times Q$)

- $K \otimes (\text{LFSR}_P) = 0 \ldots 0$
- $K \otimes (\text{LFSR}_P \oplus \text{rand}_\gamma) \approx \text{rand}_{\gamma^{w(K)}}$

DECRYPTION

K	**private key**, sparse multiple of P
$\otimes$	
$10011011\ldots10101011$	$\big(m\|\|\ldots\|\|m\big) \oplus \mathsf{rand}_\gamma \oplus \mathsf{LFSR}_P$
$= 0100\ldots1101$	$\big(\tilde{m}\|\|\ldots\|\|\tilde{m}\big) \oplus \mathsf{rand}_{\gamma^{\mathsf{w}(K)}}$

$\Rightarrow$ can compute $\tilde{m}$ (count majority), and recover m:

$$m \leftarrow \psi(\tilde{m})$$

with ψ a **linear map** defined by K

DECRYPTION RELIABILITY

$\psi(m)$ repeated

$$N = \frac{\ell - \deg(K)}{|m|} \text{ times}$$

Decrypt incorrectly $\Leftrightarrow$ **majority logic fails** $\Leftrightarrow$ at least one bit of $\psi(m)$ is noised more than half the times.

$$\Pr[\text{bad decryption}] \approx |m| \cdot \varphi\Big(-\sqrt{\frac{N\gamma^{2w}}{1-\gamma^{2w}}}\Big)$$

with φ the CDF of $\mathcal{N}(0,1)$.

KEY GENERATION

Problem:
Find a pair (K, P), with K a **sparse multiple** of P, of **given degree and weight**, and P of degree in $[d_{\min}, d_{\max}]$.

Repeat

- pick a random K of given degree and weight
- factorize this K
- if K has a factor P of degree $\in [d_{\min}, d_{\max}]$, **return** (P, K)

(in practice $\deg(K) > 15\,000, \deg(P) > 5\,000$)

EXAMPLE OF PARAMETERS

For **80-bit security**,

- plaintext of $|m| = 128$ bits
- ciphertext of $\ell = 56\,000$ bits
- public-key is polynomial P of degree $\in [7\,150, 8\,000]$
- private-key is polynomial K of degree 24 500 and weight 51
- noise has bias 0.98
- decryption fails with probability 2^{-23}

SECURITY OF **TCHo**

PRIVATE KEY RECOVERY

Can decrypt if

- the private key K is known, **OR IF**
- another sparse multiple of degree $\leq \deg(K)$ is known

Computational problem LWPM

- Parameters: w, d, d_P, $0 < d_P < d$ and $w \ll d$.
- Instance: P of degree d_P
- Question: find a multiple of P of degree $\leq d$ and weight $\leq w$.

Strategies: exhaustive search, generalized birthday paradox, syndrome decoding.

$\Rightarrow$ for LWPM in time $\Omega(2^\lambda)$, need

$$\binom{d}{w-1} \leq 2^{d_P} \quad \text{and} \quad w \log \frac{d}{d_P} \geq \lambda$$

BASIC SECURITY PROPERTIES

TCHo...

- is **XOR-malleable**,

$$\mathsf{Enc}(m) \oplus \Delta = \mathsf{Enc}(m \oplus \Delta)$$

- can be **inverted** by a CCA adversary: given challenge ciphertext c, just query for $m \leftarrow \mathsf{Dec}(c \oplus \Delta)$, and recover original message $m \oplus \Delta$.

- can instanciate a **KEM** in hybrid encryption scheme, to provide IND-CCA security.

SEMANTIC SECURITY

Idea:
How to **distinguish** a ciphertext

$$C_1 = (m||\ldots||m) \oplus \mathrm{rand}_\gamma \oplus \mathrm{LFSR}_P,$$

from (uniform) random bits $C_2 = \mathrm{rand}_0$?

Compute $C_i \oplus m \Rightarrow$ reduces to distinguish $\mathrm{rand}_\gamma \oplus \mathrm{LFSR}_P$ from rand_0

Strategy:

- Directly distinguish $\mathrm{rand}_\gamma \oplus \mathrm{LFSR}_P$ from rand_0
- Find $\tilde{P}$, a sparse multiple of P, and distinguish

$$\tilde{P} \otimes (\mathrm{rand}_\gamma \oplus \mathrm{LFSR}_P) = \mathrm{rand}_{\gamma^{w(\tilde{P})}}$$

from

$$\tilde{P} \otimes \mathrm{rand}_0 = \mathrm{rand}_0$$

PERFORMANCES OF **TCHo**

BENCHMARKS' SETTINGS

- machine: **P4 3 GHz** cache 1 Mb (lasecpc15)
- `C++` code, compiled with `gcc 4.1.2`,
  flags `-O3 -march=pentium4`
- Use Shoup's **NTL** lib. for matrix operations and polynomial factorization (algo: Cantor-Zassenhaus, probabilistic)
- Timings given for **one message**, taking **average** values

RESULTS

Parameters:

- 128-bit plaintext, 54 Kb ciphertext
- $\deg(P) \in [7\,150, 8\,000]$, $\deg(K) = 24\,500$, $\mathrm{w}(K) = 51$

Encryption:

- 45 ms (without precomputation)
- < 1 ms (with precomputation of 54 Kb)

Decryption:

- 108 ms (without precomputation)
- 20 ms (with precomputation of ψ)

Key generation:

- ≈ 20 min

RESULTS

Parameters:

- 128-bit plaintext, 146 Kb ciphertext
- $\deg(P) \in [6\,000, 8\,795]$, $\deg(K) = 17\,600$, $\mathrm{w}(K) = 81$

Encryption:

- 100 ms (without precomputation)
- < 1 ms (with precomputation of 146 Kb)

Decryption:

- 260 ms (without precomputation)
- 130 ms (with precomputation of ψ)

Key generation:

- ≈ 2 min

PERFORMANCES

"Why do you give software timings for a **hardware** *cipher ?"*
$\rightarrow$ did not have the opportunity to implement HW.

Expected **much faster** on hardware devices, since

- Enc = only XOR and LFSR
- Dec = only GF(2) linear algebra
- parallelism is possible

Alternative **mode**: encrypt $\psi(m)$, decrypt directly to m
(negligible speed-up)

CONCLUSION

SUMMARY

TCHo is. . .

- based on the hardness of finding a sparse polynomial multiple
- quantum-cpu resistant
- semantically secure

FURTHER WORK

More experiments:

- benchmarks on FPGA, ASIC, etc.
- suitable for RFID tags ?

More analysis:

- speed-up key generation ?
- replace huge LFSR by. . . ?
- identify weak instances ?
- solve LWPM efficiently ?

Papers & slides available at
`lasecwww.epfl.ch`
`www.131002.net`

Asymmetric encryption with 2 XOR's: the cipher **TCHo**

Jean-Philippe Aumasson

n|w University of Applied Sciences Northwestern Switzerland
School of Engineering