

Snake Oil Crypto:

How I stopped to worry and started to love crypto

Team CIRCL

<https://www.d4-project.org/>

2019/11/27

Jean-Louis Huynen



- Cryptography 101,
- Cryptography and Network captures,
- D4 passiveSSL Collection,
- Leveraging OpenPGP metedata,
- Checking for weak crypto.

Cryptography 101

- **Plaintext** P: Text in clear,
- **Encryption** E: Process of disguising the plaintext to hide its content,
- **Ciphertext** C: Result of the Encryption process,
- **Decryption** D: Process of reverting encryption, transforming C into P,
- **Encryption Key** EK: Key to encrypt P into C,
- **Decryption Key** DK: Key to decrypt C into P,
- **Cryptanalysis**: Analysis of C to recover P without knowing K.

- **Confidentiality** : Ensure the secrecy of the message except for the **intended** recipient,
- **Authentication** : Proving a party's identity,
- **Integrity** : Verifying that data transmitted were not altered,
- **Non-repudiation** : Proving that the sender sent a given message.

- **In-transit encryption:** protects data while it is transferred from one machine to another,
- **At-rest encryption:** protects data stored on one machine.

It [cipher] should not require secrecy, and it should not be a problem if it falls into enemy hands.

There is no security in obscurity.

Black Box - Attackers may only see inputs / outputs:

- **Ciphertext-Only Attackers (COA)**: see only the ciphertext,
- **Known-Plaintext Attackers (KPA)**: see ciphertext and plaintext,
- **Chosen-Plaintext Attacker (CPA)**: encrypt plaintext, and see ciphertext,
- **Chosen-Ciphertext Attakers (CCA)**: encrypt plaintext, decrypt ciphertext.

Grey Box - Attackers see cipher's implementation:

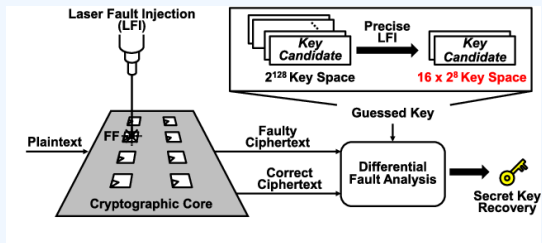
- **Side-Channel Attacks:** study the behavior of the implementation, for instance **timing attacks**¹:
 - ▶ Osvik, Shamir, Tromer [OSTo6]: Recover AES-256 secret key of Linux's dmccrypt in just 65 ms
 - ▶ AlFardan, Paterson [AFP13]: "Lucky13" recovers plaintext of CBC-mode encryption in pretty much all TLS implementations
 - ▶ Yarom, Falkner [YF14]: Attack against RSA-2048 in GnuPG 1.4.13: "On average, the attack is able to recover 96.7% of the bits of the secret key by observing a single signature or decryption round."
 - ▶ Benger, van de Pol, Smart, Yarom [BvdPSY14]: "reasonable level of success in recovering the secret key" for OpenSSL ECDSA using secp256k1 "with as little as 200 signatures"

Most recent one: tpm-fail [24420]

We discovered timing leakage on Intel firmware-based TPM (fTPM) as well as in STMicroelectronics' TPM chip. Both exhibit secret-dependent execution times during cryptographic signature generation. While the key should remain safely inside the TPM hardware, we show how this information allows an attacker to recover 256-bit private keys from digital signature schemes based on elliptic curves.

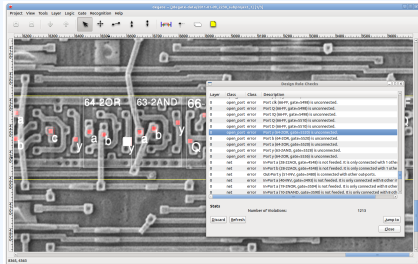
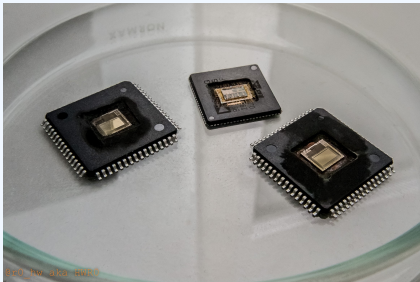
■ Invasive Attacks:

- ▶ injecting faults [MFS⁺18],



ATTACKERS MODEL V

- ▶ decapping chips², reverse engineering^{3 4}, etc.



¹<https://cryptojedi.org/peter/data/croatia-20160610.pdf>

²<https://siliconpron.org/wiki/doku.php?id=decap:start>

³<http://siliconzoo.org>

⁴<http://degate.org>



Cryptography and Network captures

D4 passiveSSL Collection

Leveraging OpenPGP metadata

Checking for weak crypto

IoT devices **are often the weakest devices** on a network:

- Usually the result of cheap engineering,
- sloppy patching cycles,
- sometimes forgotten—not monitored,
- few hardening features enabled.

We feel a bit safer when they use TLS, but should we?

⁵<https://github.com/d4-project/snake-oil-crypto>

Keep a log of links between:

- x509 certificates,
- ports,
- IP address,
- client (ja3),
- server (ja3s),

“JA3 is a method for creating SSL/TLS client fingerprints that should be easy to produce on any platform and can be easily shared for threat intelligence.”⁶

Pivot on additional data points during Incident Response

⁶<https://github.com/salesforce/ja3>

Collect and **store** x509 certificates and TLS sessions:

- Public keys type and size,
- moduli and public exponents,
- curves parameters.

Detect anti patterns in crypto:

- Moduli that share one prime factor,
- Moduli that share both prime factors, or private exponents,
- Small factors,
- Nonces reuse / common prefix or suffix, etc.

Focus on low hanging fruits that appeal to attackers

Researchers have shown that several devices generated their keypairs at boot time without enough entropy⁷:

```
prng.seed(seed)
p = prng.generate_random_prime()
// prng.add_entropy()
q = prng.generate_random_prime()
n = p*q
```

Given $n=pq$ and $n' = pq'$ it is trivial to recover the shared p by computing their **Greatest Common Divisor (GCD)**, and therefore **both private keys**⁸.

⁷Bernstein, Heninger, and Lange: <http://facthacks.cr.yp.to/>

⁸<http://www.loyalty.org/~schoen/rsa/>

In Snake-Oil-Crypto we compute GCD⁹ between:


- between certificates having the same issuer,
- between certificates having the same subject,
- on keys collected from various sources (PassiveSSL, Certificate Transparency, shodan, censys, etc.),


“Check all the keys that we know of for vendor X”

⁹using Bernstein's Batch GCD algorithm

2019-11-08

Name: crypto-material 

References: 0 

Referenced by: 6 

uses Object 13800 (network: x509)

uses Object 13801 (network: x509)

uses Object 13802 (network: x509)

uses Object 13803 (network: x509)

uses Object 13804 (network: x509)

uses Object 13805 (network: x509)

<input type="checkbox"/>	2019-11-08	Other	p: text	12732045980491482532629620809854872609730718866846479950748763 99251101386987265586481573653124576541684265313376164608426942 4192867704218331356123978614869
<input type="checkbox"/>	2019-11-08	Other	q: text	None
<input type="checkbox"/>	2019-11-08	Other	rsa-modulus-size: text	1024
<input type="checkbox"/>	2019-11-08	Other	type: text	RSA

The MISP feed:

- **Allows** for checking automatic checking by an IDS on hashed values,
- **contains** thousands on broken keys from a dozen of vendors,
- **will be accessible upon request (info@circl.lu).**

In the future:

- **Automatic** the vendor checks by performing TF-IDF on x509's subjects,
- **automatic** vendors notification.

- ✓ sensor-d4-tls-fingerprinting ¹⁰: **Extracts** and **fingerprints** certificates, and **computes** TLSH fuzzy hash.
- ✓ analyzer-d4-passivessl ¹¹: **Stores** Certificates / PK details in a PostgreSQL DB.
- snake-oil-crypto ¹²: **Performs** crypto checks, push results in MISP for notification
- lookup-d4-passivessl ¹³: **Exposes** the DB through a public REST API.

¹⁰github.com/D4-project/sensor-d4-tls-fingerprinting





¹¹github.com/D4-project/analyzer-d4-passivessl

¹²github.com/D4-project/snake-oil-crypto




¹³github.com/D4-project/lookup-d4-passivessl



GET IN TOUCH IF YOU WANT TO JOIN/SUPPORT THE PROJECT, HOST A PASSIVE SSL SENSOR OR CONTRIBUTE

- Collaboration can include research partnership, sharing of collected streams or improving the software.
- Contact: info@circl.lu
- <https://github.com/D4-Project> -
https://twitter.com/d4_project

-  *TPM-FAIL: TPM MEETS TIMING AND LATTICE ATTACKS*, 29TH USENIX SECURITY SYMPOSIUM (USENIX SECURITY 20) (BOSTON, MA), USENIX ASSOCIATION, AUGUST 2020.
-  NADHEM J. AL FARDAN AND KENNETH G. PATERSON, *LUCKY THIRTEEN: BREAKING THE TLS AND DTLS RECORD PROTOCOLS*, PROCEEDINGS OF THE 2013 IEEE SYMPOSIUM ON SECURITY AND PRIVACY (WASHINGTON, DC, USA), SP '13, IEEE COMPUTER SOCIETY, 2013, PP. 526–540.
-  ROSS J. ANDERSON, *SECURITY ENGINEERING: A GUIDE TO BUILDING DEPENDABLE DISTRIBUTED SYSTEMS*, 2 ED., WILEY PUBLISHING, 2008.
-  JEAN-PHILIPPE AUMASSON, *SERIOUS CRYPTOGRAPHY: A PRACTICAL INTRODUCTION TO MODERN ENCRYPTION*, NO STARCH PRESS, 2017.

REFERENCES II

-  NAOMI BENGER, JOOP VAN DE POL, NIGEL P. SMART, AND YUVAL YAROM, “OOH AAH... JUST A LITTLE BIT” : A SMALL AMOUNT OF SIDE CHANNEL CAN GO A LONG WAY, CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS – CHES 2014 (BERLIN, HEIDELBERG) (LEJLA BATINA AND MATTHEW ROBshaw, EDS.), SPRINGER BERLIN HEIDELBERG, 2014, PP. 75–92.
-  DIETER GOLLMANN, *COMPUTER SECURITY* (3. ED.), WILEY, 2011.
-  KOHEI MATSUDA, TATSUYA FUJII, NATSU SHOJI, TAKESHI SUGAWARA, KAZUO SAKIYAMA, YU-ICHI HAYASHI, MAKOTO NAGATA, AND NORIYUKI MIURA, A 286 F2/CELL DISTRIBUTED BULK-CURRENT SENSOR AND SECURE FLUSH CODE ERASER AGAINST LASER FAULT INJECTION ATTACK ON CRYPTOGRAPHIC PROCESSOR, *IEEE JOURNAL OF SOLID-STATE CIRCUITS* **53** (2018), NO. 11, 3174–3182.
-  ALFRED J. MENEZES, SCOTT A. VANSTONE, AND PAUL C. VAN OORSCHOT, *HANDBOOK OF APPLIED CRYPTOGRAPHY*, 1ST ED., CRC PRESS, INC., BOCA RATON, FL, USA, 1996.

-  DAG ARNE OSVIK, ADI SHAMIR, AND ERAN TROMER, *CACHE ATTACKS AND COUNTERMEASURES: THE CASE OF AES*, TOPICS IN CRYPTOLOGY – CT-RSA 2006 (BERLIN, HEIDELBERG) (DAVID POINTCHEVAL, ED.), SPRINGER BERLIN HEIDELBERG, 2006, PP. 1–20.
-  YUVAL YAROM AND KATRINA FALKNER, *FLUSH+RELOAD: A HIGH RESOLUTION, LOW NOISE, L3 CACHE SIDE-CHANNEL ATTACK*, 23RD USENIX SECURITY SYMPOSIUM (USENIX SECURITY 14) (SAN DIEGO, CA), USENIX ASSOCIATION, AUGUST 2014, PP. 719–732.