From ROCA (Fun & troubles with RSA keypairs) to improved security certification

seccerts

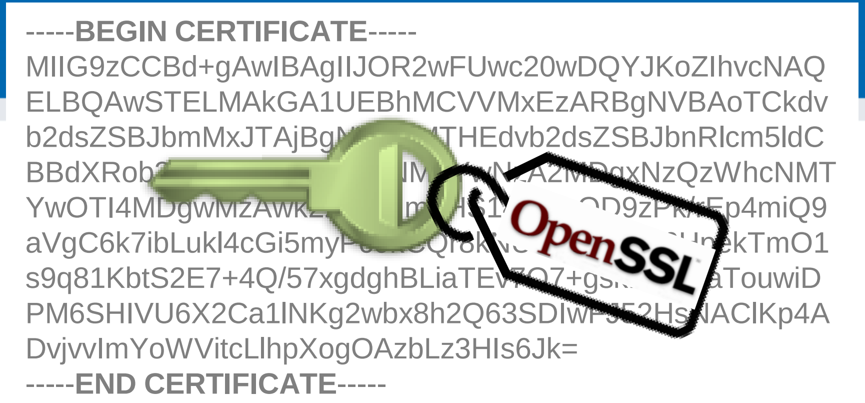
Vashek Matyáš ***matyas@fi.muni.cz*** ***@rngsec***

Centre for Research on Cryptography and Security, Masaryk University

Joint work with: *Petr Švenda (credits for slides!), Matúš Nemec, Marek Sýs, Dušan Klinec, Jan Jančár, Adam Janovský, Peter Sekan, Rudolf Kvašnovský, David Formánek, David Komárek and others*

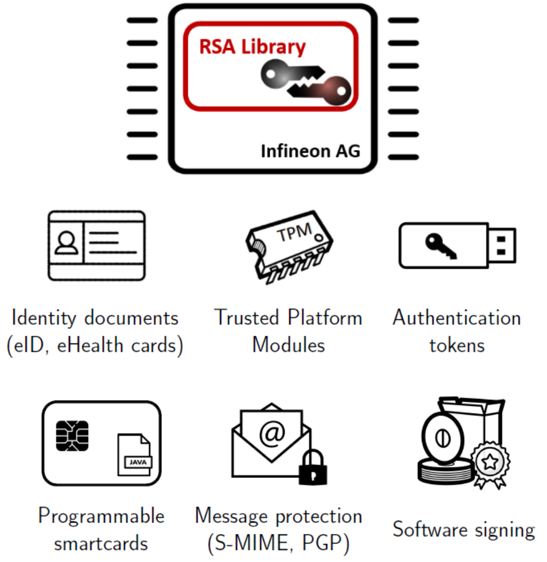
Overview

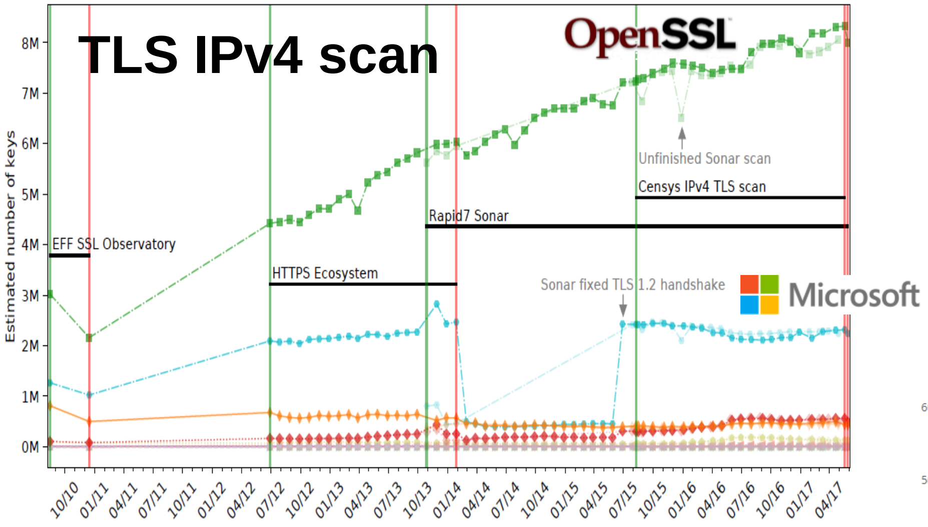
- Motivation: information leakage in RSA public keys
- Learning phase: analysis of large number of RSA keypairs
- Applications of classification capability
- Smartcards and RSA keypair generation
- **Security certification and possible improvements**



```
-----BEGIN CERTIFICATE-----
MIIG9zCCBd+gAwIBAgIIJOR2wFUwc20wDQYJKoZIhvcNAQ
ELBQAwSTELMAkGA1UEBhMCVVMxEzARBgNVBAoTCkdv
b2dsZSBJbmMxJTAjBg[illegible]THEdvb2dsZSBJbnRlcm5ldC
BBdXRob[illegible]IM[illegible]NzA2MDgxNzQzWhcNMT
YwOTI4MDgwMzAwk[illegible]m[illegible]1[illegible]OD9zPk[illegible]Ep4miQ9
aVgC6k7ibLukl4cGi5myF[illegible]QTrok[illegible]kTmO1
s9q81KbtS2E7+4Q/57xgdghBLiaTEv7O7+gs[illegible]aTouwiD
PM6SHIVU6X2Ca1INKg2wbx8h2Q63SDIwF[illegible]2Hs[illegible]ACIKp4A
DvjvvImYoWVitcLlhpXogOAzbLz3HIs6Jk=
-----END CERTIFICATE-----
```

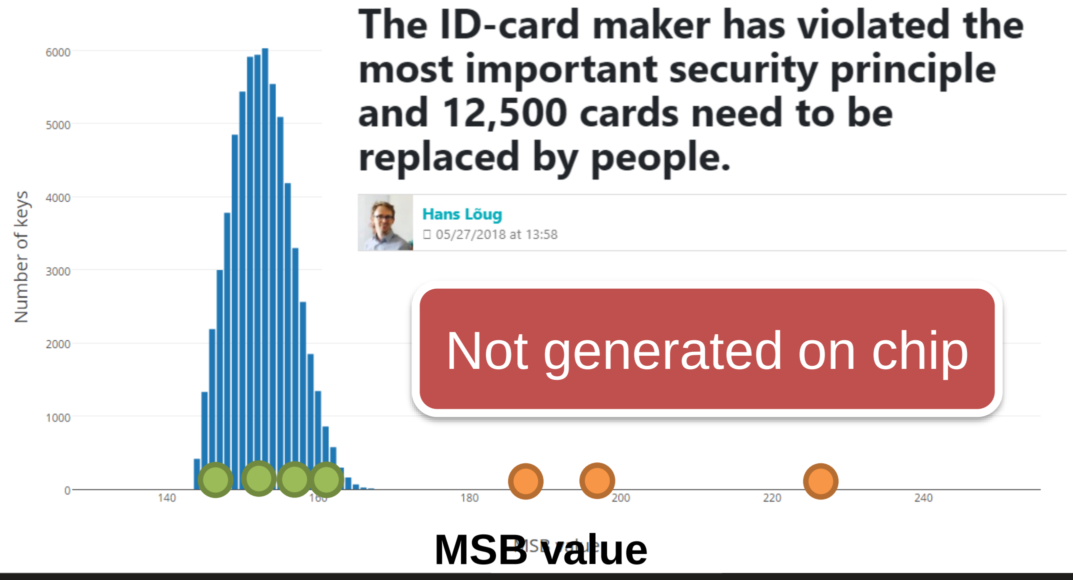
ROCA: factorable RSA (CRoCS, 10/2017)





Popularity of libraries (CRoCS, 11/2017)

EE eID injected keys (Arnis Paršovs, 05/2018)



The ID-card maker has violated the most important security principle and 12,500 cards need to be replaced by people.

Hans Lõug
05/27/2018 at 13:58

Not generated on chip

Single points of failure

zerodayclothing.com

- We already try to avoid single points of failure at many places
  - – Personal: dual control, people from different backgrounds…
  - – Technical: Load-balancing web servers, RAID, periodic backups…
  - – Supply chain: no reliance on single supplier…
- Problems: Appropriate trade-off between security, cost and usability

- Typical process
  1. (Hidden) existence of single point of failure
  2. System once failed => analysis => identification of point of failure
  3. Mitigate for the next time => redundancy, removal of single point of failure
- Problem: What if failure is very rare, but with disastrous impact?

RSA primer – what does it mean and why should I care?

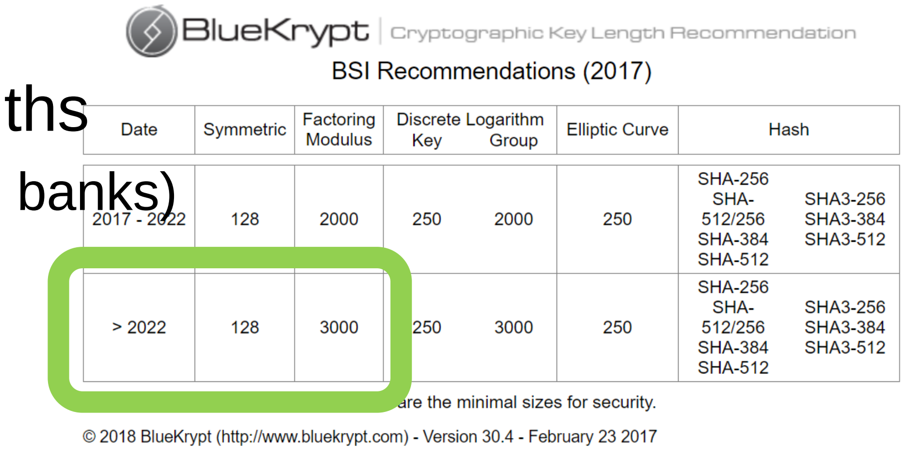
- RSA is widely used public-key cryptosystem (1977)
- Used for digital signatures (mail, software distribution, contracts…)
- Used for key exchange (HTTPS/TLS, PGP…)
- Private part: random primes P and Q, private exponent d
- Public part: public exponent e (often 65,537), modulus N

$$P \times Q = N$$

- Factorization attack: compute primes P and Q from the knowledge of N
- Problem: How to generate a large prime (1024- or 2048-bit length)?

RSA is much more than a description of basic algorithm

- Proper key lengths, key management, secure and optimized implementation
- Certifications, adaptation to changes…
- RSA security over time:
  - 512 bits originally assumed secure, now 2048 phased out in 2022 (BSI)
  - Faster factorization algorithms (NFS) with faster machines, quantum computers
  - Design and coding flaws, faulty TRNG, side-channel attacks, padding oracles…
- BTW: Banks are still using very short RSA key lengths
  - 768 & 896 bits (7 out of 11 tested EMV cards issued by EU banks)
  - No security margin for almost any problem



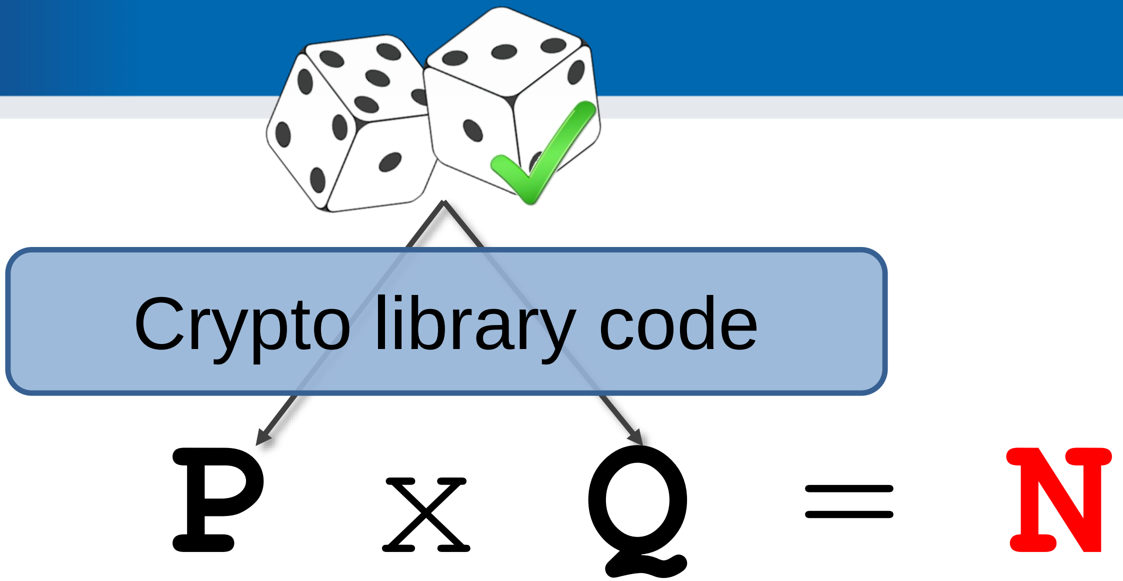
BlueKrypt | Cryptographic Key Length Recommendation

BSI Recommendations (2017)

Date	Symmetric	Factoring Modulus	Discrete Logarithm		Elliptic Curve	Hash	
			Key	Group			
2017 - 2022	128	2000	250	2000	250	SHA-256 SHA-512/256 SHA-384 SHA-512	SHA3-256 SHA3-384 SHA3-512
> 2022	128	3000	250	3000	250	SHA-256 SHA-512/256 SHA-384 SHA-512	SHA3-256 SHA3-384 SHA3-512

are the minimal sizes for security.

© 2018 BlueKrypt (http://www.bluekrypt.com) - Version 30.4 - February 23 2017



RSA public key

N = 9782D7123C330444C88E279BF321EE84AC39524F1D84026327B04F32E1E930FC81588010178DC75FCBF8258A068071317245D08817988813C4173495A922A41DA429A964F738020076EFFE7ED5811088873C6E58EEF1CDC9005966 E72368B51A821FC699E9C3FD66B377E2DF2485DC401DD99CC125890E5D969A6AC8B

e = 10001

OpenSSL
Infineon
Crypto Java Ca

Our initial motivation (2014)

- Long relationship with smartcards, JavaCards and FOSS
  - – Analysis for Czech National Security Authority (2002-2009)
  - – JCAlgTest.org, JCMathLib, CesTa, JCProfiler, curated list of JC apps…
- Cryptographic smartcards are pervasive (SIM, EMV, eID, tokens…)
- Yet smartcard industry is very closed
  - – NDA just to see detailed specifications, proprietary APIs, no design details…
- Idea (2014):
  - – Take cards we have at lab and bunch of open-source libraries
  - – Generate large number of RSA keypairs and compare similarities
  - – Infer the implementation of RSA key generation and spot problems

Common Criteria

FIPS VALIDATED 140-2 ™

Analysis of large number of RSA keys

LEARNING PHASE

https://seccerts.org/

22 software libraries and versions

ARM mbed
FlexiProvider
Harnessing the power of the Java Cryptography Architecture
Nettle
GnuPG
cryptlib
JAVA
OpenSSL
OpenJDK
GNU CRYPTO
STRONG CRYPTIX CRYPTO
Microsoft
wolfSSL
OPEN SOURCE, OPEN ACADEMIA, OPEN MINDS
LibTom PROJECTS
Botan
Crypto++
PGP
FIPS VALIDATED 140-2

16 types of smart

G&D Crypto Java Card

Infin Crypto

Gemalto Crypto Java Card

NXP Crypto

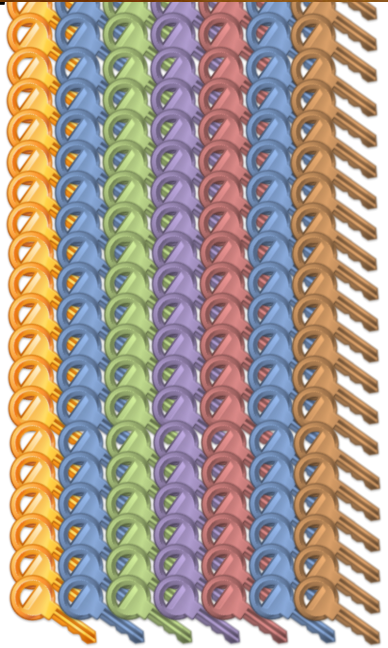
Oberthur Crypto Java Card

Feitian Crypto Java Card

1 000 000 x

```
Gen_RSA_keypair()
```

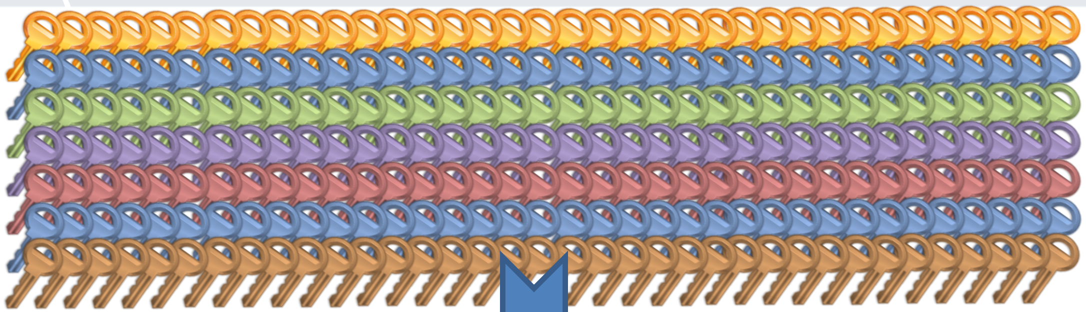
60+ million RSA ke



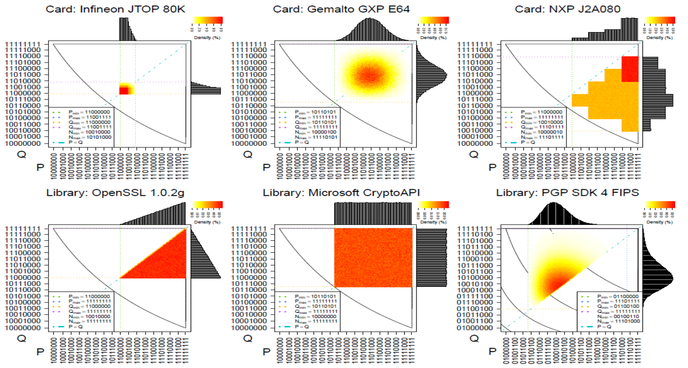
60+ million fresh RSA keypairs (P, Q, N)

https://seccerts.org/

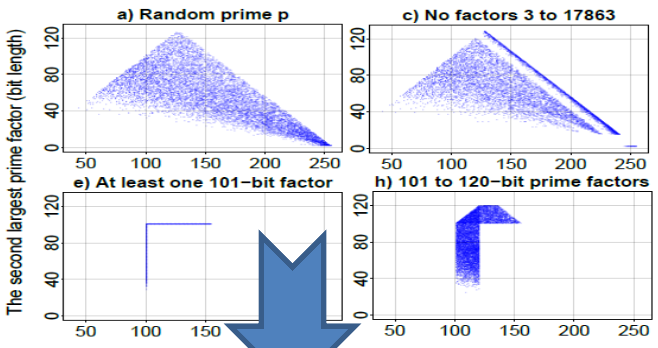
22 sw. libraries
16 smart cards



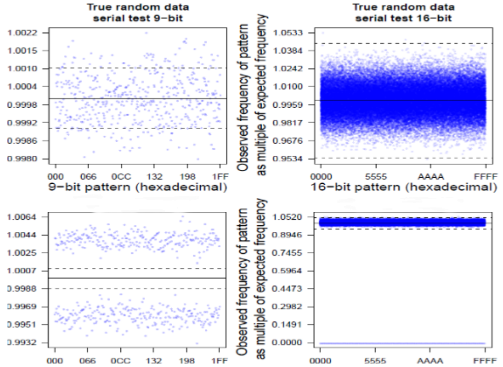
Distribution of primes (MSB)



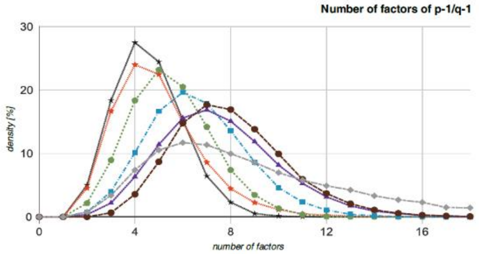
Large factors of p-1 / p+1



Bit stream statistics



Number of factors

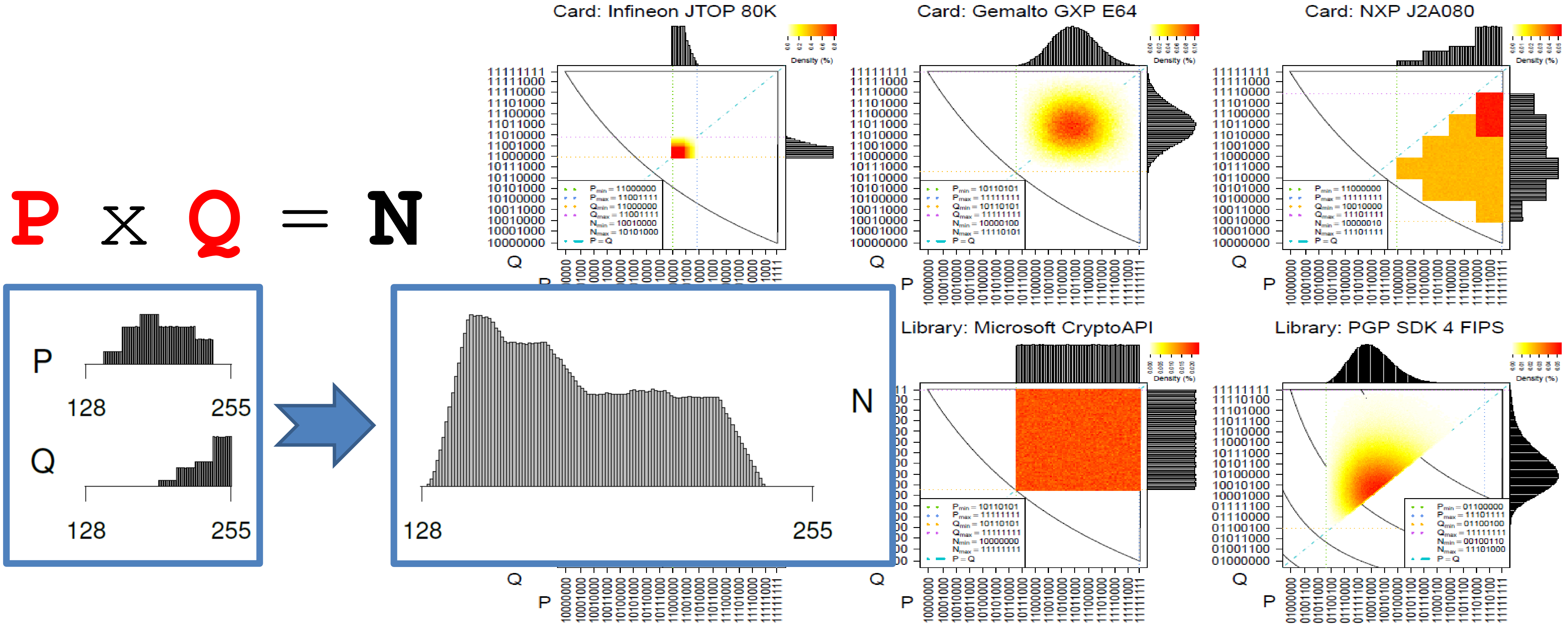


and more…

- Various implementation choices to generate large primes P & Q
- Small bias, but enough to attribute public key to particular library
  - Best paper award at USENIX Security 2016

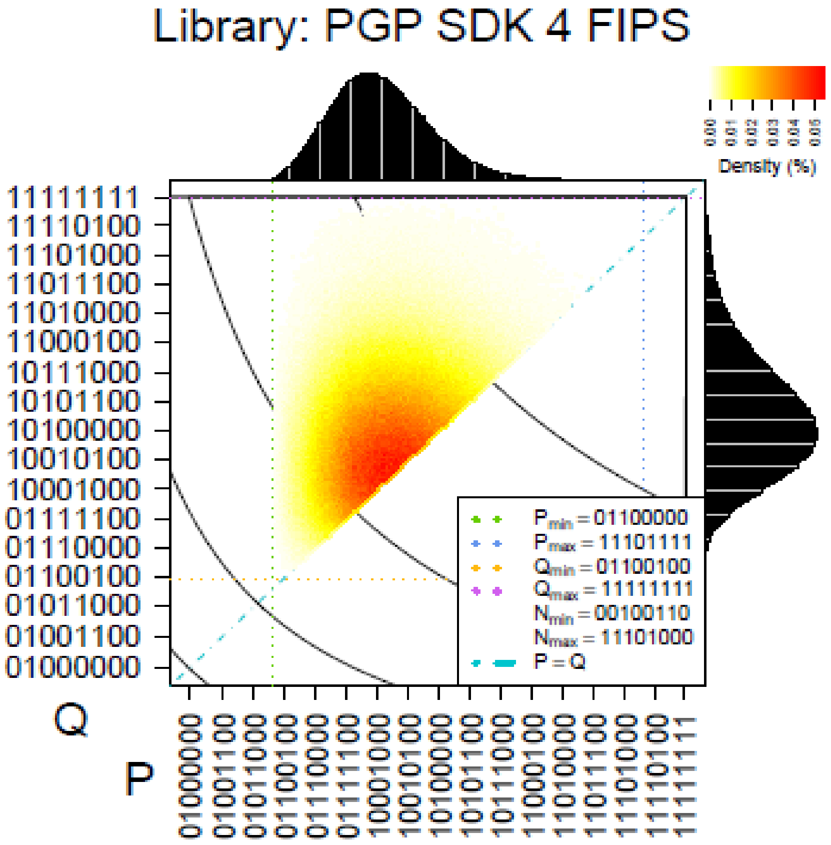
usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

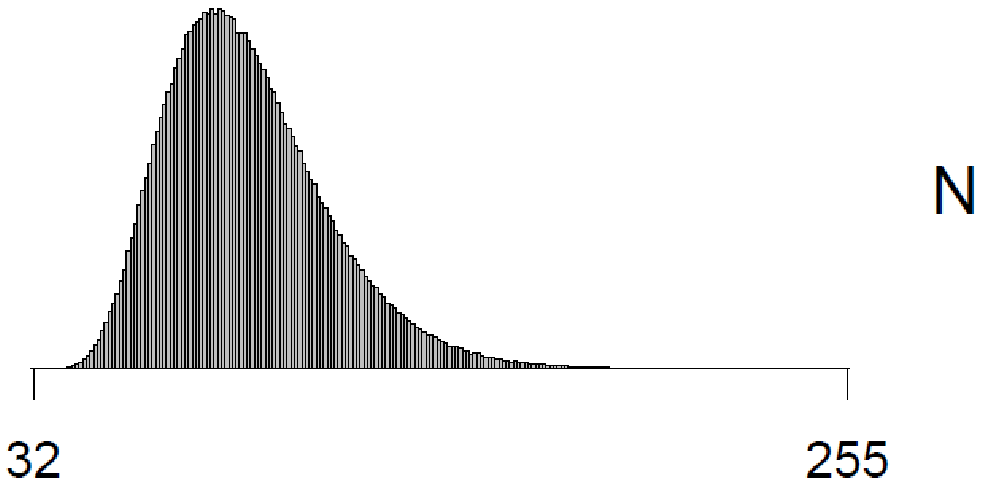
Heatmap of primes’ most significant byte



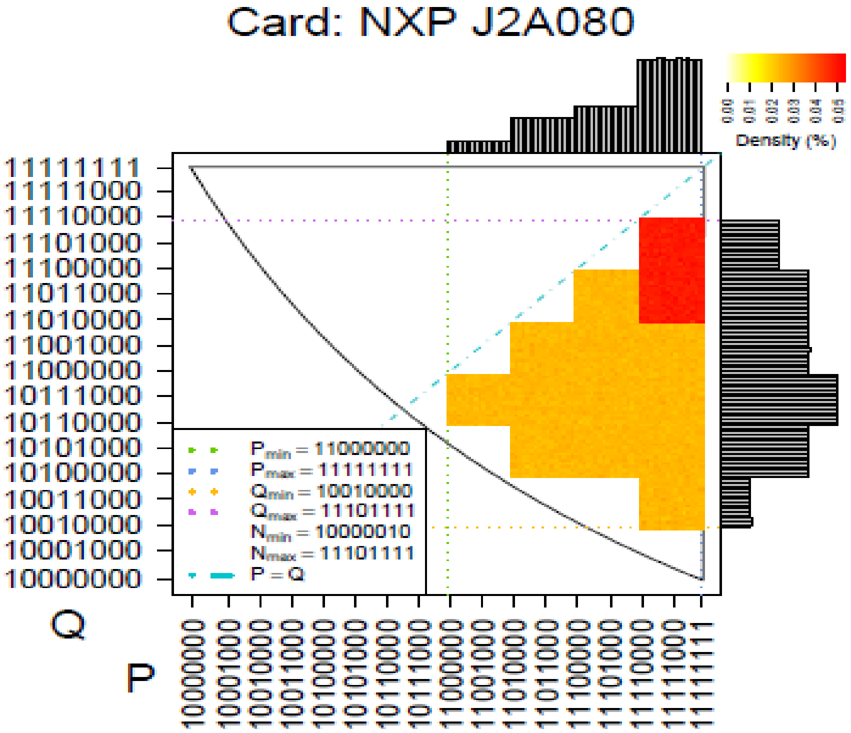
P x Q = N

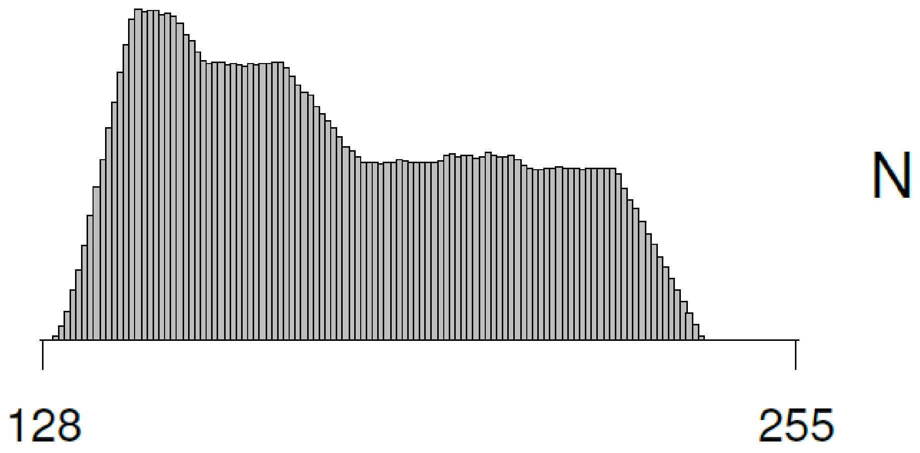
Wide diversity of modulus MSB distribution observed



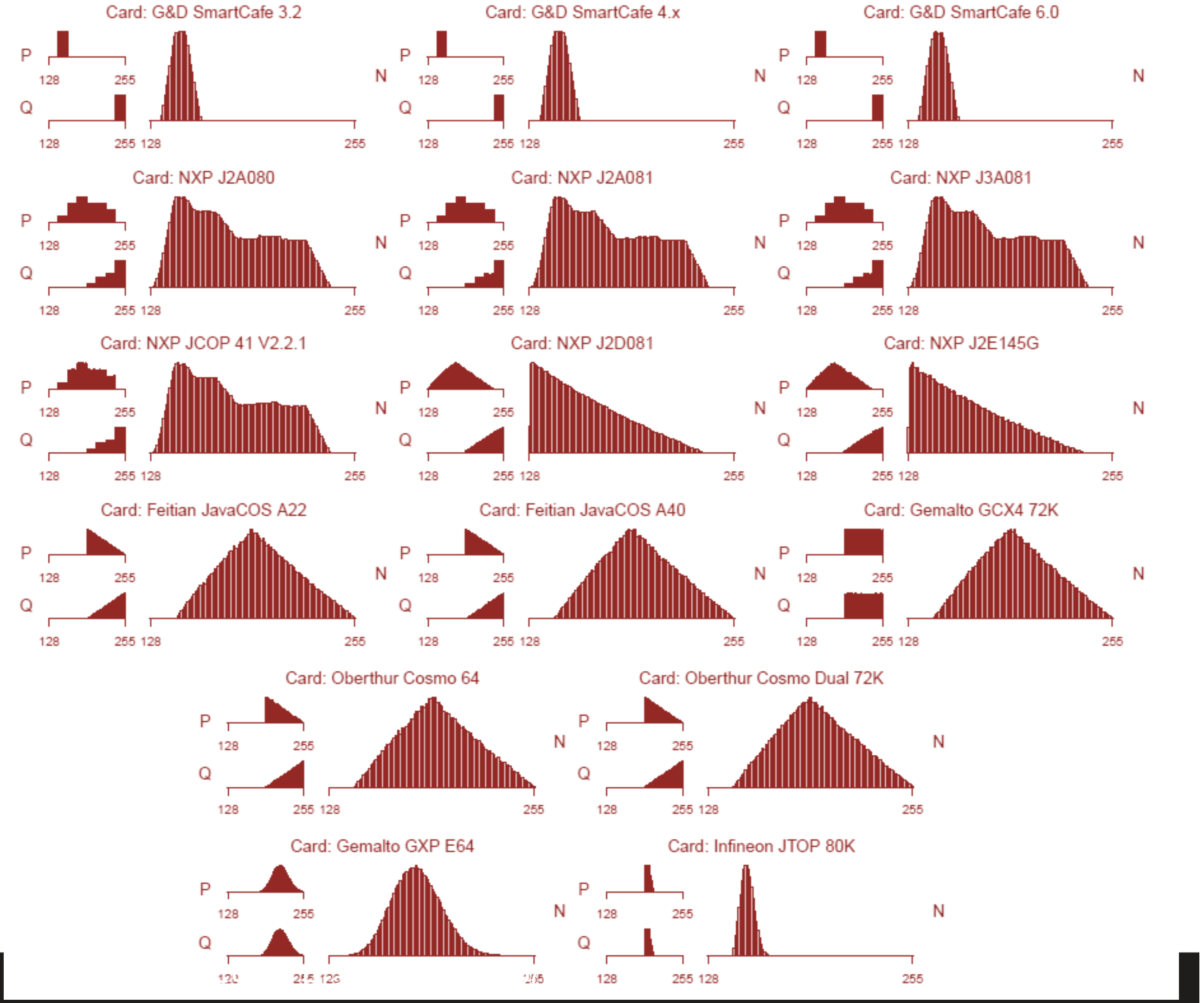


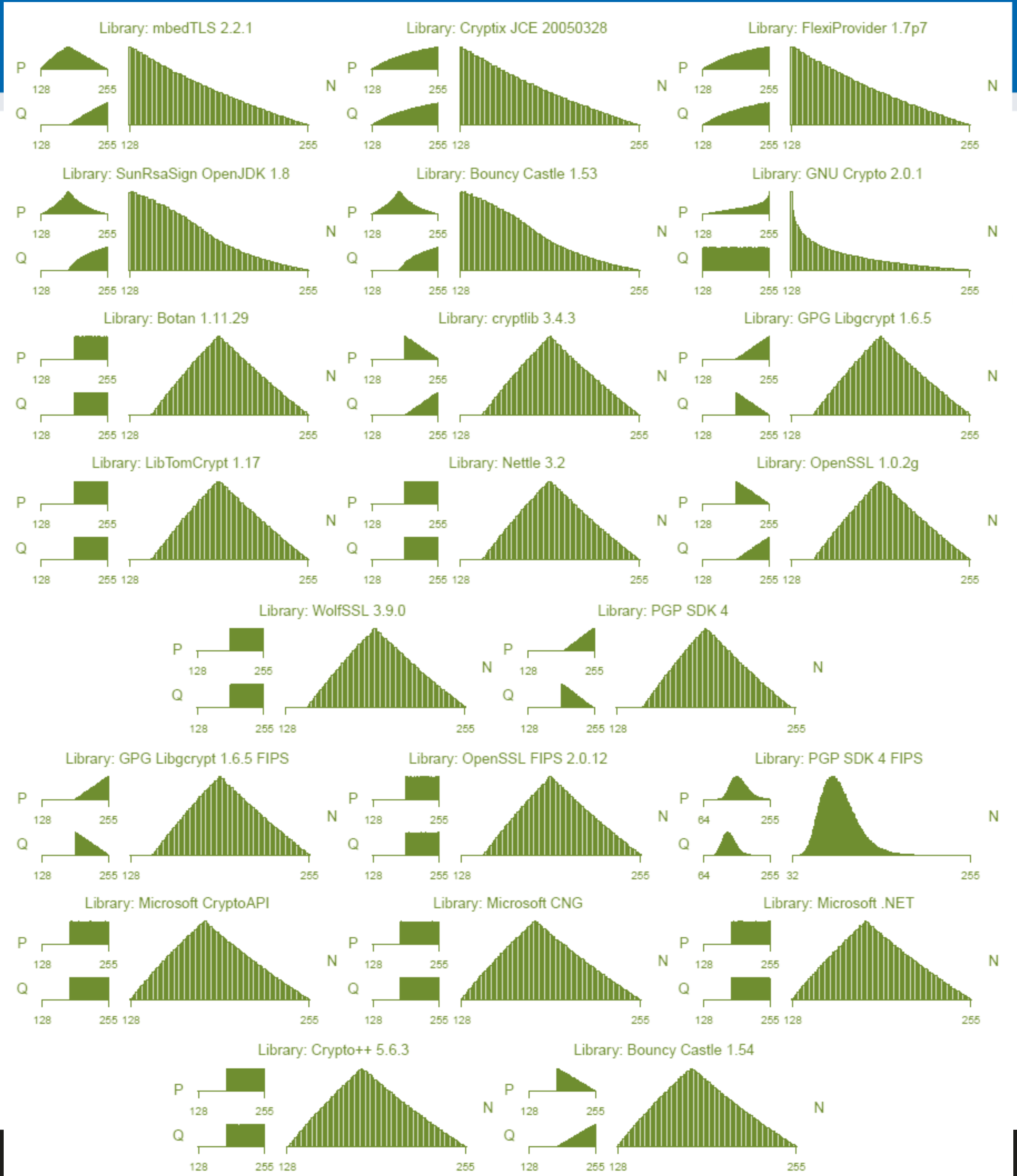
Wide diversity of modulus MSB distribution observed



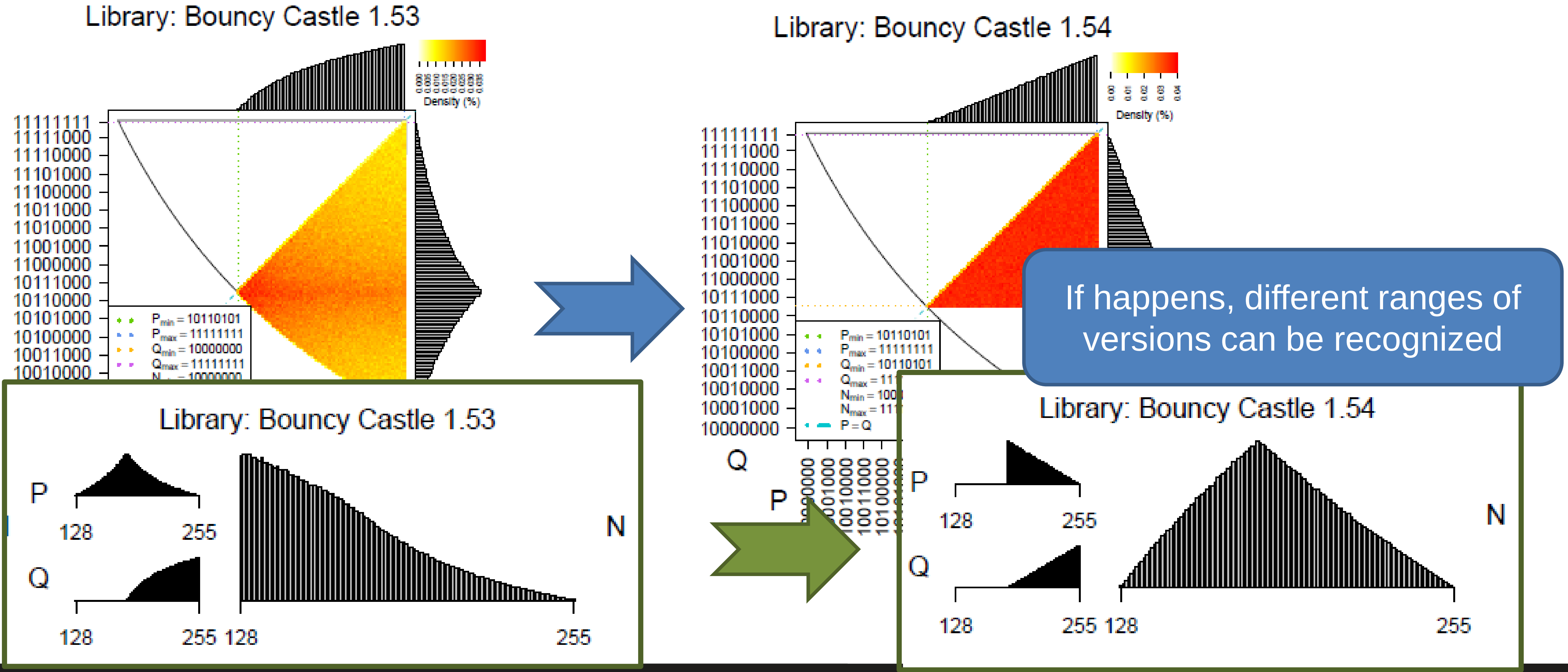


MSB of modulus – libs/cards


Card: G&D SmartCafe 3.2
Card: G&D SmartCafe 4.x
Card: G&D SmartCafe 6.0
Card: NXP J2A080
Card: NXP J2A081
Card: NXP J3A081
Card: NXP JCOP 41 V2.2.1
Card: NXP J2D081
Card: NXP J2E145G
Card: Feitian JavaCOS A22
Card: Feitian JavaCOS A40
Card: Gemalto GCX4 72K
Card: Oberthur Cosmo 64
Card: Oberthur Cosmo Dual 72K
Card: Gemalto GXP E64
Card: Infineon JTOP 80K

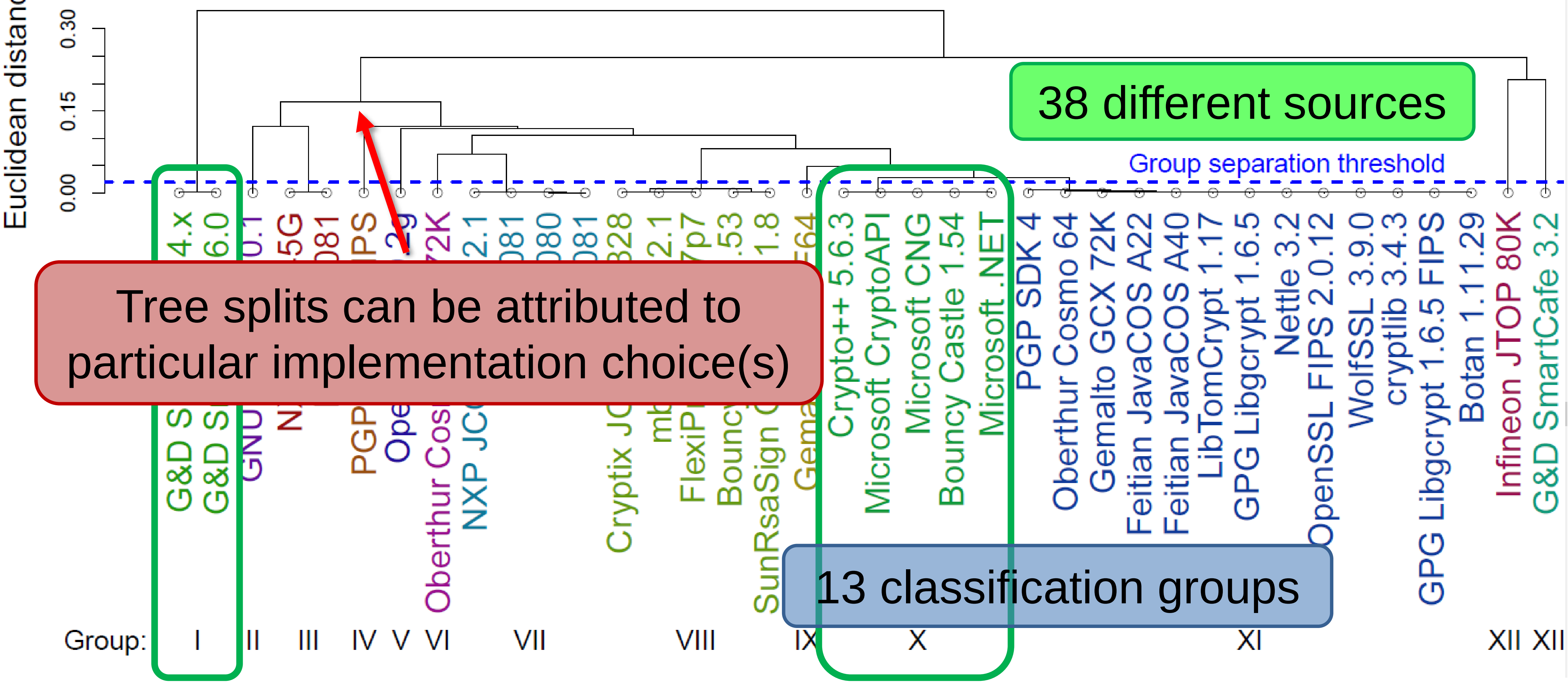

Library: mbedTLS 2.2.1
Library: Cryptix JCE 20050328
Library: FlexiProvider 1.7p7
Library: SunRsaSign OpenJDK 1.8
Library: Bouncy Castle 1.53
Library: GNU Crypto 2.0.1
Library: Botan 1.11.29
Library: cryptlib 3.4.3
Library: GPG Libgcrypt 1.6.5
Library: LibTomCrypt 1.17
Library: Nettle 3.2
Library: OpenSSL 1.0.2g
Library: WolfSSL 3.9.0
Library: PGP SDK 4
Library: GPG Libgcrypt 1.6.5 FIPS
Library: OpenSSL FIPS 2.0.12
Library: PGP SDK 4 FIPS
Library: Microsoft CryptoAPI
Library: Microsoft CNG
Library: Microsoft .NET
Library: Crypto++ 5.6.3
Library: Bouncy Castle 1.54

Occasional change with library/device revision



If happens, different ranges of versions can be recognized

Similarity of analyzed sources (classification groups)

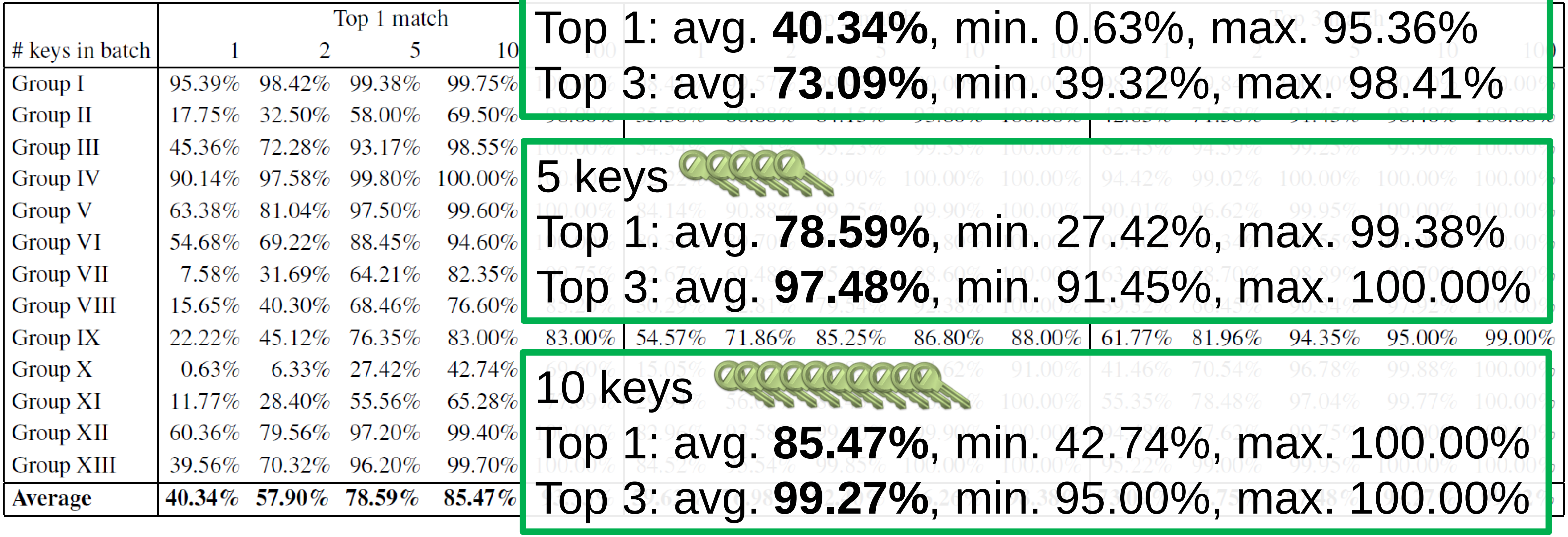


38 different sources

Tree splits can be attributed to particular implementation choice(s)

13 classification groups

Classification accuracy (test set, 10k keys/source)



# keys in batch	Top 1 match					[illegible]					Top 3 match				
	1	2	5	10	100	1	2	5	10	100	1	2	5	10	100
Group I	95.39%	98.42%	99.38%	99.75%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Group II	17.75%	32.50%	58.00%	69.50%	98.00%	35.58%	80.88%	84.15%	93.80%	100.00%	42.85%	71.58%	91.45%	98.40%	100.00%
Group III	45.36%	72.28%	93.17%	98.55%	100.00%	54.34%	[illegible]	95.23%	99.53%	100.00%	82.45%	94.59%	99.25%	99.90%	100.00%
Group IV	90.14%	97.58%	99.80%	100.00%	[illegible]	[illegible]	[illegible]	99.90%	100.00%	100.00%	94.42%	99.02%	100.00%	100.00%	100.00%
Group V	63.38%	81.04%	97.50%	99.60%	100.00%	84.14%	90.88%	99.25%	99.90%	100.00%	90.01%	96.62%	99.95%	100.00%	100.00%
Group VI	54.68%	69.22%	88.45%	94.60%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Group VII	7.58%	31.69%	64.21%	82.35%	99.75%	32.67%	69.48%	[illegible]	98.60%	100.00%	[illegible]	88.70%	98.89%	99.70%	[illegible]
Group VIII	15.65%	40.30%	68.46%	76.60%	[illegible]	30.29%	[illegible]	79.54%	92.38%	100.00%	59.32%	66.45%	90.54%	97.92%	100.00%
Group IX	22.22%	45.12%	76.35%	83.00%	83.00%	54.57%	71.86%	85.25%	86.80%	88.00%	61.77%	81.96%	94.35%	95.00%	99.00%
Group X	0.63%	6.33%	27.42%	42.74%	69.60%	15.05%	[illegible]	[illegible]	[illegible]	91.00%	41.46%	70.54%	96.78%	99.88%	100.00%
Group XI	11.77%	28.40%	55.56%	65.28%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	100.00%	55.35%	78.48%	97.04%	99.77%	100.00%
Group XII	60.36%	79.56%	97.20%	99.40%	[illegible]	82.96%	[illegible]	[illegible]	99.90%	[illegible]	[illegible]	97.62%	99.75%	[illegible]	[illegible]
Group XIII	39.56%	70.32%	96.20%	99.70%	100.00%	84.52%	[illegible]	99.85%	100.00%	100.00%	95.22%	99.00%	99.95%	100.00%	100.00%
Average	**40.34%**	**57.90%**	**78.59%**	**85.47%**	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

1 key

Top 1: avg. **40.34%**, min. 0.63%, max. 95.36%

Top 3: avg. **73.09%**, min. 39.32%, max. 98.41%

5 keys

Top 1: avg. **78.59%**, min. 27.42%, max. 99.38%

Top 3: avg. **97.48%**, min. 91.45%, max. 100.00%

10 keys

Top 1: avg. **85.47%**, min. 42.74%, max. 100.00%

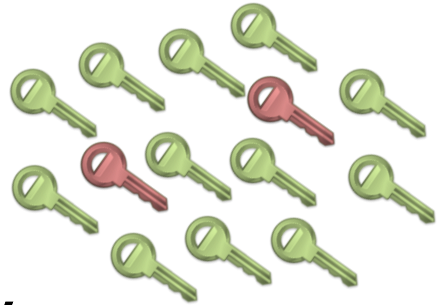
Top 3: avg. **99.27%**, min. 95.00%, max. 100.00%

How we can use classification in real world?

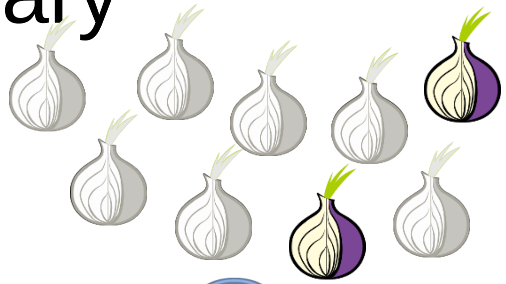
APPLICATION OF CLASSIFICATION

Impact (of the possibility) of public key classification

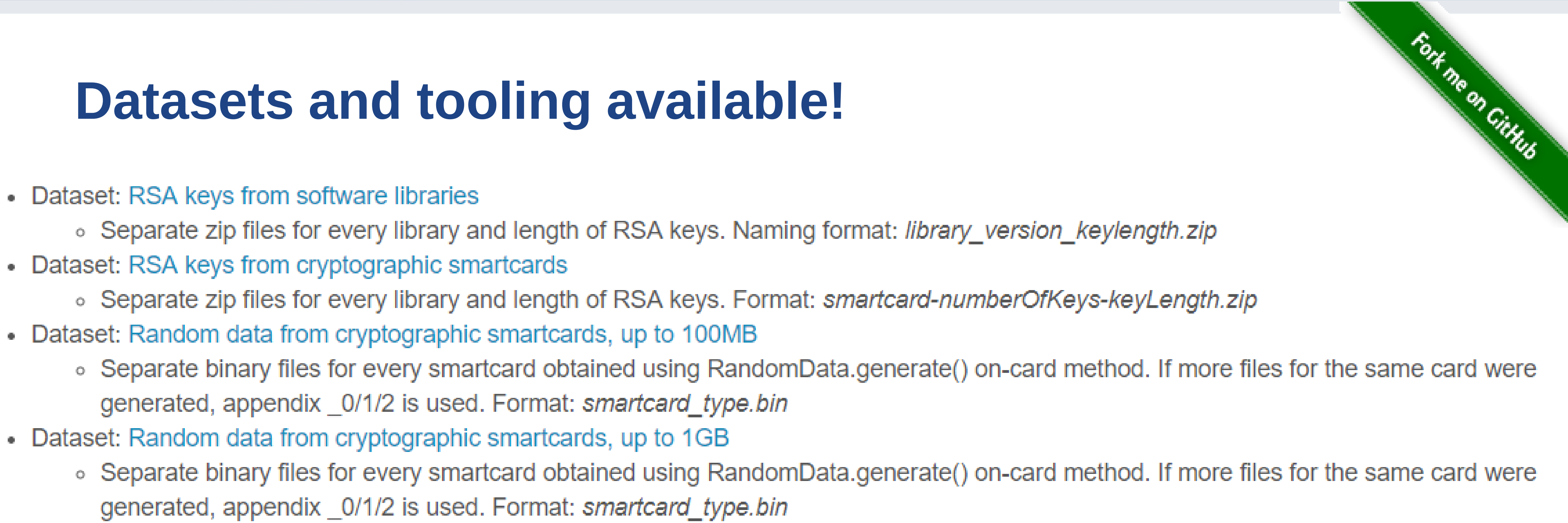
- Information leakage vulnerability



- Statistics: current usage trends (TLS/SSH…)
- Quick search for other keys from vulnerable library
- Forensics: source lib/device of weak keys
- De-anonymization: linking Tor hidden services
- Audit: identify source libs in target organization



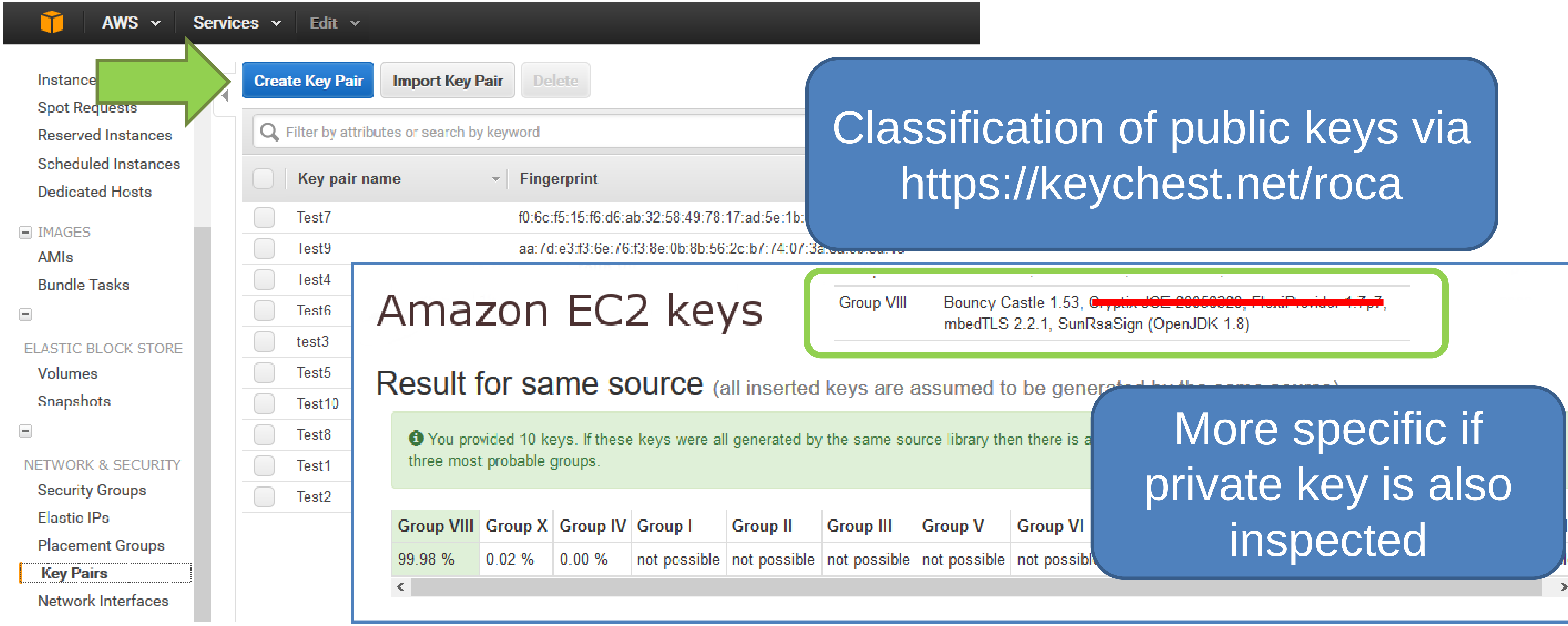
Datasets and tooling available!



- Dataset: RSA keys from software libraries
  - Separate zip files for every library and length of RSA keys. Naming format: *library_version_keylength.zip*
- Dataset: RSA keys from cryptographic smartcards
  - Separate zip files for every library and length of RSA keys. Format: *smartcard-numberOfKeys-keyLength.zip*
- Dataset: Random data from cryptographic smartcards, up to 100MB
  - Separate binary files for every smartcard obtained using RandomData.generate() on-card method. If more files for the same card were generated, appendix _0/1/2 is used. Format: *smartcard_type.bin*
- Dataset: Random data from cryptographic smartcards, up to 1GB
  - Separate binary files for every smartcard obtained using RandomData.generate() on-card method. If more files for the same card were generated, appendix _0/1/2 is used. Format: *smartcard_type.bin*

- We are still extending database of libraries and devices
  - If you have access to unlisted one (e.g., HSM, closed-source lib…), let us know
  - We need (ideally) 1million 512b RSA keypairs + 10k 1024/2048b for verification

Audit: What Amazon EC2 uses to generate RSA keys?



Key pair name	Fingerprint
Test7	f0:6c:f5:15:f6:d6:ab:32:58:49:78:17:ad:5e:1b:
Test9	aa:7d:e3:f3:6e:76:f3:8e:0b:8b:56:2c:b7:74:07:3a
Test4	
Test6	
test3	
Test5	
Test10	
Test8	
Test1	
Test2	

Classification of public keys via https://keychest.net/roca

Amazon EC2 keys

Group VIII	Bouncy Castle 1.53, ~~Cryptix JCE 20050328, FlexiProvider 1.7p7,~~ mbedTLS 2.2.1, SunRsaSign (OpenJDK 1.8)

Result for same source (all inserted keys are assumed to be generated by the same source)

You provided 10 keys. If these keys were all generated by the same source library then there is a three most probable groups.

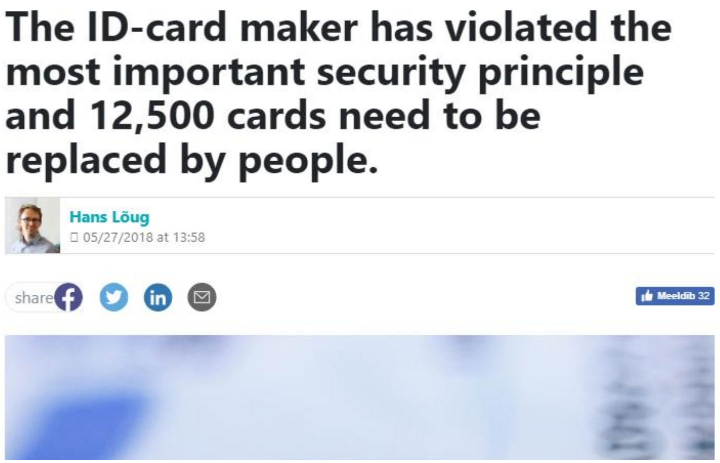
Group VIII	Group X	Group IV	Group I	Group II	Group III	Group V	Group VI
99.98 %	0.02 %	0.00 %	not possible	not possible	not possible	not possible	not possibl

More specific if private key is also inspected

EE eID injected keys
(Arnis Paršovs, 05/2018)

A problem reported from Estonia (17.5.2018)

- Estonian eIDs generate private key always on chip (by design)
  - Some keys found to be injected from outside
- Found by observed discrepancy in public key properties (MSB)

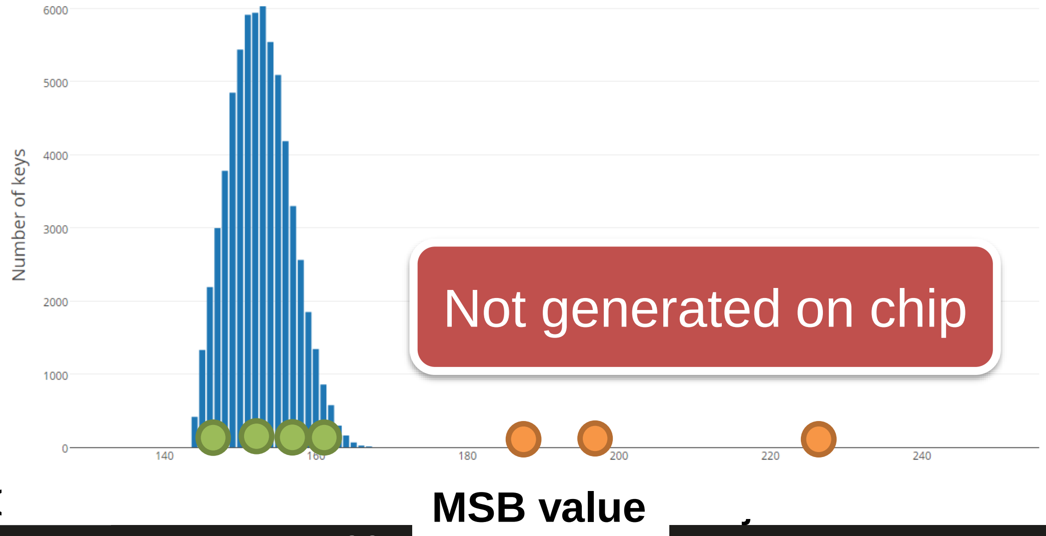


The ID-card maker has violated the most important security principle and 12,500 cards need to be replaced by people.

Hans Lõug
05/27/2018 at 13:58

share

Meeldib 32

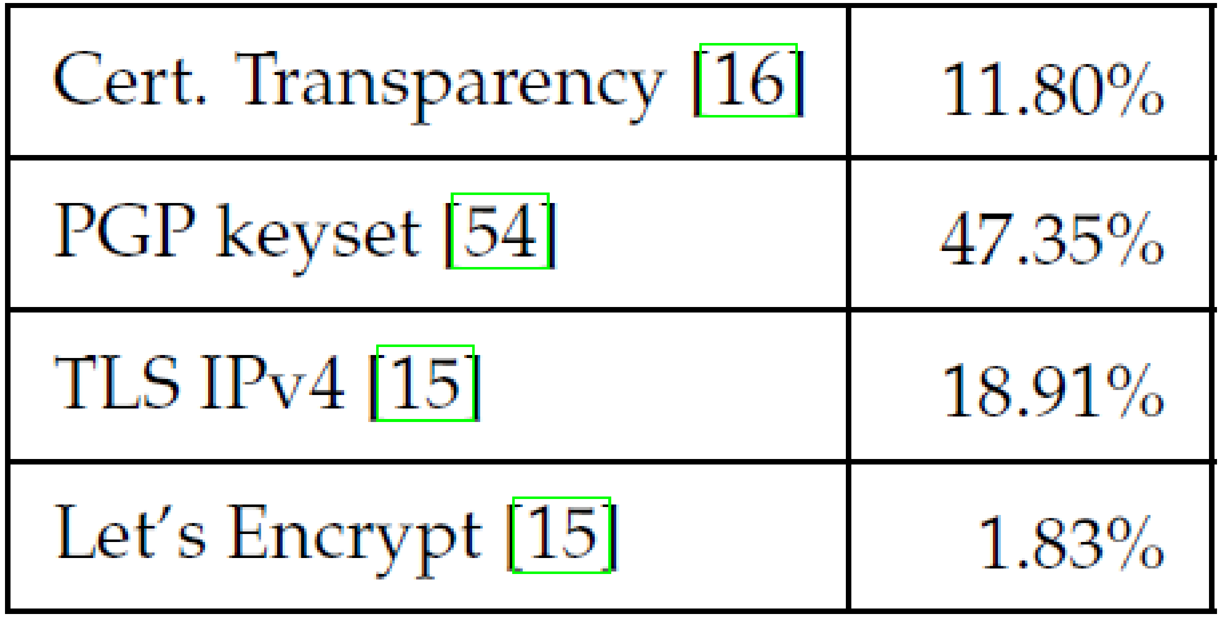


Not generated on chip

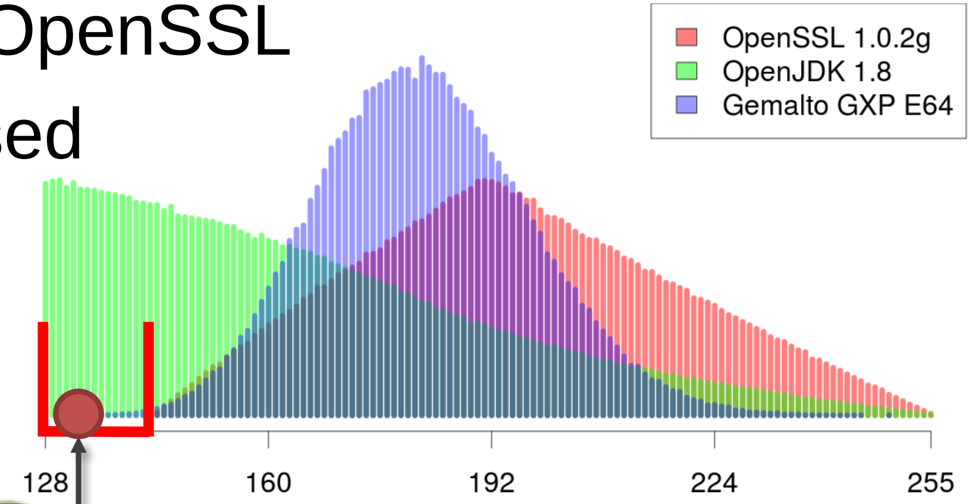
https://geenius.ee/uudis/id-kaartide-tootja-rikkus-tahtsaimat-turvapo

Sanity check: keys which *cannot* be from OpenSSL

- Keys with mask value never generated by OpenSSL
- Advantage: all keys from dataset can be used



Dataset	!OpenSSL
Cert. Transparency [16]	11.80%
PGP keyset [54]	47.35%
TLS IPv4 [15]	18.91%
Let's Encrypt [15]	1.83%



OpenSSL rare in PGP

Leaves ~81 % for OpenSSL

OpenSSL is default client

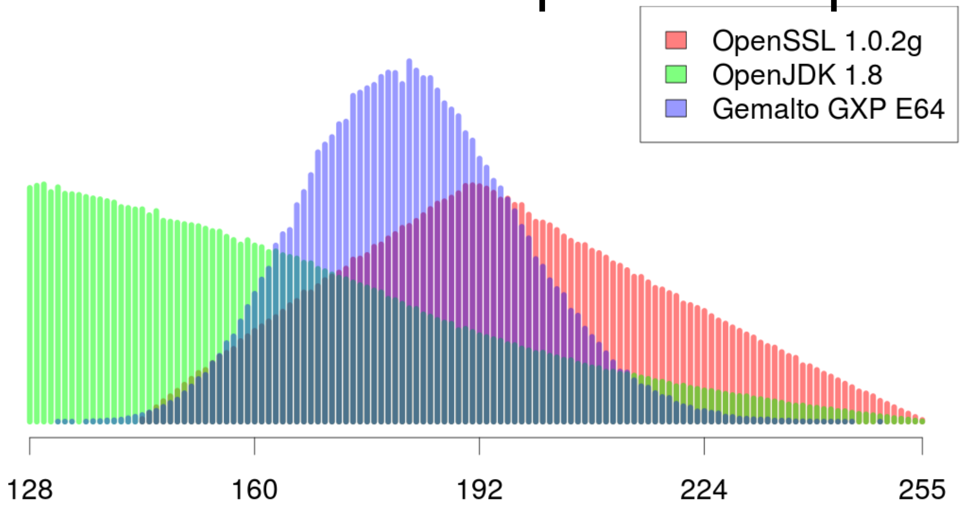
How to defend against possibility of classification?

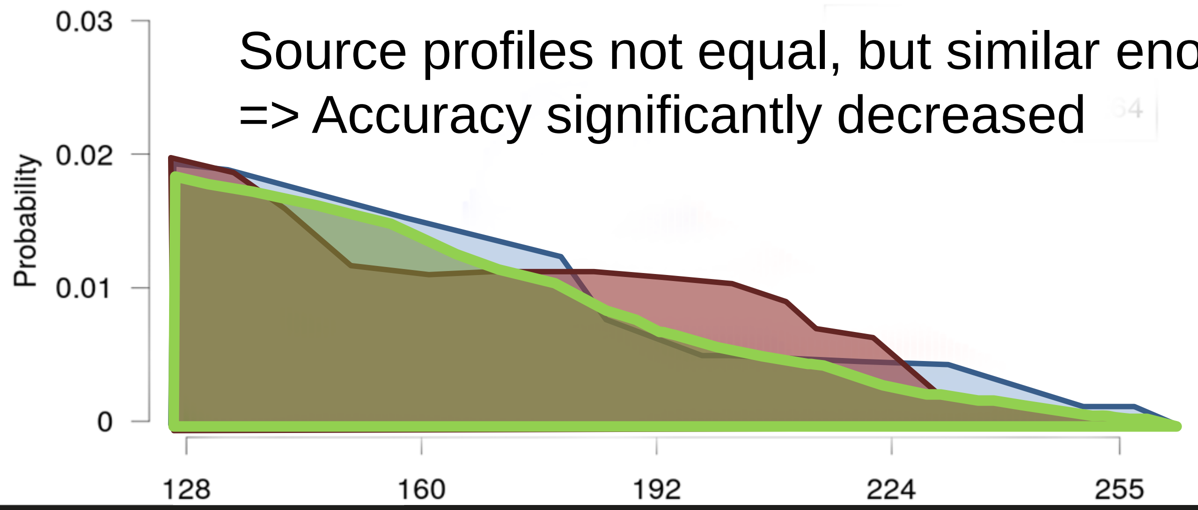
MITIGATION

How to defend against public key classification?

1. Developers of libraries

- Unify RSA key generation
  – Unlikely to happen soon, changes in critical part of code, legacy binaries…
- Plan to make minimal code changes to libs to decrease accuracy
  – Then Pull requests to upstream



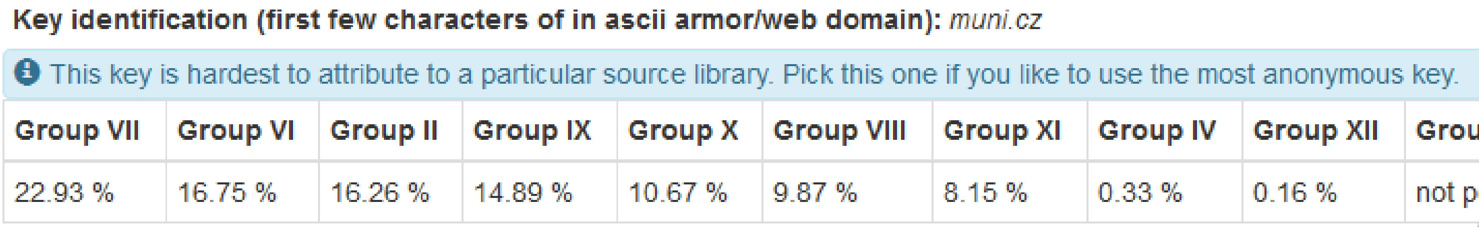


Source profiles not equal, but similar enough
=> Accuracy significantly decreased

How to defend against public key classification?

2. Users of libraries

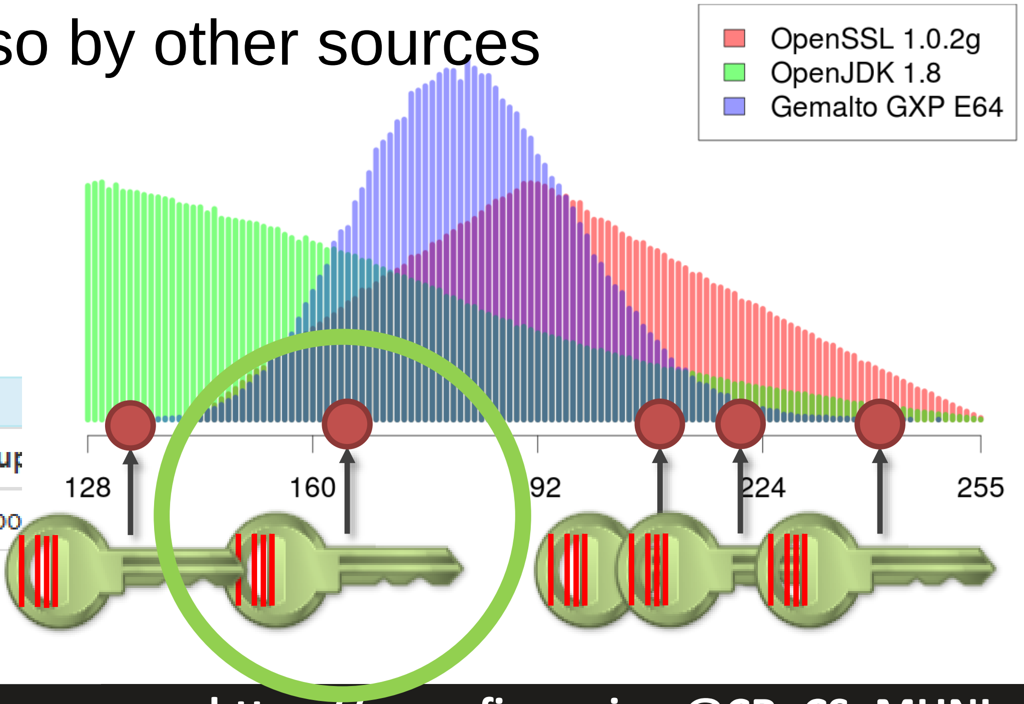
- Select one from multiple generated keys
  - – Generate multiple keys, pick least “specific” one
  - – Key with high probability to be generated also by other sources
  - – Only about 5 keys required on average



Key identification (first few characters of in ascii armor/web domain): *muni.cz*

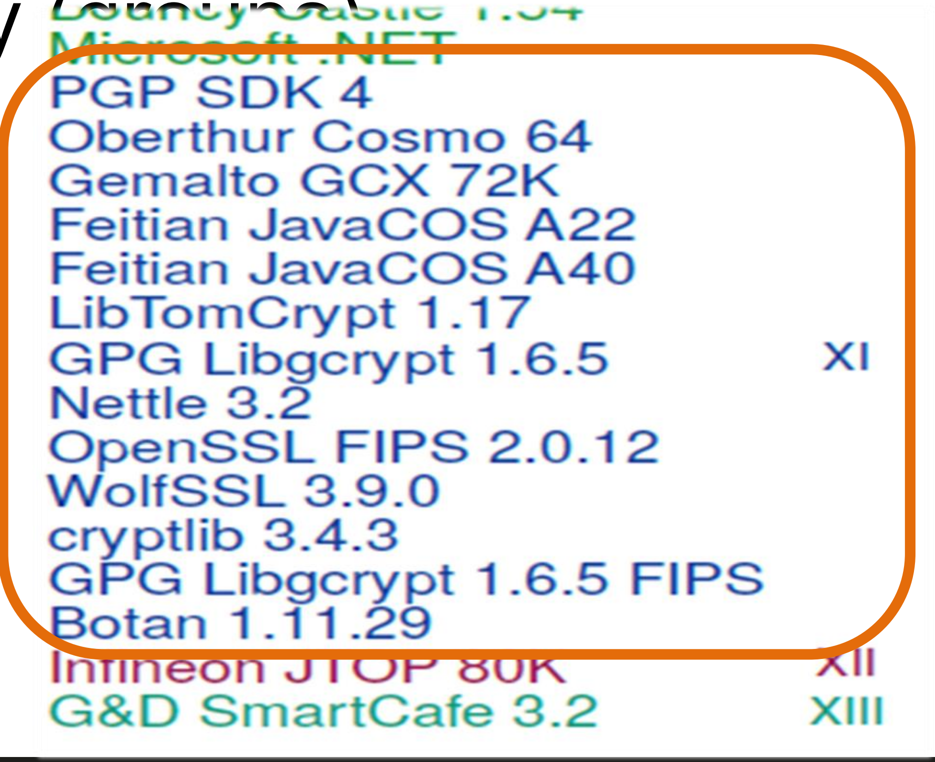
ℹ This key is hardest to attribute to a particular source library. Pick this one if you like to use the most anonymous key.

Group VII	Group VI	Group II	Group IX	Group X	Group VIII	Group XI	Group IV	Group XII	Grou
22.93 %	16.75 %	16.26 %	14.89 %	10.67 %	9.87 %	8.15 %	0.33 %	0.16 %	not po



Limitations of the current work

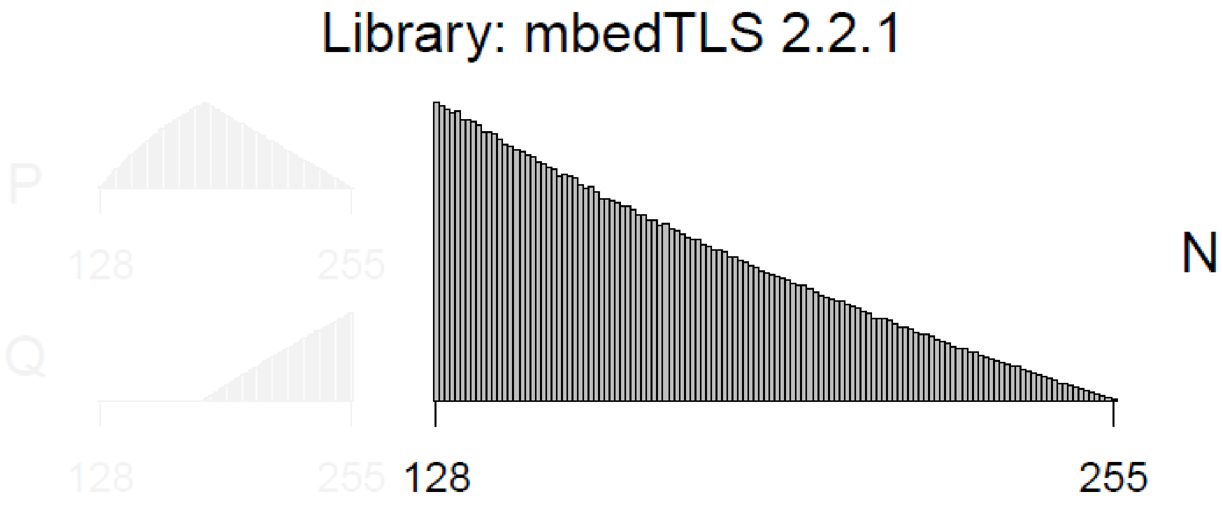
1. Lower accuracy with single key only (40% on avg.)
   – Better if prior probability is estimated
2. Can’t distinguish all libraries mutually (groups)
   – Better results if private key is available
3. Some sources missing (HSMs…)
   – Will be misclassified at the moment
   – Adding more sources, please contribute!
4. Can’t distinguish versions of libs
   – Until key generation algorithm changes

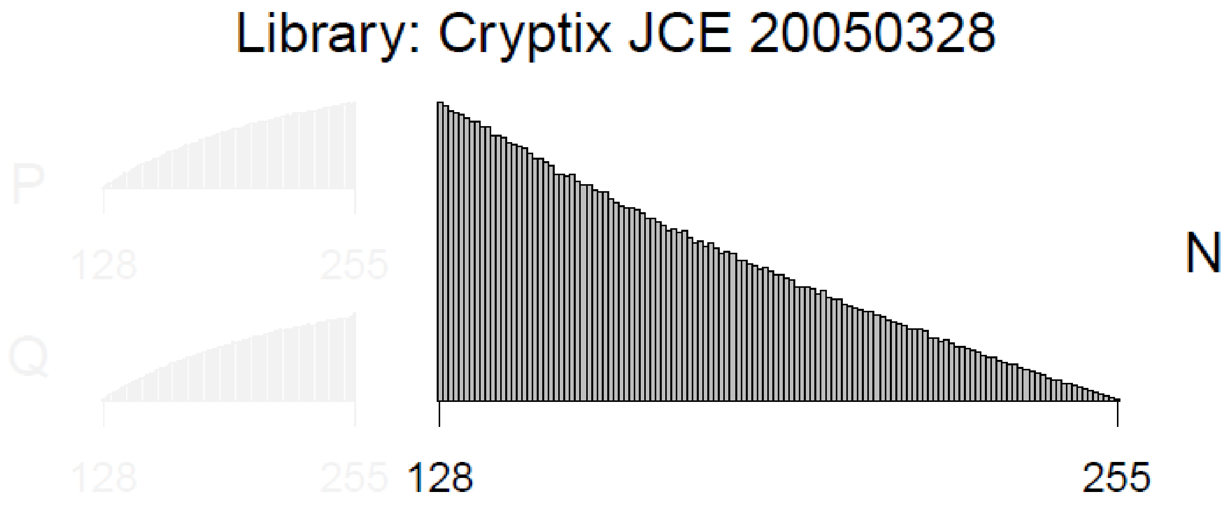


BouncyCastle 1.54	
Microsoft .NET	
PGP SDK 4	
Oberthur Cosmo 64	
Gemalto GCX 72K	
Feitian JavaCOS A22	
Feitian JavaCOS A40	
LibTomCrypt 1.17	
GPG Libgcrypt 1.6.5	XI
Nettle 3.2	
OpenSSL FIPS 2.0.12	
WolfSSL 3.9.0	
cryptlib 3.4.3	
GPG Libgcrypt 1.6.5 FIPS	
Botan 1.11.29	
Infineon JTOP 80K	XII
G&D SmartCafe 3.2	XIII

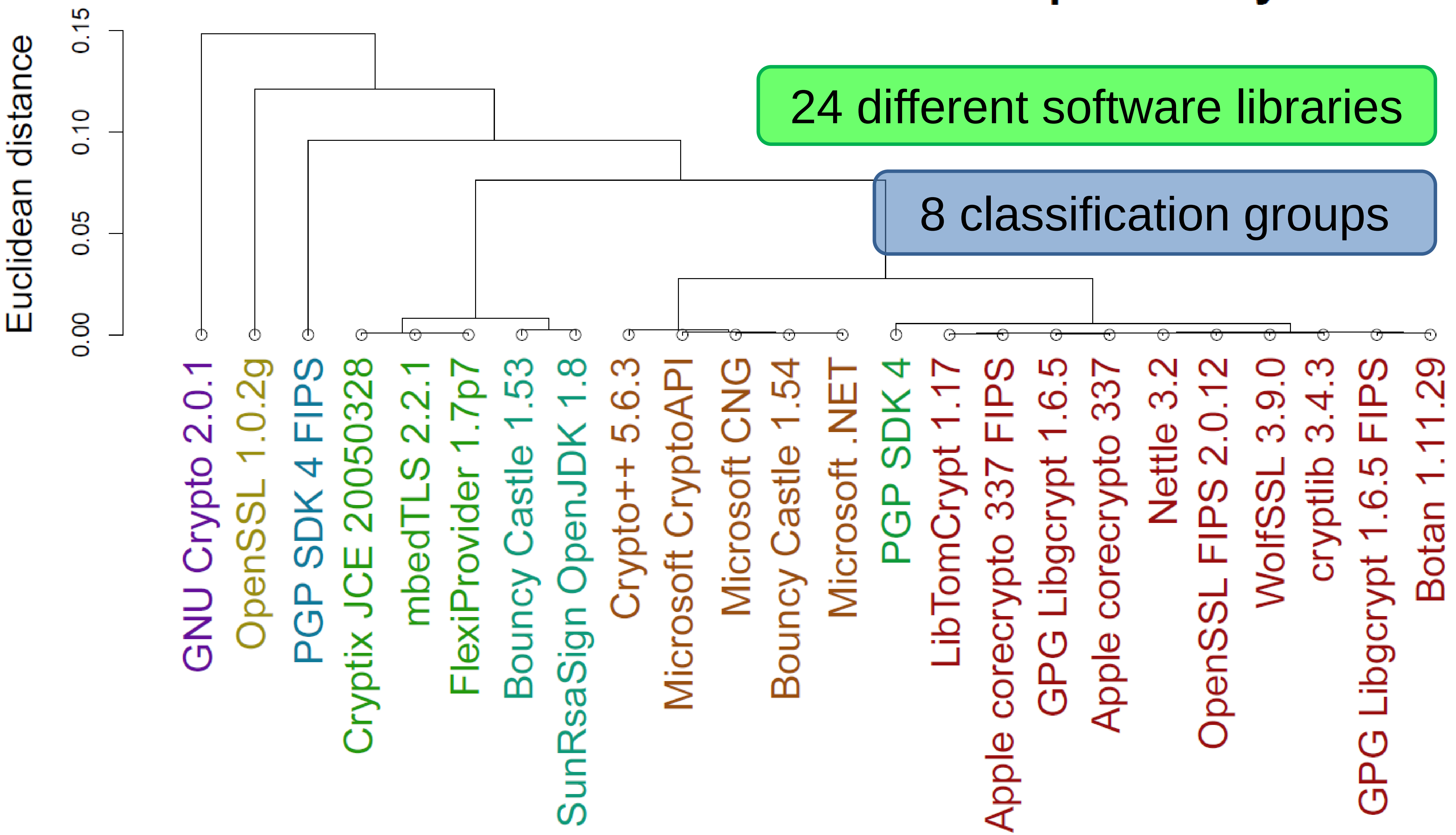
WHAT IF PRIVATE KEYS ARE AVAILABLE?

More information available in private keys


Library: mbedTLS 2.2.1


Library: Cryptix JCE 20050328

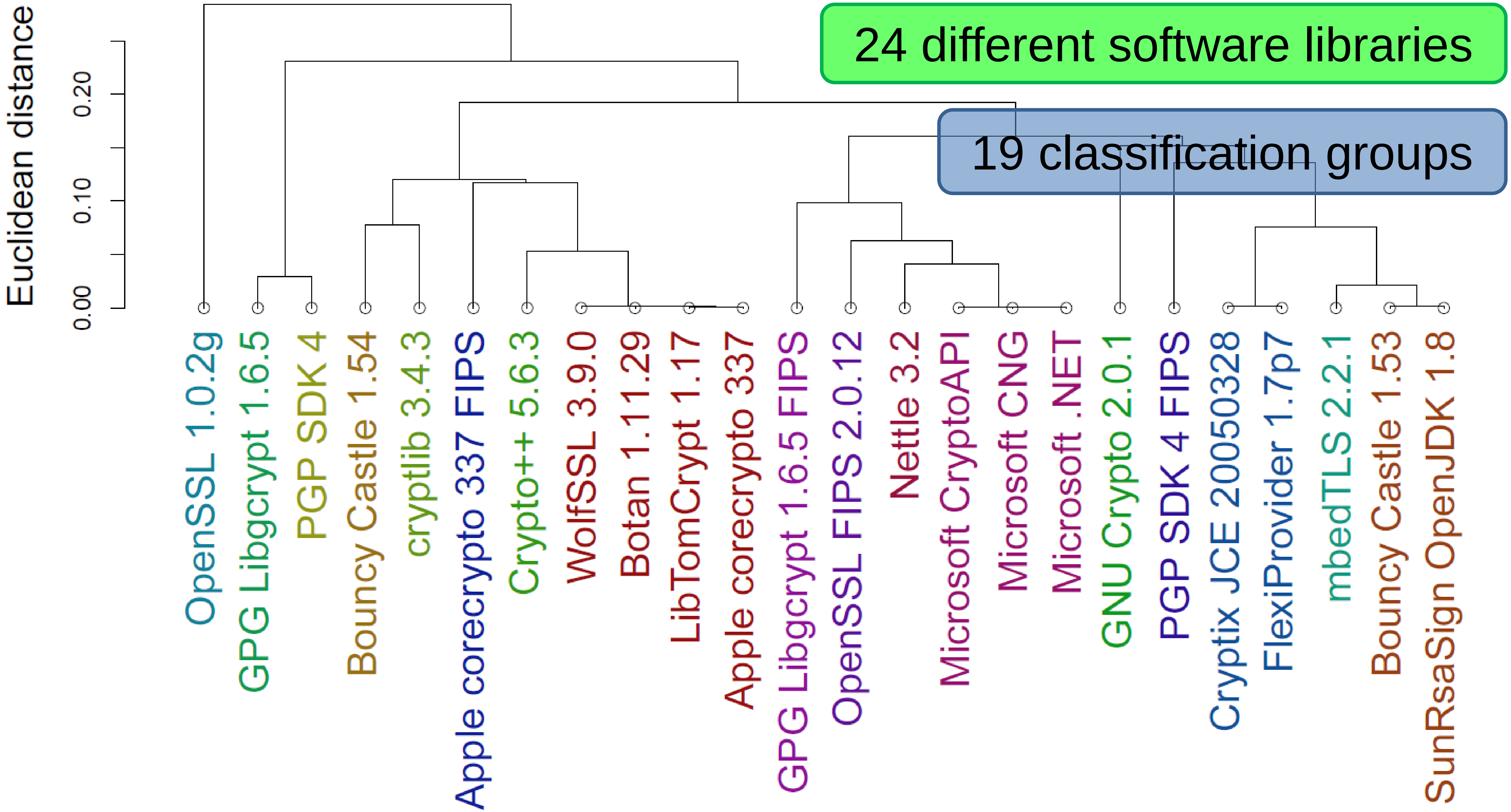
Difference in libraries based on public keys



24 different software libraries

8 classification groups

Difference in libraries based on private keys and factorization



24 different software libraries

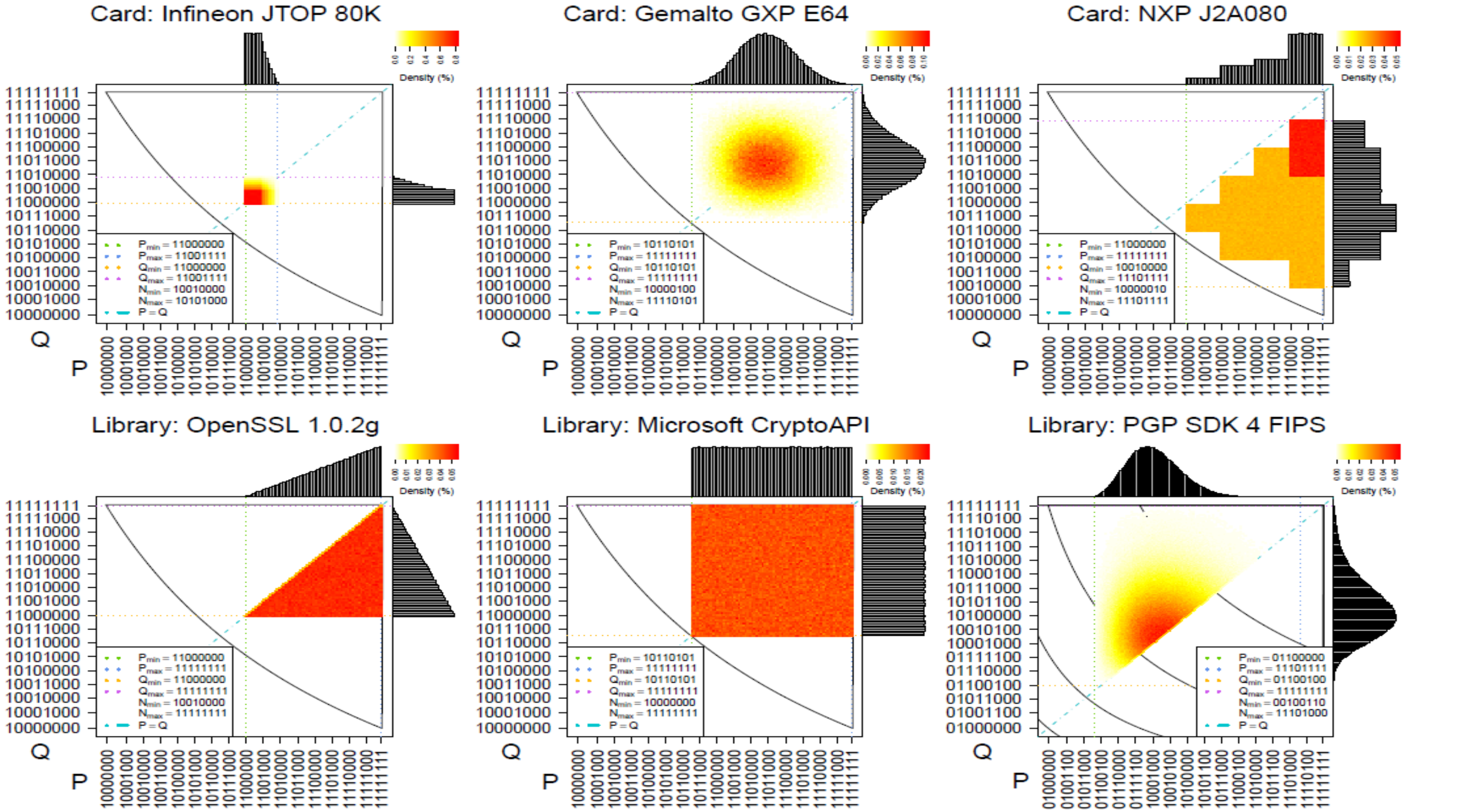
19 classification groups

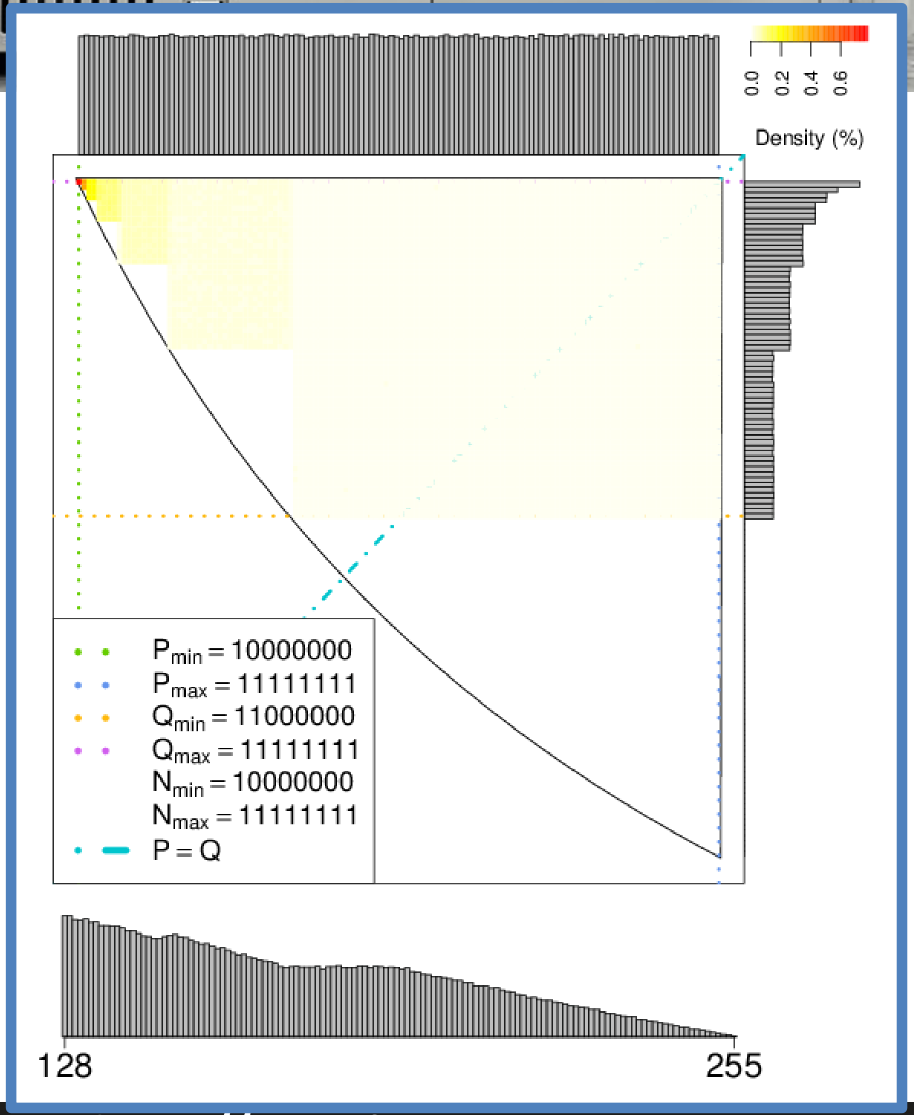
ADDING MORE SOURCES

Please contribute

- The completeness of classification database is important
- If you have access to
  - Hardware Security Modules (Thales, Safenet, IBM, Utimaco…)
  - Proprietary libraries (RSA BSafe…)
  - Software library not included yet, version with difference
  - Cryptographic smart cards
- Please contact us!

Utimaco Se50 LAN HSM





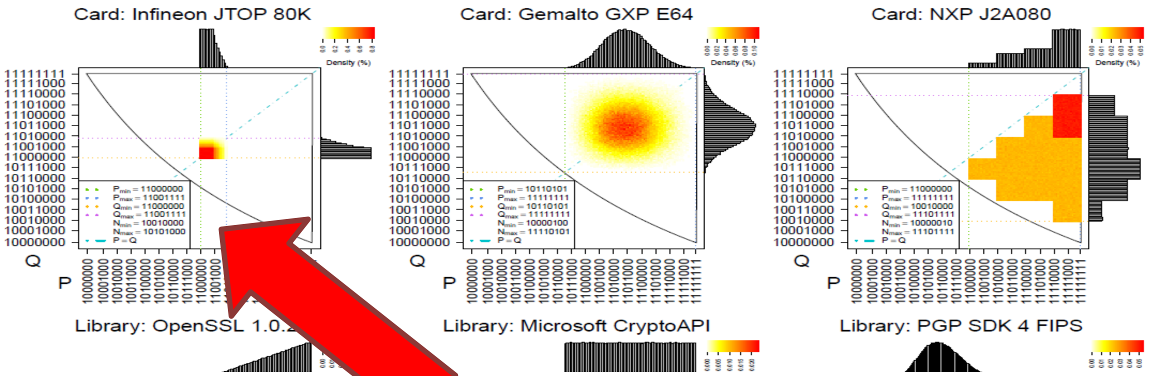
How are RSA keys generated on cryptographic smartcards

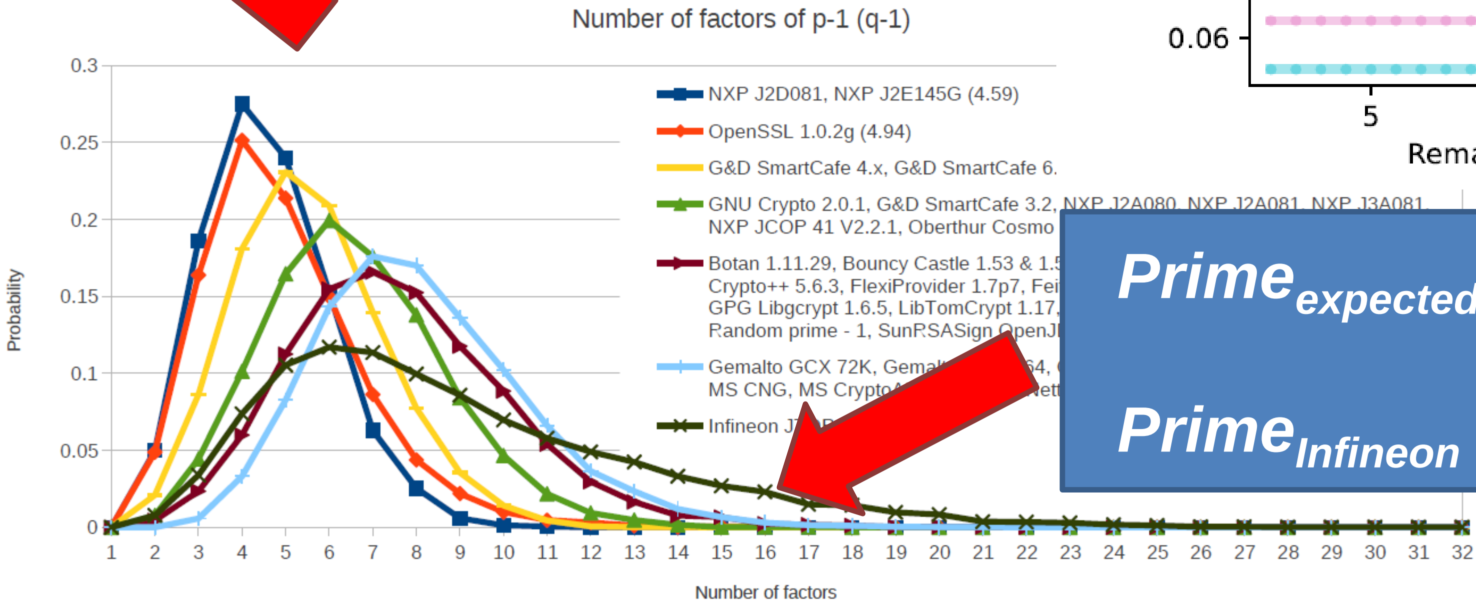
RSA ON SMARTCARDS

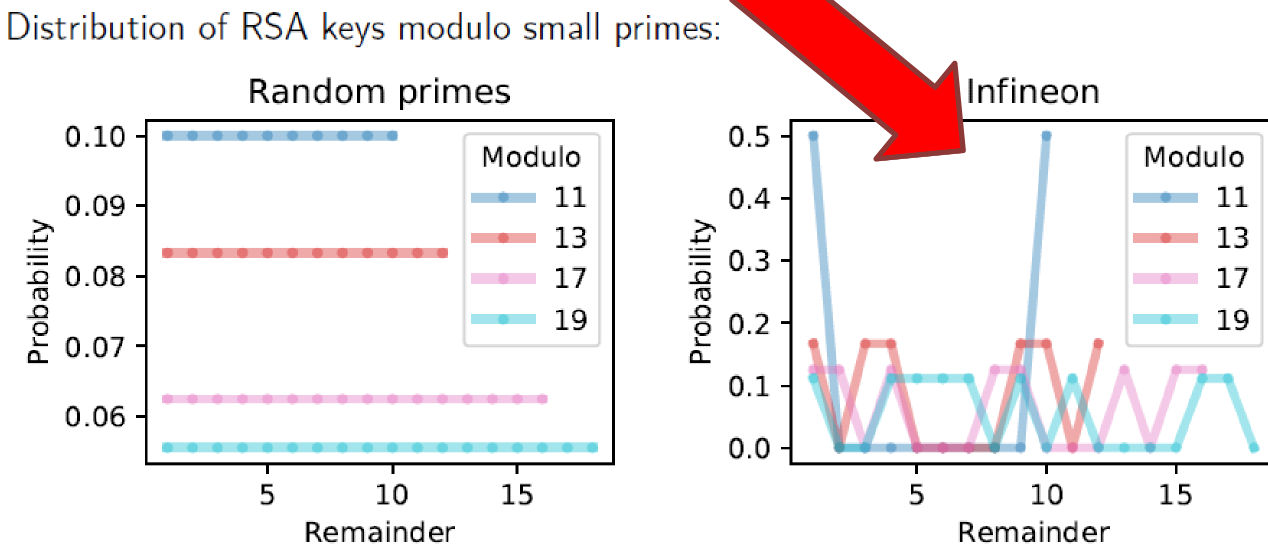
TRNG → Key: What if faulty TRNGs?

- Good source of randomness is critical
  - – TRNG can be weak or malfunctioning
- How to inspect TRNG correctness?
  - – Analysis of TRNG implementation (but is usually blackbox for smartcards)
  - – Output data can be statistically tested (100MB-1GB stream)
    - NIST STS, Dieharder, TestU01 batteries
    - Behaviour in extreme condition (+70/-50° C, radiation…)
      - – Analyse data stream gathered during extreme conditions
  - – Simple power analysis of TRNG generation
    - Is hidden/unknown operation present?

We were unaware of a far bigger issue that time



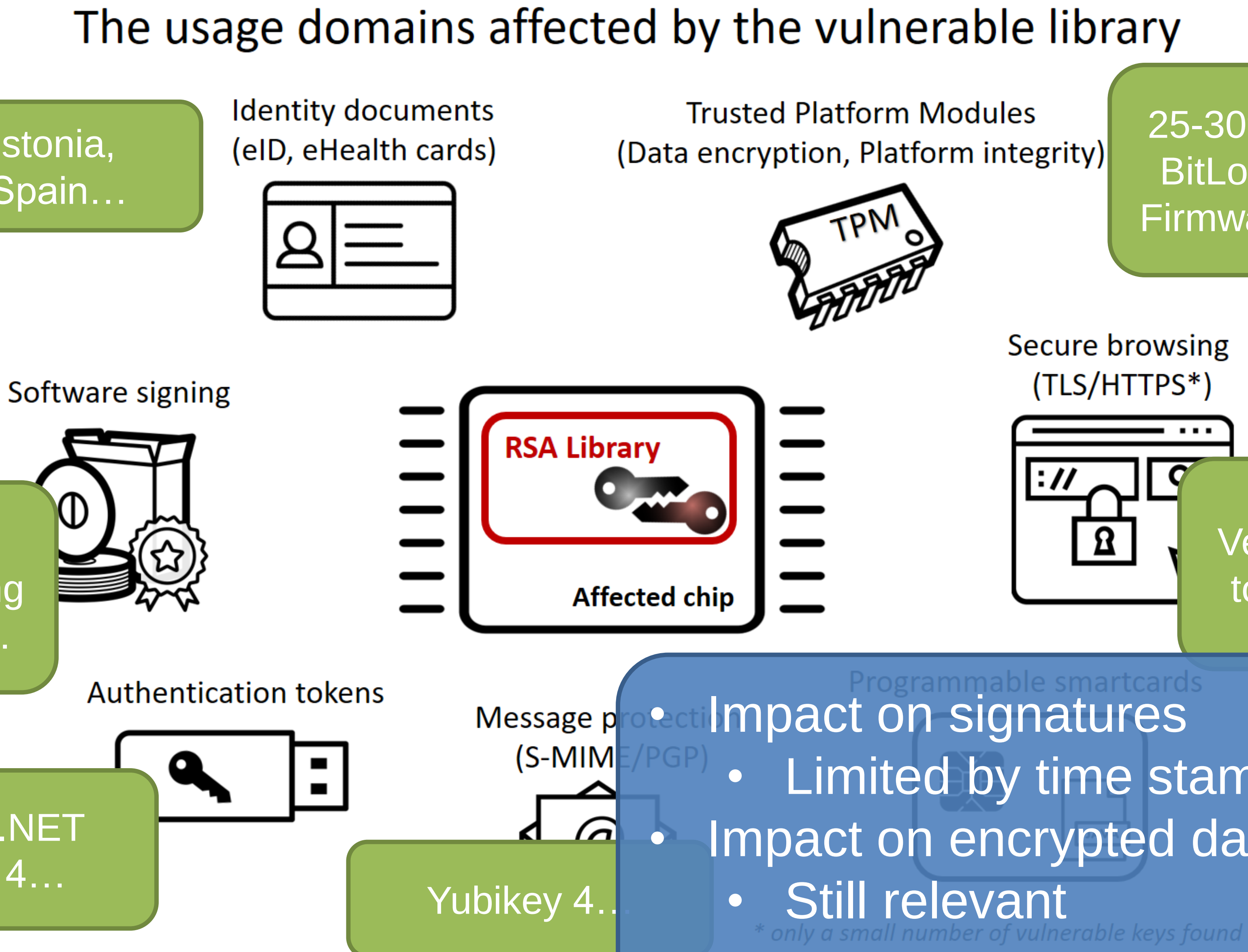




$Prime_{expected}$ = *random* ✓

$Prime_{Infineon}$ = $k * M + 65537^{a} \bmod M$

M. Nemec, M. Sys, P. Svenda, D. Klinec, V. Matyas: The Return of Coppersmith's Attack..., ACM CCS 2017


The usage domains affected by the vulnerable library
Identity documents (eID, eHealth cards)
Trusted Platform Modules (Data encryption, Platform integrity)
TPM
Software signing
RSA Library
Affected chip
Secure browsing (TLS/HTTPS*)
Authentication tokens
Message protection (S-MIME/PGP)
Programmable smartcards
* only a small number of vulnerable keys found

Austria, Estonia, Slovakia, Spain…

25-30% TPMs worldwide, BitLocker, ChromeOS… Firmware update available

Commit signing, Application signing GitHub, Maven…

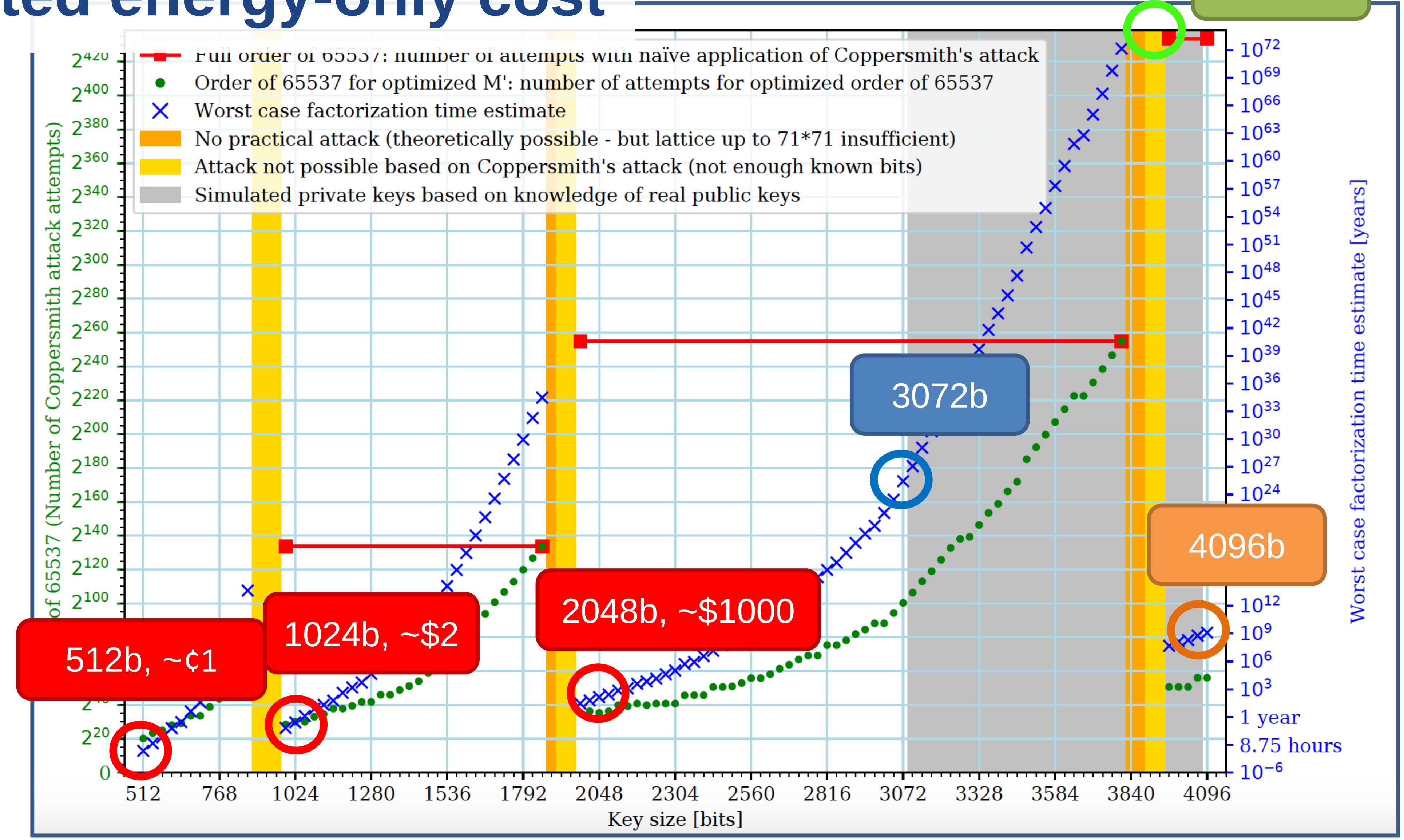
Very few keys, but all tied to SCADA management

Gemalto .NET Yubikey 4…

Yubikey 4…

- Impact on signatures
  - Limited by time stamps + revocation
- Impact on encrypted data
  - Still relevant

Estimated energy-only cost



3936b

3072b

4096b

512b, ~¢1

1024b, ~$2

2048b, ~$1000

What is the cost of an attack on RSA 2048b?

- Our paper (2017): $20,000 average price on Amazon AWS
  - Estimate: energy-only price is likely around $1000
- Lange, Bernstein (2017) – 25% faster attack (LLL chaining)
  - Found in three days and without an access to our paper!
- Estonian RIA (04/2018): “several thousand euros” energy price
- Our work (2018): algorithmic improvement, 2x faster
- Implementation speedups by graphic cards, FPGA…
  - Not (publicly) tested (typical speed-up factor 3-10x)
- Attacks only get better with time…

OSINT: Responsible disclosure & Revoked TLS certificates

- End of January 2017: Proof of Concept attack (1024b keys factorized)
- **Feb 1st: Infineon notified (email to contact at crypto group)**

2017-04-06 CA DATEV ZSM

2017-04-?? New Yubico PGP keys

- Mid May: First Infineon's customers contact us back for verification
- **Jun 20th: Incident report ID 163484, Austria eHealth certs revoked**

2017-06-30 D-Trust GmbH
2017-07-04 Deutsche Telekom AG
2017-08-10 anilyugen.com

- Sept 5th: Estonia publicly announced eID issue

2017-09-25
ChamberSign Qualified CA
D-TRUST Qualified CA

- **Oct 16th: Public disclosure (detection tool)**

2017-10-19
scada.emsglobal.net
alarms.realtimeautomation.net

- **Oct 30th: Full paper with details published (ACM CCS)**

2017-11-02 More SCADA-related certs
2017-11-03 Many *.kapsch certificates

What were impacted parties typically struggling with?

- Is this attack really practical or “just” theoretical?
- How to mitigate / update already distributed cards/tokens?
  - Estonia remote update of eIDs JavaCard application (RSA → ECC)
  - Slovakia RSA 2048b → RSA 3072b
  - Yubico: free token replacement
- Is migration to 3072b safe? (BSI says ok)
- What is actually certified? (TRNG→primes→key→use of private key)
- How to revoke large number of certificates?

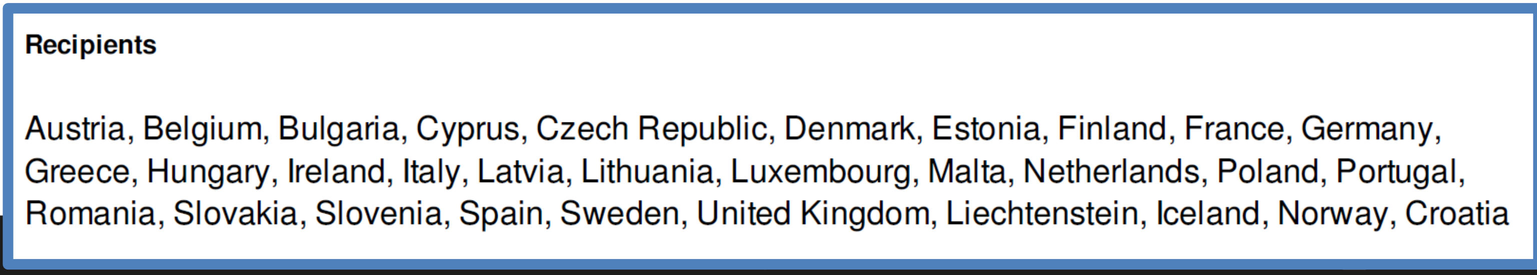
Are there any positives from ROCA vulnerability?

- Critical, long-present vulnerability mitigated
  - Vulnerable keys testing incorporated in administrators tools (Let’s Encrypt…)
- Speed-up transition to ECC or at least longer RSA keys
- Changes to standard - verifiable RSA keypair generation from seed
- Changes to certification process - more scrutiny for key generation
- Sparked discussion about more efficient information sharing (eIDAS)
- …

Another argument for more openness and certification transparency?

Responsible disclosure I.

- (NIST responsible disclosure guidelines followed)
- End of January 2017: Proof of Concept attack (1024b keys factorized)
- Feb 1st: Infineon notified (email to contact at crypto group)
- Mid May: First Infineon’s customers contact us back for verification
  - Change of some PGP keys in second half of April
- Jun 20th: Incident report ID 163484, Austria eHealth certs revoked
  - Countries around Europe should have been notified
  - BUT: unspecific third party failure, concrete vendor named (but not Infineon)

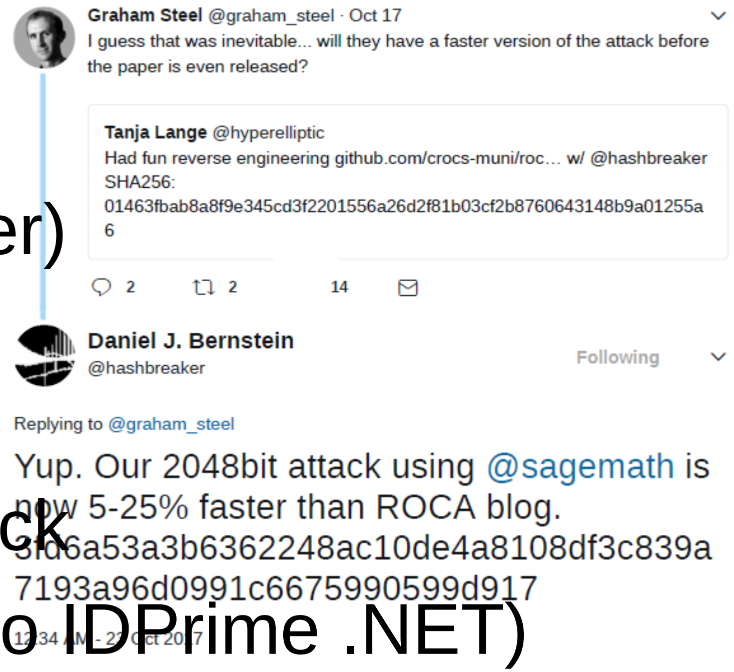


Recipients

Austria, Belgium, Bulgaria, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, United Kingdom, Liechtenstein, Iceland, Norway, Croatia

Responsible disclosure II.

- Last week Aug: vulnerable new EE certs detected (LDAP scan)
- Aug 30th: EE CERT formally contacted by us
- Sept 5th: Estonia publicly announced eID issue
- Oct 10th: Microsoft Patch Tuesday (TPMs, Bitlocker)
- Oct 16th: Public disclosure (coincide with KRACK)
  - Impact announced by us, detection tool released
- Oct 23rd: Lange& Bernstein announced faster attack
- Vulnerable devices from year 2007 found (Gemalto IDPrime .NET)
- Oct 30th: Full paper with details published (ACM CCS)
- 2/3.11. Slovakia/Estonia revokes 300k/760k certificates (60M in Spain)



Graham Steel @graham_steel · Oct 17

I guess that was inevitable... will they have a faster version of the attack before the paper is even released?

> **Tanja Lange** @hyperelliptic
> Had fun reverse engineering github.com/crocs-muni/roc… w/ @hashbreaker
> SHA256:
> 01463fbab8a8f9e345cd3f2201556a26d2f81b03cf2b8760643148b9a01255a
> 6

Daniel J. Bernstein
@hashbreaker

Replying to @graham_steel

Yup. Our 2048bit attack using @sagemath is now 5-25% faster than ROCA blog. 3fd6a53a3b6362248ac10de4a8108df3c839a
7193a96d0991c6675990599d917

12:34 AM - 23 Oct 2017

This particular flaw on Infineon side, but wider problem

- Why such a strong secrecy around the whole smartcard industry?
  - I cannot buy newer cards (lucky if ICFabDate > 2020)
  - Security best practices checklist for JC development available only after NDA…
  - I cannot use more secure version of crypto functions (not exposed via public API)
  - Research prototypes using ECPoint cannot be published (NDA)
- Smartcards not secure enough if more complete information published?
  - (not calling for completely open-source hardware, but more openness beneficial)
- Certification process does not seem to cover all steps of keygen
  - TRNG (input) and use of private key (side-channels, faults) covered
  - How primes are created from TRNG omitted
- Certification process seems to “reward” secrecy to some extent
  - No developer samples, no public detailed specs…

ANALYZING SECURITY CERTIFICATIONS…

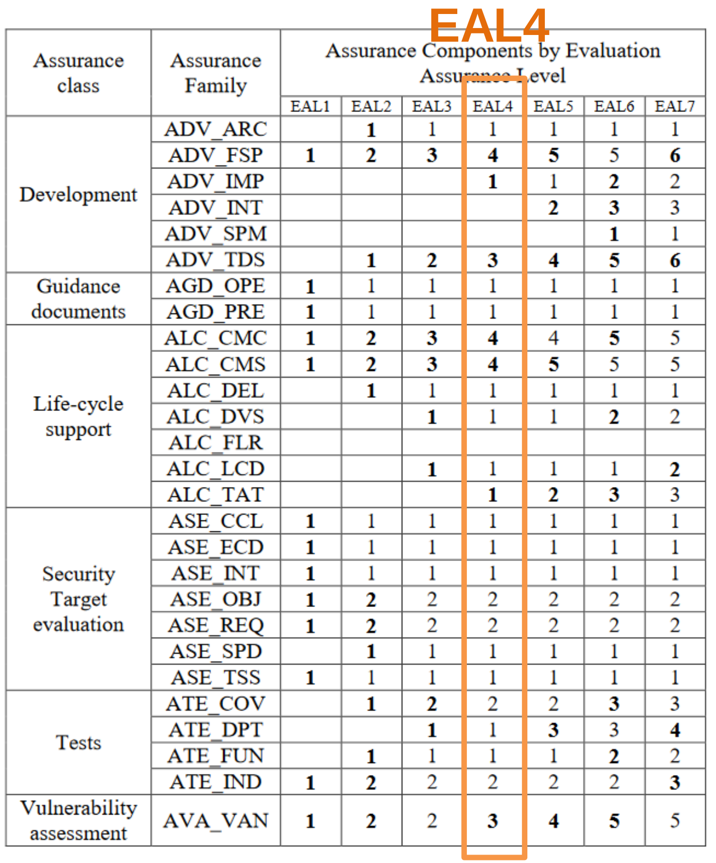
Cyber Security for Europe

Key points

1. The current state of security certification is unsatisfactory
2. More utility and transparency can be obtained already within the current system (=> seccerts project)
3. Data-based analysis can identify beneficial aspects of certification
4. Less trust in third parties, more openness, more end-user replicability (make community-provided analysis easier (aka replicable CI with deterministic builds)

Common Criteria certification reminder

- Evaluation Assurance Level (EAL) corresponds to extent of scrutiny
  - EAL1-7, augmented - particular EAL also mandates minimal SAR levels
  - Certificates mutually recognized up to EAL 2, up to EAL 4 inside EU
    - Common Criteria Recognition Arrangement (CCRA)
- Claims validated by accredited laboratories/evaluation facilities
  - If successful, product certificate is given and published
    - by Certificate Authorizing Members (e.g., French ANSSI, German BSI)
    - validity period typically 3 or 6 years
  - Maintenance Report(s) – smaller changes which doesn't require full recertification, or just continuation
    - submitted by vendor, again validated by lab
  - Labs comply with ISO/IEC 17025, national cert. bodies approved against ISO/IEC 17065



EAL4

Assurance class	Assurance Family	Assurance Components by Evaluation Assurance Level						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Development	ADV_ARC		**1**	1	1	1	1	1
	ADV_FSP	**1**	**2**	**3**	**4**	**5**	5	**6**
	ADV_IMP				**1**	1	**2**	2
	ADV_INT					**2**	**3**	3
	ADV_SPM						**1**	1
	ADV_TDS		**1**	**2**	**3**	**4**	**5**	**6**
Guidance documents	AGD_OPE	**1**	1	1	1	1	1	1
	AGD_PRE	**1**	1	1	1	1	1	1
Life-cycle support	ALC_CMC	**1**	**2**	**3**	**4**	4	**5**	5
	ALC_CMS	**1**	**2**	**3**	**4**	**5**	5	5
	ALC_DEL		**1**	1	1	1	1	1
	ALC_DVS			**1**	1	1	**2**	2
	ALC_FLR							
	ALC_LCD			**1**	1	1	1	**2**
	ALC_TAT				**1**	**2**	**3**	3
Security Target evaluation	ASE_CCL	**1**	1	1	1	1	1	1
	ASE_ECD	**1**	1	1	1	1	1	1
	ASE_INT	**1**	1	1	1	1	1	1
	ASE_OBJ	**1**	**2**	2	2	2	2	2
	ASE_REQ	**1**	**2**	2	2	2	2	2
	ASE_SPD		**1**	1	1	1	1	1
	ASE_TSS	**1**	1	1	1	1	1	1
Tests	ATE_COV		**1**	**2**	2	2	**3**	3
	ATE_DPT			**1**	1	**3**	3	**4**
	ATE_FUN		**1**	1	1	1	**2**	2
	ATE_IND	**1**	**2**	2	2	2	2	**3**
Vulnerability assessment	AVA_VAN	**1**	**2**	2	**3**	**4**	**5**	5

Cyber Security for Europe

Documents produced and publicly available

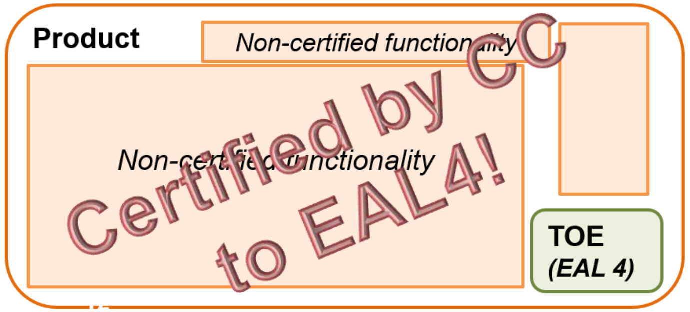
- Documents produced and/or publicly available
  - Security Target document – provided by vendor (or on behalf) to Evaluation facility
  - Certification Report – issued by Cert. Auth. Member (e.g., French ANSSI), after checks by accredited Evaluation facility/lab (e.g., Serma Technologies)
  - Maintenance Report(s) – smaller changes that don’t require full recertification
  - Protection Profiles documents – template for specific functionality, single vendor or collaborative
  - CSV/HTML pages with some additional metadata, summary documents
    - automatically generated by CC portal, Cert. Auth. Members…
- *(Additional confidential documents shared between vendor and lab)*

NIST FIPS 140-2 certification primer

- Security Requirements for Cryptographic Modules
  - More specific domain than Common Criteria - both hardware and software
- Module – evaluated item with some security/cryptographic functionality
  - Certificate #3820
- Algorithm - implementation of security algorithm by given module
  - List of approved algorithms
    - e.g., AES in GCM mode, RSA key wrapping, SHA2 hash function...
  - Other algorithms possibly available in non-FIPS mode
- Public documents: Security Policy document, certificate web page

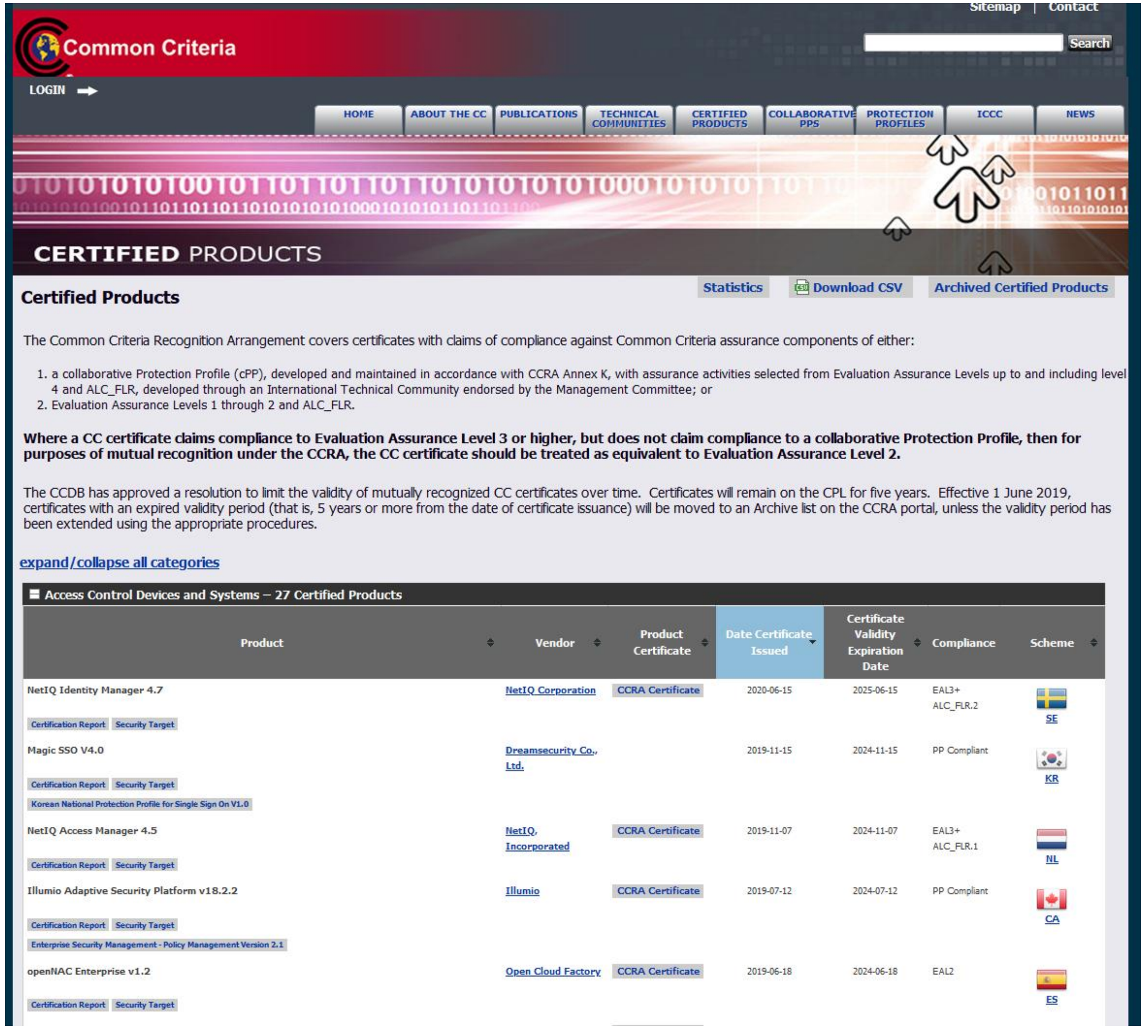
Some problems…

- CC certification is costly and takes long time (>$100k, >3 months)
  - Works well for static, long-time usable products (hardware, smartcards…)
  - CC generally not suitable for quickly changing products (software in cloud with daily updates…)
- Hard to interpret actual security by end-users
  - Evaluation only with respect to ToE (crucial parts can be put out-of-scope by vendor)
    - Marketing claims like “Common Criteria certified” (important is ToE details, achieved EAL, PP conformance, laboratory used…) or “Common Criteria ready”
  - Product is changing (sw/hw updates) – what is actually certified?
- How well was product scrutinized by testing laboratory?
  - Lack of public details, tools used, configurations and results…
  - Exact procedures under NDA and IP of labs/vendors



https://seccerts.org/

Common Criteria: https://www.commoncriteriaportal.org/
FIPS140-2: https://csrc.nist.gov/projects/cryptographic-module-validation-program/



Certified Products

The Common Criteria Recognition Arrangement covers certificates with claims of compliance against Common Criteria assurance components of either:

1. a collaborative Protection Profile (cPP), developed and maintained in accordance with CCRA Annex K, with assurance activities selected from Evaluation Assurance Levels up to and including level 4 and ALC_FLR, developed through an International Technical Community endorsed by the Management Committee; or
2. Evaluation Assurance Levels 1 through 2 and ALC_FLR.

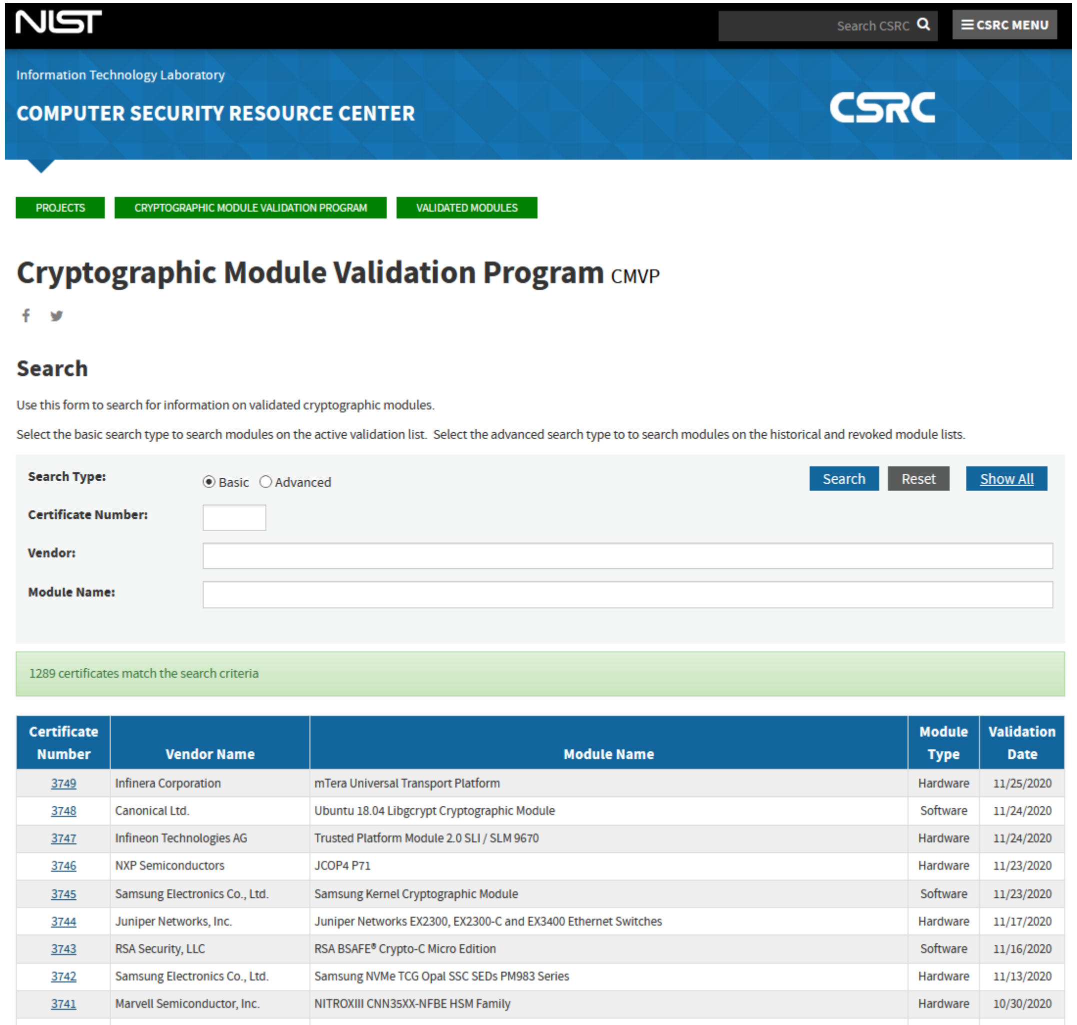
Where a CC certificate claims compliance to Evaluation Assurance Level 3 or higher, but does not claim compliance to a collaborative Protection Profile, then for purposes of mutual recognition under the CCRA, the CC certificate should be treated as equivalent to Evaluation Assurance Level 2.

The CCDB has approved a resolution to limit the validity of mutually recognized CC certificates over time. Certificates will remain on the CPL for five years. Effective 1 June 2019, certificates with an expired validity period (that is, 5 years or more from the date of certificate issuance) will be moved to an Archive list on the CCRA portal, unless the validity period has been extended using the appropriate procedures.

expand/collapse all categories

Access Control Devices and Systems – 27 Certified Products

Product	Vendor	Product Certificate	Date Certificate Issued	Certificate Validity Expiration Date	Compliance	Scheme
NetIQ Identity Manager 4.7 Certification Report Security Target	NetIQ Corporation	CCRA Certificate	2020-06-15	2025-06-15	EAL3+ ALC_FLR.2	SE
Magic SSO V4.0 Certification Report Security Target Korean National Protection Profile for Single Sign On V1.0	Dreamsecurity Co., Ltd.		2019-11-15	2024-11-15	PP Compliant	KR
NetIQ Access Manager 4.5 Certification Report Security Target	NetIQ, Incorporated	CCRA Certificate	2019-11-07	2024-11-07	EAL3+ ALC_FLR.1	NL
Illumio Adaptive Security Platform v18.2.2 Certification Report Security Target Enterprise Security Management - Policy Management Version 2.1	Illumio	CCRA Certificate	2019-07-12	2024-07-12	PP Compliant	CA
openNAC Enterprise v1.2 Certification Report Security Target	Open Cloud Factory	CCRA Certificate	2019-06-18	2024-06-18	EAL2	ES


NIST
Search CSRC
CSRC MENU
Information Technology Laboratory
COMPUTER SECURITY RESOURCE CENTER
CSRC
PROJECTS
CRYPTOGRAPHIC MODULE VALIDATION PROGRAM
VALIDATED MODULES

Cryptographic Module Validation Program CMVP

Search

Use this form to search for information on validated cryptographic modules.

Select the basic search type to search modules on the active validation list. Select the advanced search type to to search modules on the historical and revoked module lists.

Search Type: ● Basic ○ Advanced

Certificate Number:

Vendor:

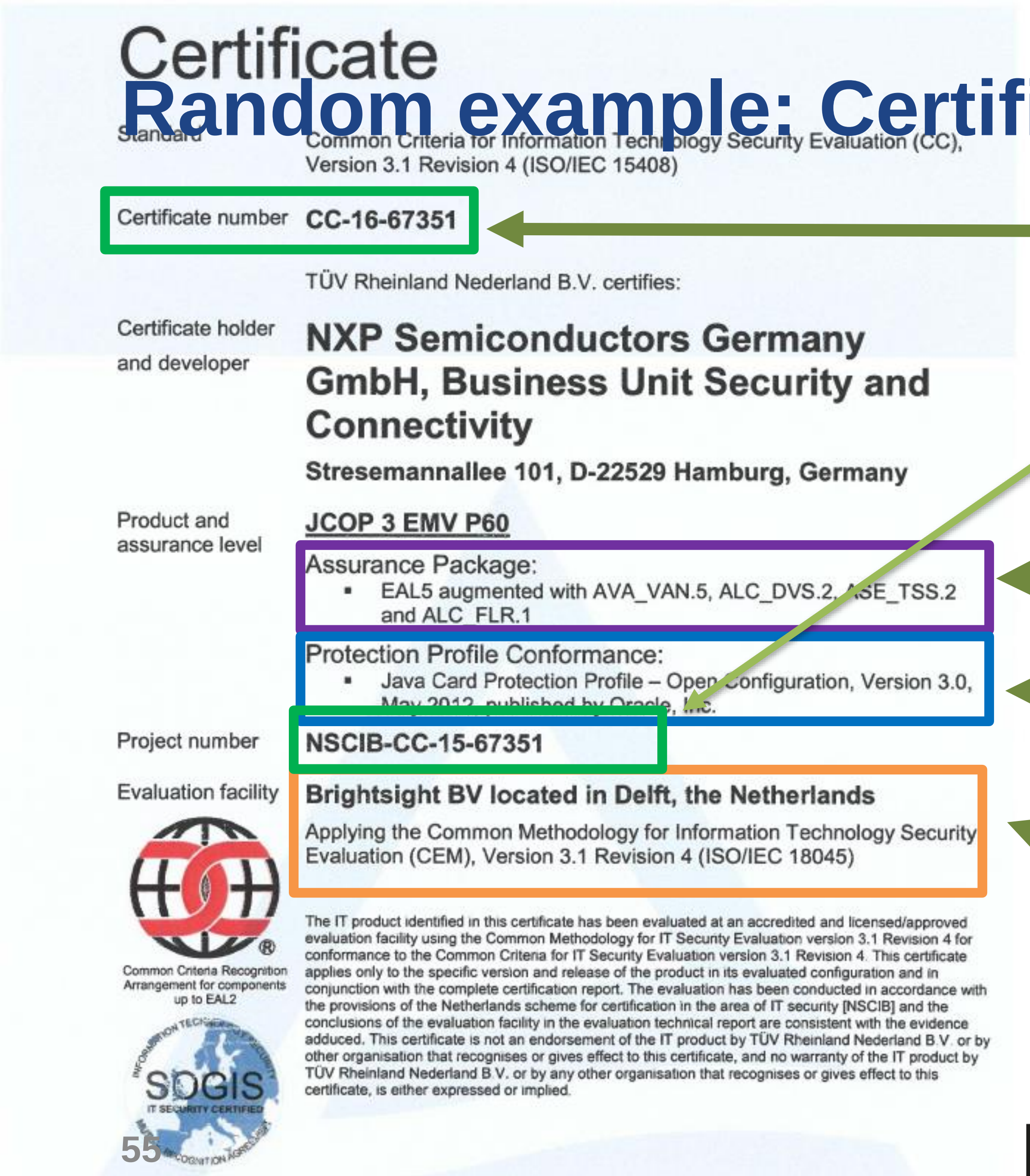
Module Name:

Search Reset Show All

1289 certificates match the search criteria

Certificate Number	Vendor Name	Module Name	Module Type	Validation Date
3749	Infinera Corporation	mTera Universal Transport Platform	Hardware	11/25/2020
3748	Canonical Ltd.	Ubuntu 18.04 Libgcrypt Cryptographic Module	Software	11/24/2020
3747	Infineon Technologies AG	Trusted Platform Module 2.0 SLI / SLM 9670	Hardware	11/24/2020
3746	NXP Semiconductors	JCOP4 P71	Hardware	11/23/2020
3745	Samsung Electronics Co., Ltd.	Samsung Kernel Cryptographic Module	Software	11/23/2020
3744	Juniper Networks, Inc.	Juniper Networks EX2300, EX2300-C and EX3400 Ethernet Switches	Hardware	11/17/2020
3743	RSA Security, LLC	RSA BSAFE® Crypto-C Micro Edition	Software	11/16/2020
3742	Samsung Electronics Co., Ltd.	Samsung NVMe TCG Opal SSC SEDs PM983 Series	Hardware	11/13/2020
3741	Marvell Semiconductor, Inc.	NITROXIII CNN35XX-NFBE HSM Family	Hardware	10/30/2020

Random example: Certificate doc



Certificate

Standard Common Criteria for Information Technology Security Evaluation (CC), Version 3.1 Revision 4 (ISO/IEC 15408)

Certificate number **CC-16-67351**

TÜV Rheinland Nederland B.V. certifies:

Certificate holder and developer **NXP Semiconductors Germany GmbH, Business Unit Security and Connectivity**

Stresemannallee 101, D-22529 Hamburg, Germany

Product and assurance level **JCOP 3 EMV P60**

Assurance Package:

- EAL5 augmented with AVA_VAN.5, ALC_DVS.2, ASE_TSS.2 and ALC_FLR.1

Protection Profile Conformance:

- Java Card Protection Profile – Open Configuration, Version 3.0, May 2012, published by Oracle, Inc.

Project number **NSCIB-CC-15-67351**

Evaluation facility **Brightsight BV located in Delft, the Netherlands**

Applying the Common Methodology for Information Technology Security Evaluation (CEM), Version 3.1 Revision 4 (ISO/IEC 18045)

Common Criteria Recognition Arrangement for components up to EAL2

SOGIS IT SECURITY CERTIFIED

The IT product identified in this certificate has been evaluated at an accredited and licensed/approved evaluation facility using the Common Methodology for IT Security Evaluation version 3.1 Revision 4 for conformance to the Common Criteria for IT Security Evaluation version 3.1 Revision 4. This certificate applies only to the specific version and release of the product in its evaluated configuration and in conjunction with the complete certification report. The evaluation has been conducted in accordance with the provisions of the Netherlands scheme for certification in the area of IT security [NSCIB] and the conclusions of the evaluation facility in the evaluation technical report are consistent with the evidence adduced. This certificate is not an endorsement of the IT product by TÜV Rheinland Nederland B.V. or by other organisation that recognises or gives effect to this certificate, and no warranty of the IT product by TÜV Rheinland Nederland B.V. or by any other organisation that recognises or gives effect to this certificate, is either expressed or implied.

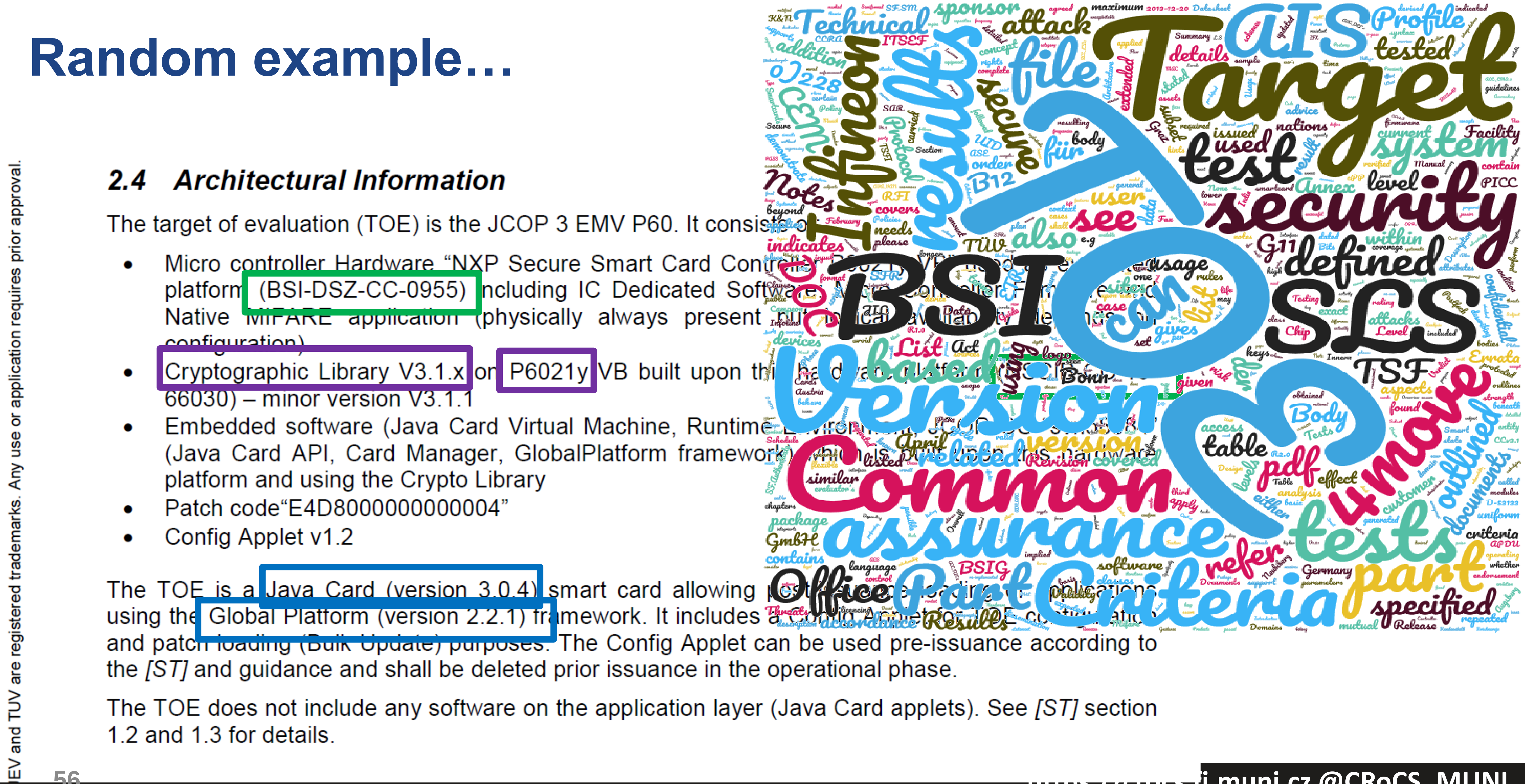
Certification number/ID
(specific to certificate producer/country)

Achieved Evaluation Assurance Level (EAL)
EAL5 + 4 additional SARs

Conforming to Java Card Protection Profile, v3.0 from May 2012

Testing laboratory / evaluation facility

Random example…



EV and TUV are registered trademarks. Any use or application requires prior approval.

2.4 Architectural Information

The target of evaluation (TOE) is the JCOP 3 EMV P60. It consis

- Micro controller Hardware “NXP Secure Smart Card Contr platform (BSI-DSZ-CC-0955) including IC Dedicated Softw Native MIFARE application (physically always present configuration)
- Cryptographic Library V3.1.x on P6021y VB built upon thi 66030) – minor version V3.1.1
- Embedded software (Java Card Virtual Machine, Runtime (Java Card API, Card Manager, GlobalPlatform framework platform and using the Crypto Library
- Patch code“E4D8000000000004”
- Config Applet v1.2

The TOE is a Java Card (version 3.0.4) smart card allowing using the Global Platform (version 2.2.1) framework. It includes a and patch loading (Bulk Update) purposes. The Config Applet can be used pre-issuance according to the [ST] and guidance and shall be deleted prior issuance in the operational phase.

The TOE does not include any software on the application layer (Java Card applets). See [ST] section 1.2 and 1.3 for details.

WHAT IF YOU HAVE CRYSTAL BALL REGARDING THE CERTIFIED PRODUCTS?

Mental exercise – What I need to do to (re)verify security of purchased certified product?

- What was certified (ToE)?
- How were claims tested?
- What tools were used, what configuration, what were thresholds, results obtained?
- How is product security monitored after certification?

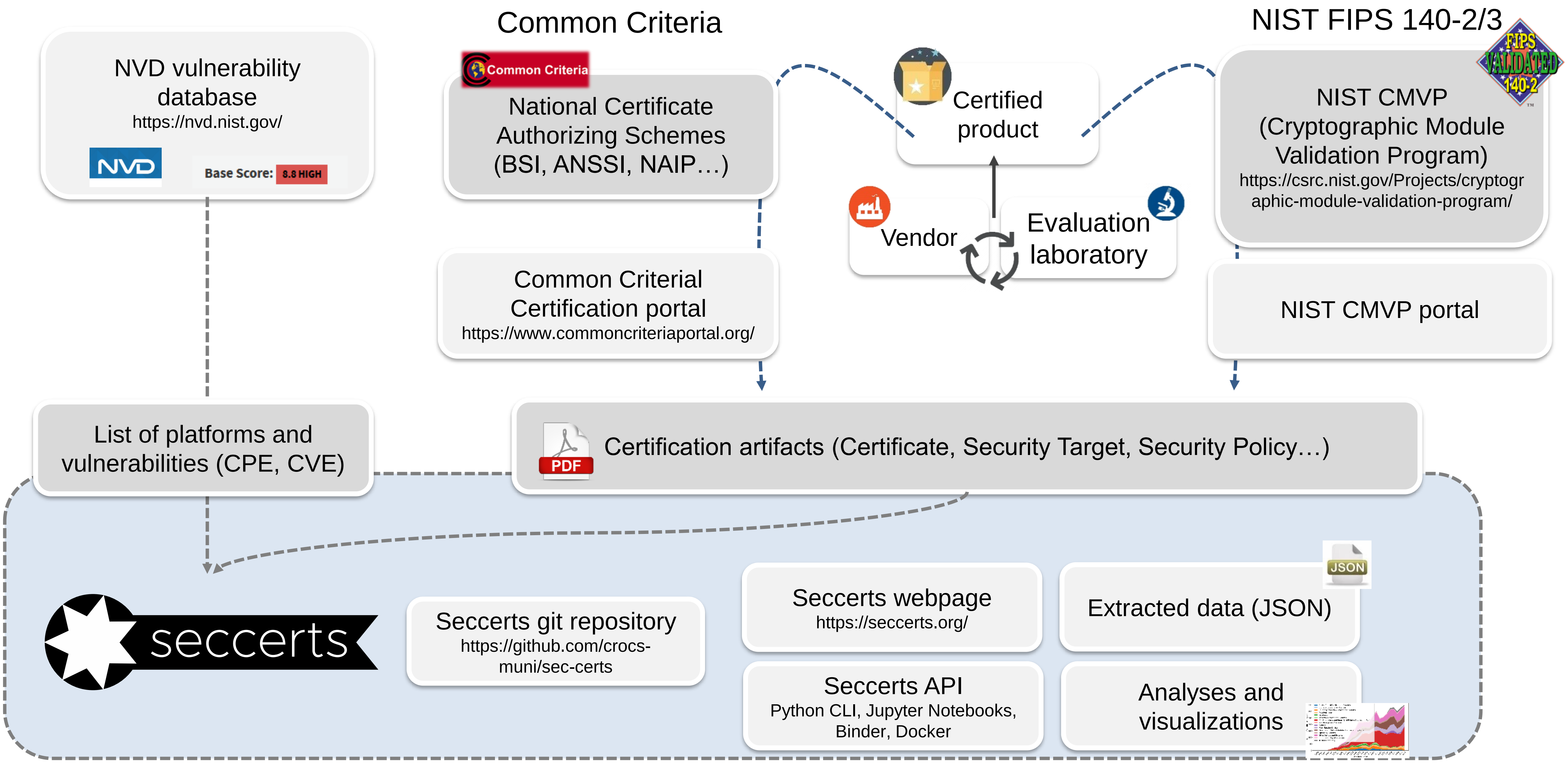
- Security target (but non-public parts), typically pdf
- Trust in eval lab, proprietary knowledge, (conflict of interests)
- Inhouse/proprietary tools, unpublished details

SECCERTS TOOL – SOME DETAILS

seccerts

- Developed since early 2020
- Fully open-source https://github.com/crocs-muni/sec-certs
- Focus on Common Criteria and NIST FIPS140 (at the moment)
- Self-hostable, programmatic Python API

https://seccerts.org/



NVD vulnerability database
https://nvd.nist.gov/

Common Criteria

National Certificate Authorizing Schemes (BSI, ANSSI, NAIP…)

Common Criterial Certification portal
https://www.commoncriteriaportal.org/

Certified product

Vendor

Evaluation laboratory

NIST FIPS 140-2/3

NIST CMVP (Cryptographic Module Validation Program)
https://csrc.nist.gov/Projects/cryptographic-module-validation-program/

NIST CMVP portal

List of platforms and vulnerabilities (CPE, CVE)

Certification artifacts (Certificate, Security Target, Security Policy…)

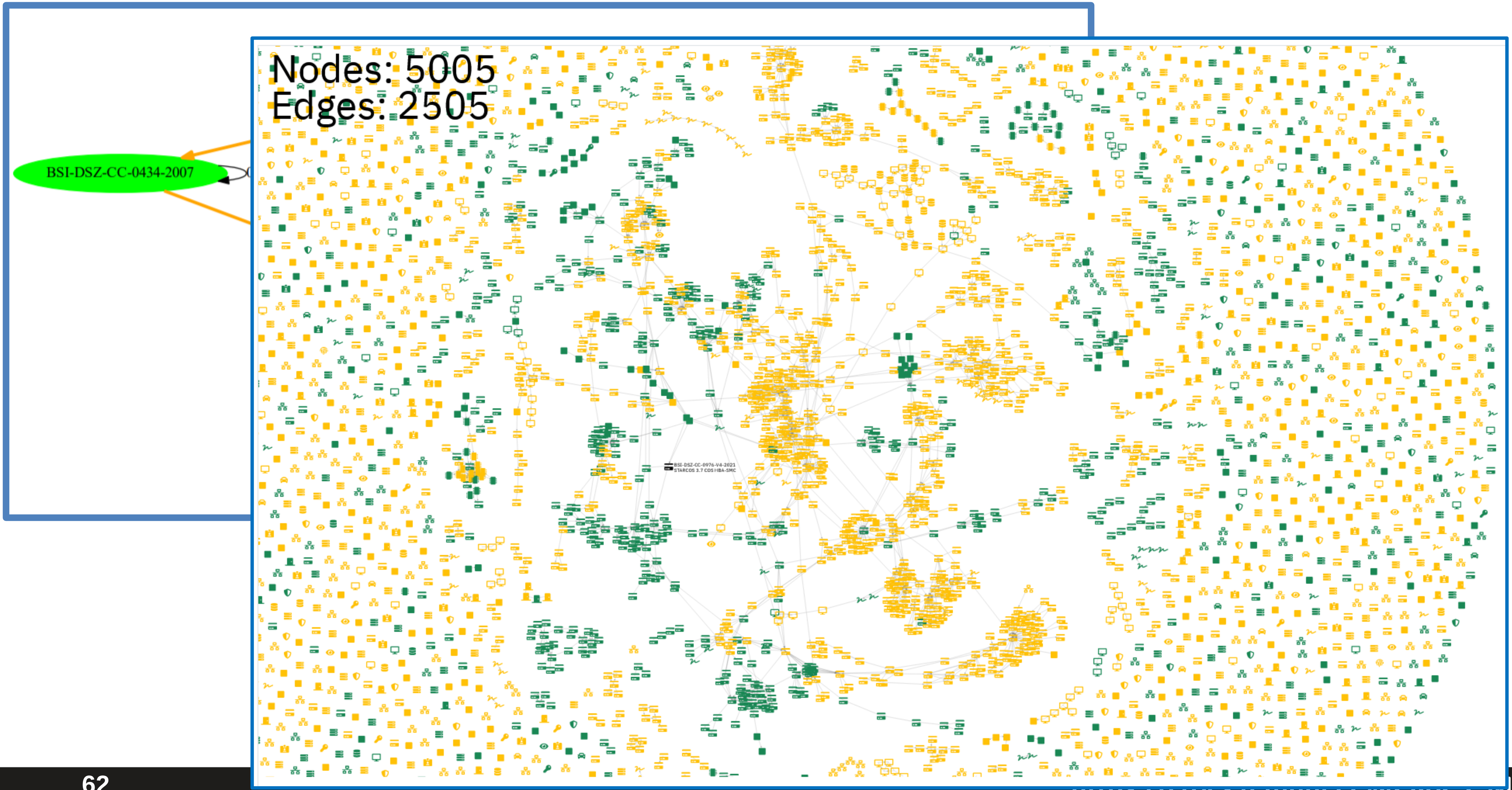
Seccerts git repository
https://github.com/crocs-muni/sec-certs

Seccerts webpage
https://seccerts.org/

Extracted data (JSON)

Seccerts API
Python CLI, Jupyter Notebooks, Binder, Docker

Analyses and visualizations


BSI-DSZ-CC-0434-2007
Nodes: 5005
Edges: 2505
BSI-DSZ-CC-0976-V4-2021
STARCOS 3.7 COS HBA-SMC

Estonia's EstEID

ID 163484

2017

Severity 3 **Root cause**

- Third party failures

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

PREMIER MINISTRE

Secrétariat général de la défense et de la sécurité nationale

Agence nationale de la sécurité des systèmes d'information

Rapport de certification ANSSI-CC-2013/55

**Plateforme jTOP INFv#46 masquée
sur composants Infineon SLE78CLX1600PM,
SLE78CLX800P et
SLE78CLX360PM**

BSI-DSZ-CC-0921-2014

Created on Jun 20, 2017 **Modified on** Jun 20, 2017

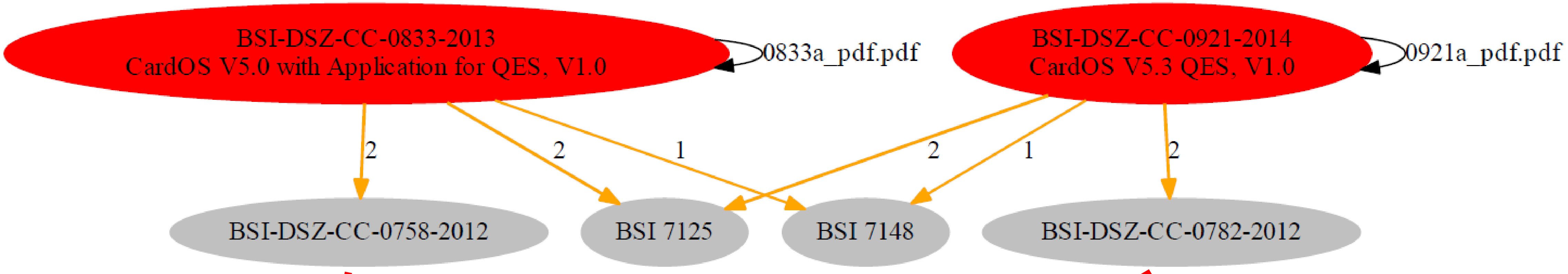
BSI-DSZ-CC-0833-2013

General description of the incident

The Austrian supervisory body has received a report on a weakness of the “asymmetric crypto library” which is used by several qualified electronic signature devices produced by Atos IT Solutions and Services GmbH, Munich, in particular • “CardOS V5.0 with Application for QES, V1.0” and • “CardOS V5.3 QES, V1.0”. The problem affects generating electronic signature creation data for use with the RSA algorithm. There is no evidence of weaknesses in generating electronic signature creation data for ECDSA or in creating electronic signatures by means of either RSA or ECDSA. Due to the mentioned weakness, a qualified trust service provider established in Austria revoked all qualified certificates issued prior to 9 June 2017 and informed both the public and the signatories affected.

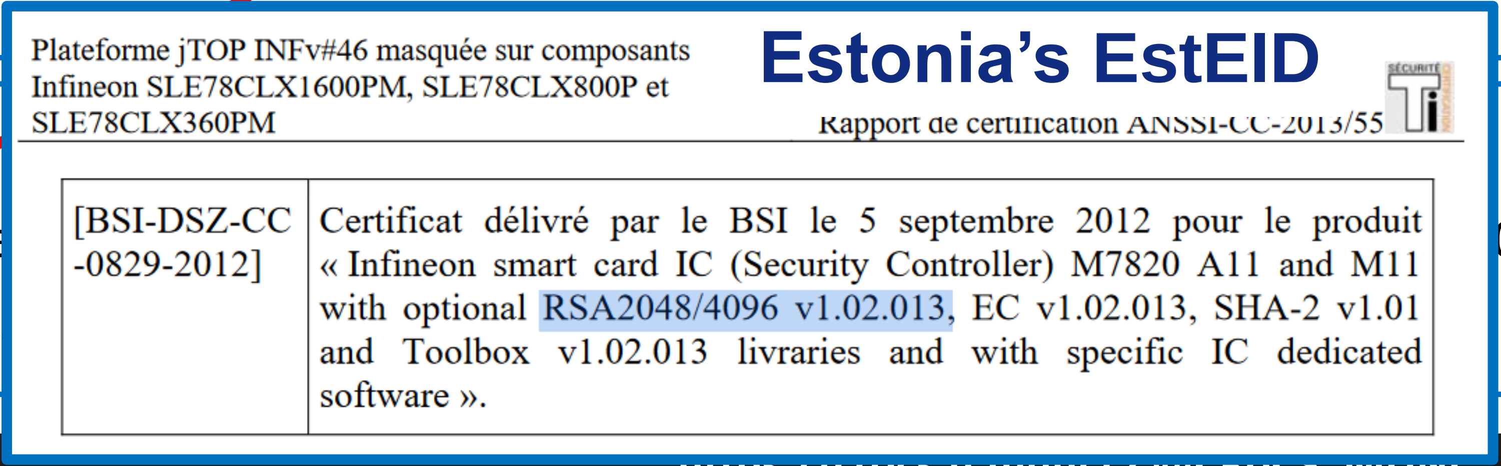
https://cybersec.ee/storage/incident-report-ID-163484-Austria.pdf

```
seccerts.py --do-find-affected BSI-DSZ-CC-0833-2013 --do-find-affected BSI-DSZ-CC-0921-2014
```



"frontpage_scan": {
  "cert_id": "BSI-DSZ-CC-0758-2012",
  "cert_item": "Infineon Security Controller M7892 A21 with optional RSA2048/4096 v1.02.013, EC v1.0
  "cert_lab": "BSI",
  "developer": "Infineon Technologies AG"

"frontpage_scan": {
  "cert_id": "BSI-DSZ-CC-0782-2012",
  "cert_item": "Infineon Security Controlle
  "cert_lab": "BSI",
  "developer": "Infineon Technologies AG"



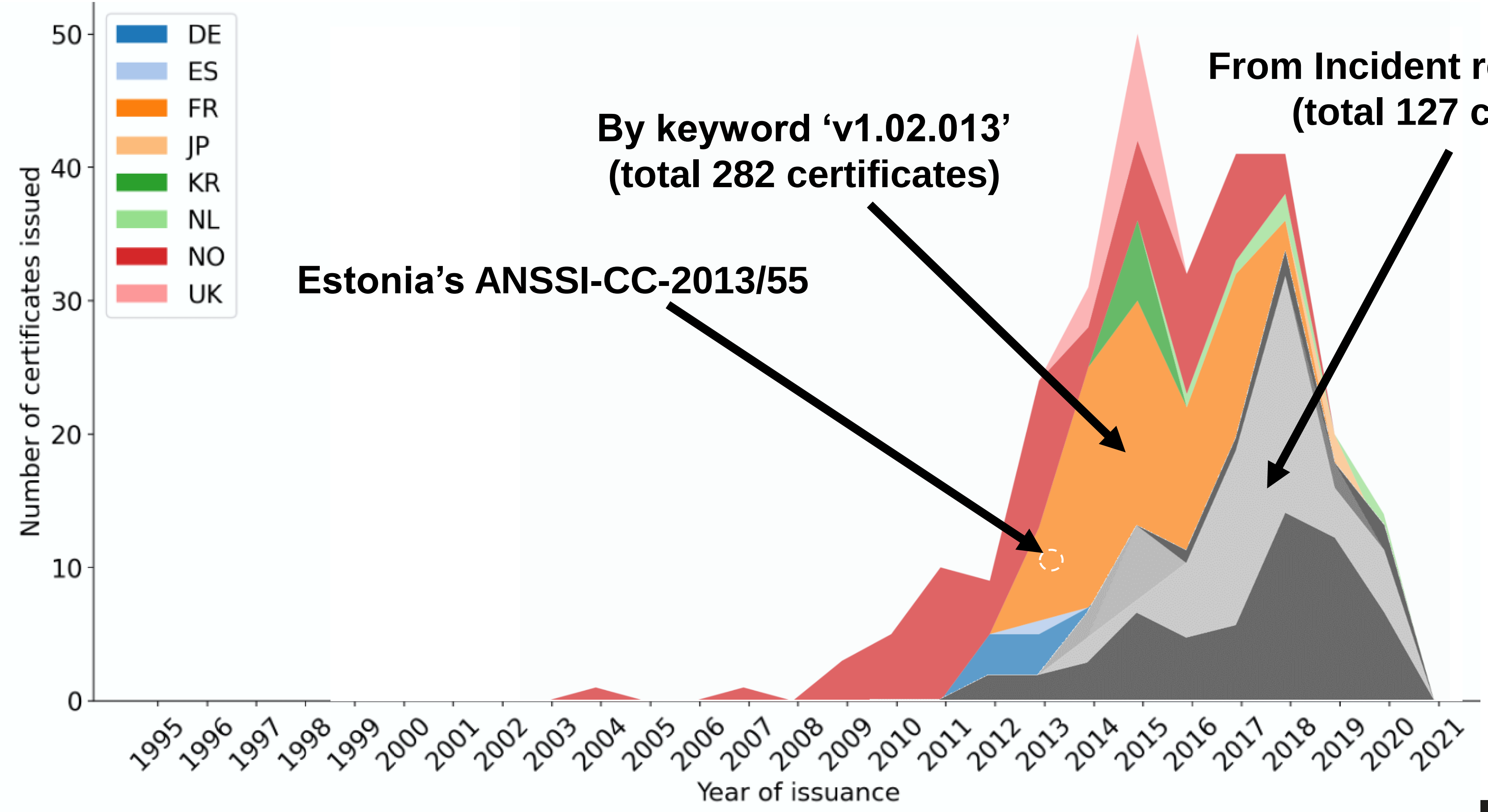
Estonia’s EstEID

Plateforme jTOP INFv#46 masquée sur composants Infineon SLE78CLX1600PM, SLE78CLX800P et SLE78CLX360PM

Rapport de certification ANSSI-CC-2013/55

[BSI-DSZ-CC -0829-2012]	Certificat délivré par le BSI le 5 septembre 2012 pour le produit « Infineon smart card IC (Security Controller) M7820 A11 and M11 with optional RSA2048/4096 v1.02.013, EC v1.02.013, SHA-2 v1.01 and Toolbox v1.02.013 livraries and with specific IC dedicated software ».

All direct and indirect references: two cert IDs from report vs. ‘v1.02.013’ keyword (RSA lib)



From Incident report ID 163484 (total 127 certificates)

By keyword ‘v1.02.013’ (total 282 certificates)

Estonia’s ANSSI-CC-2013/55

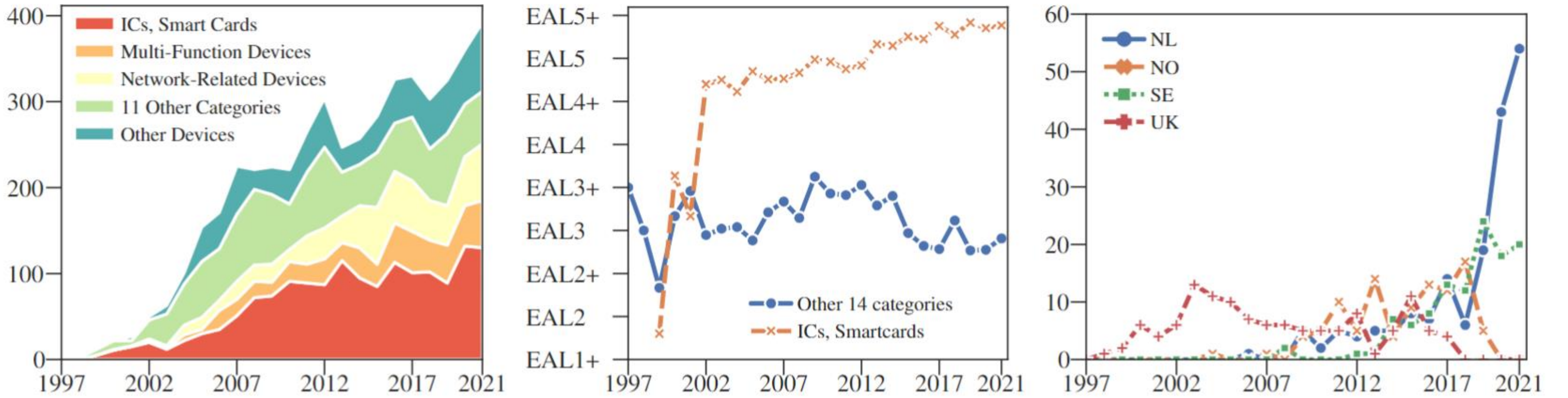
Users of the seccerts tool

- General public
  - Easy access to information (interactive webpage, info from multiple sources…)
  - Ecosystem insights: What is standardized? Change in time?
- Owners of certified devices / security researchers
  - What security claims are made?
  - Which certificates to additionally monitor?
  - Notification after new (possibly relevant) vulnerability is found
  - Analyze impact of vulnerability (e.g., ROCA case)
- Certification bodies
  - Performance of labs, suspiciously short validity, non-standard cert. claims …
  - Impact of certification requirements (SARs) on the actual security

Users of the seccerts tool

- Government agencies
  - Processing additional non-public documents
  - Attaching additional metadata (test results, powertrace…) and its governance
    - Generate seccerts “web” locally with additional information
- Certification laboratories
  - Are we comparable with other laboratories? What are the trends?
- Vendors of certified items
  - Are we under/over certifying with respect to competition?
  - Who is certifying products of our type and what were requirements in past?
- (Someone else?)

Insights into ecosystem

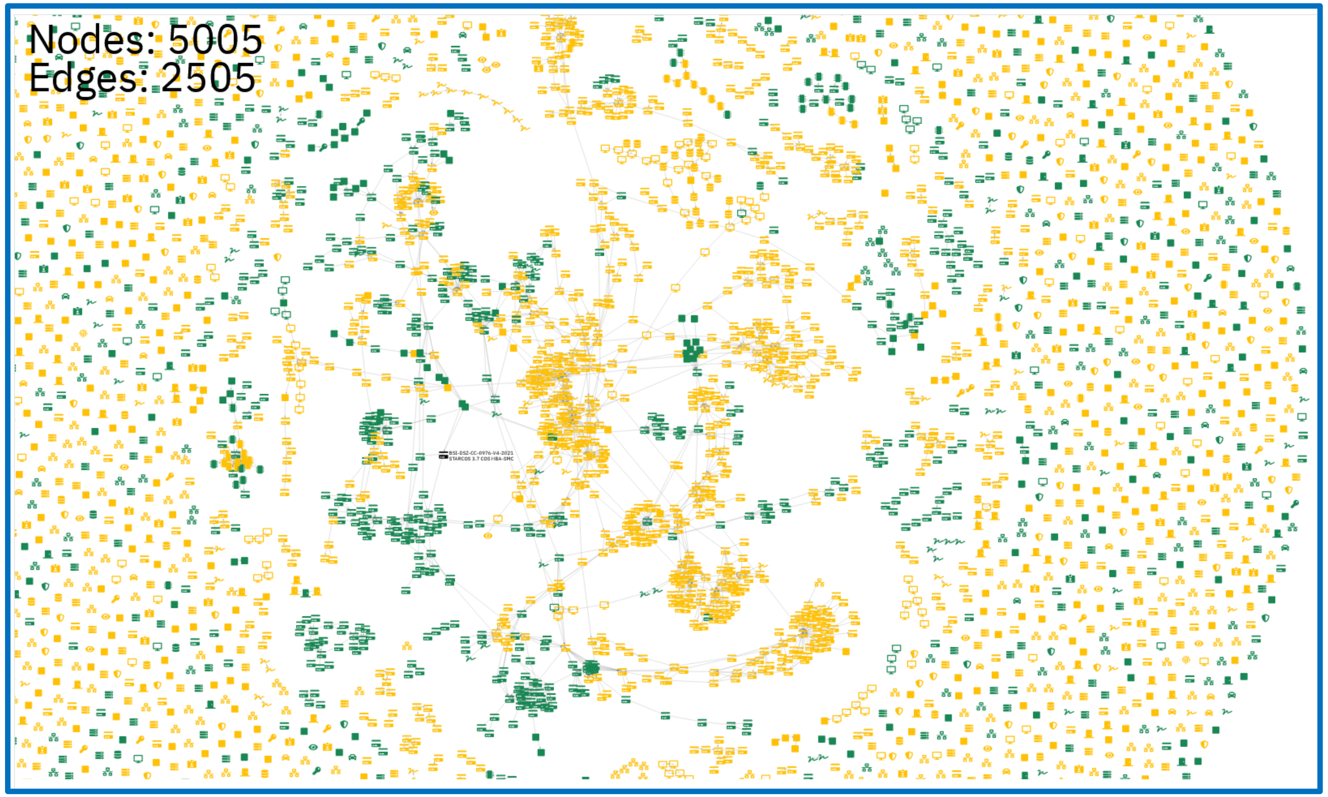


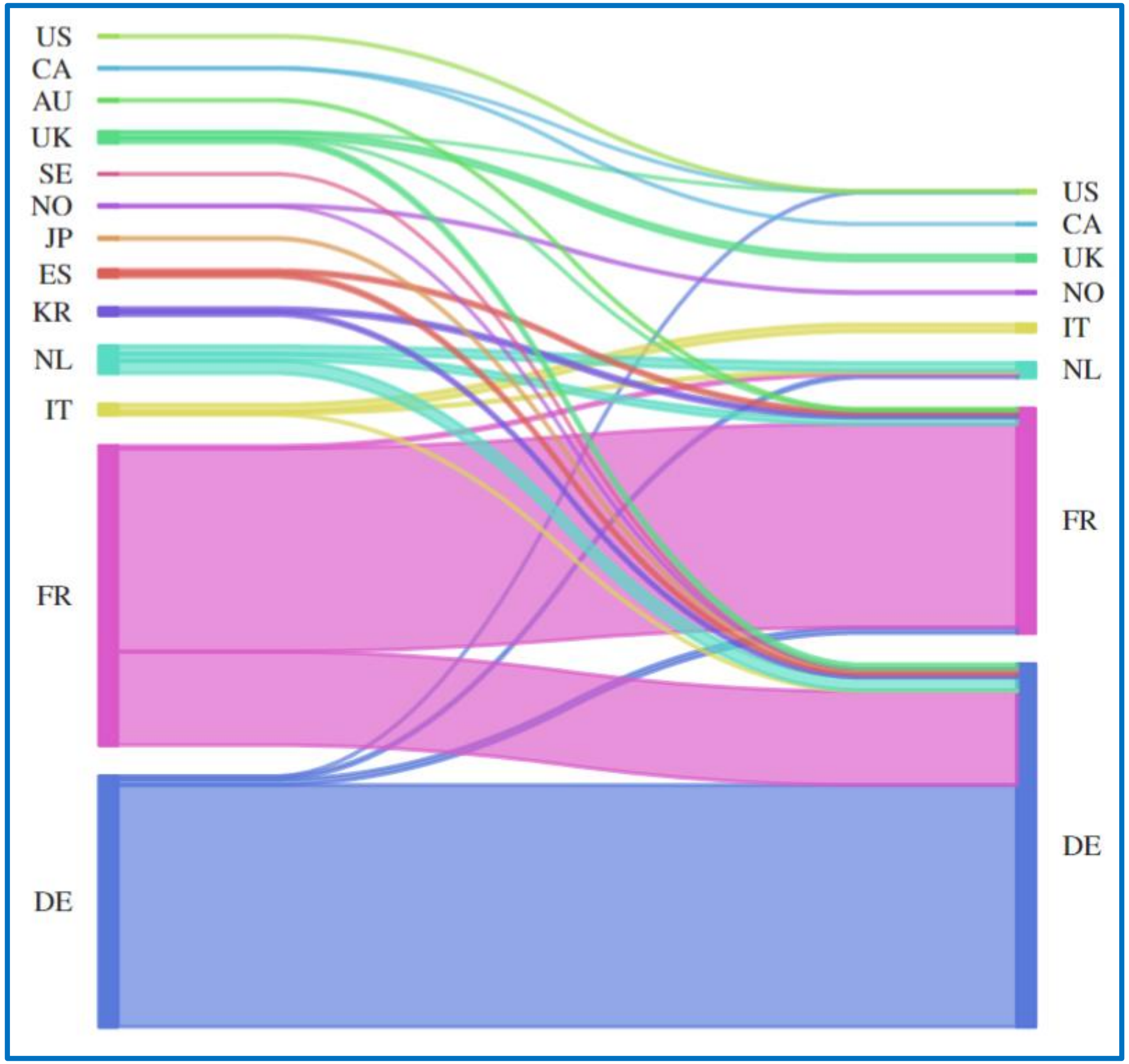
(a) Number of issued certificates in different categories.

(b) Average EAL.

(c) Number of issued certificates in selected national schemes.

Insights into ecosystem





Linking certified products to vulnerabilities

- For every certified device, we have `` `(vendor, device name, heuristically extracted versions)` ``

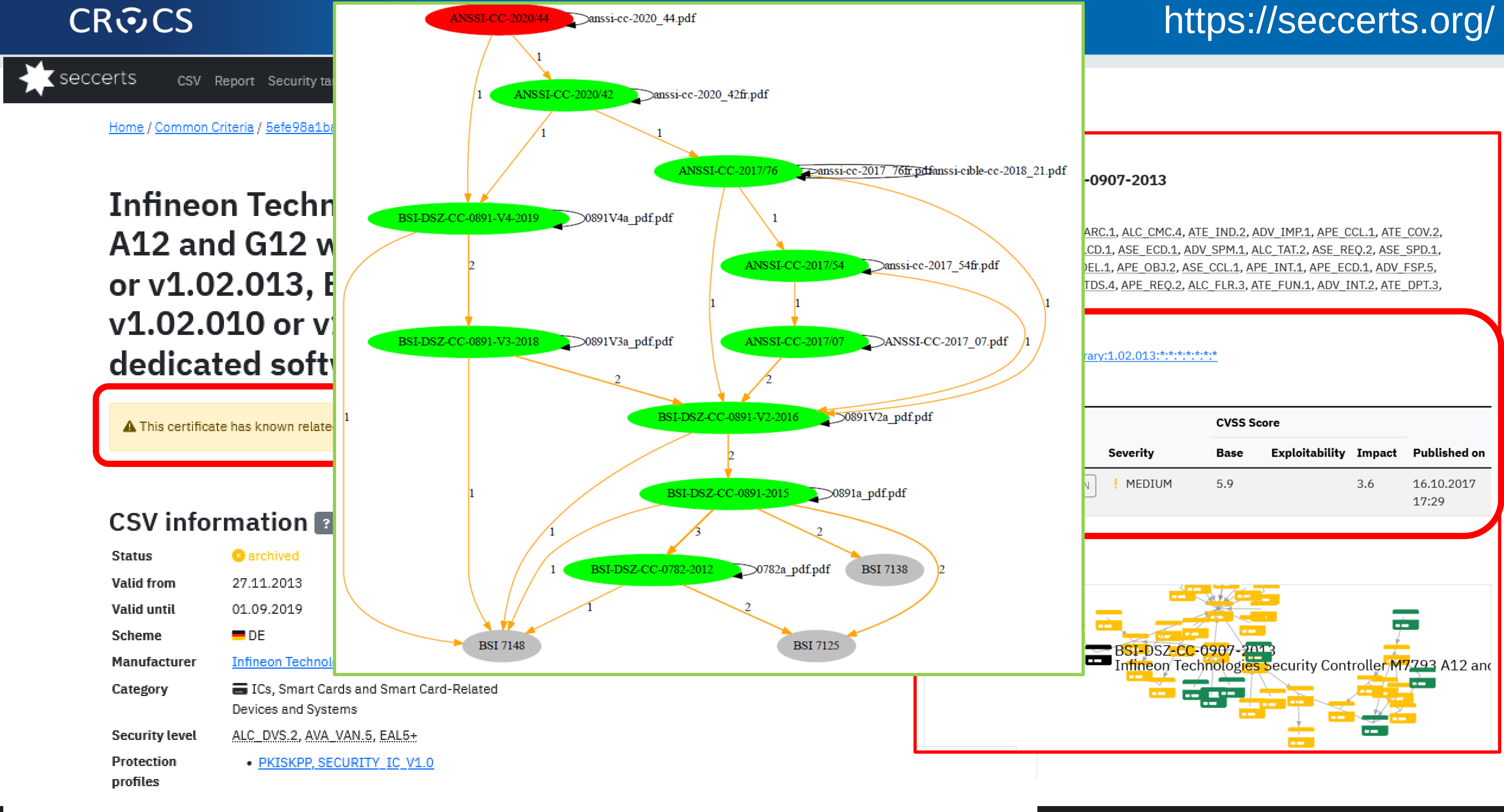
> **Infineon Technologies** Security Controller M7793 A12 and G12 with optional **RSA**2048/4096 v1.02.010 or **v1.02.013**, EC v1.02.010 or v1.02.013 and Toolbox v1.02.010 or v1.02.013 **libraries** and with specific IC-dedicated software

- Each vulnerability has a list of affected platforms specified with CPE
- RoCA vulnerability has, among others: `` `cpe:2.3:a:infineon:rsa_library:1.02.013:*:*:*:*:*:*:*` ``

💡 **Idea: Measure string similarity between certificate name and CPEs**

- Model performance $\approx 90\%$

https://seccerts.org/



seccerts CSV Report Security ta

Home / Common Criteria / 5efe98a1ba

Infineon Techn
A12 and G12 w
or v1.02.013, E
v1.02.010 or v
dedicated soft

⚠ This certificate has known relate

CSV information ?

Status	archived
Valid from	27.11.2013
Valid until	01.09.2019
Scheme	DE
Manufacturer	Infineon Technol
Category	ICs, Smart Cards and Smart Card-Related Devices and Systems
Security level	ALC_DVS.2, AVA_VAN.5, EAL5+
Protection profiles	• PKISKPP, SECURITY_IC_V1.0

-0907-2013

ARC.1, ALC_CMC.4, ATE_IND.2, ADV_IMP.1, APE_CCL.1, ATE_COV.2,
CD.1, ASE_ECD.1, ADV_SPM.1, ALC_TAT.2, ASE_REQ.2, ASE_SPD.1,
EL.1, APE_OBJ.2, ASE_CCL.1, APE_INT.1, APE_ECD.1, ADV_FSP.5,
TDS.4, APE_REQ.2, ALC_FLR.3, ATE_FUN.1, ADV_INT.2, ATE_DPT.3,

rary:1.02.013:*:*:*:*:*:*:*

	CVSS Score			
Severity	Base	Exploitability	Impact	Published on
! MEDIUM	5.9		3.6	16.10.2017 17:29

BSI-DSZ-CC-0907-2013
Infineon Technologies Security Controller M7793 A12 and

Some steps to improve certification transparency

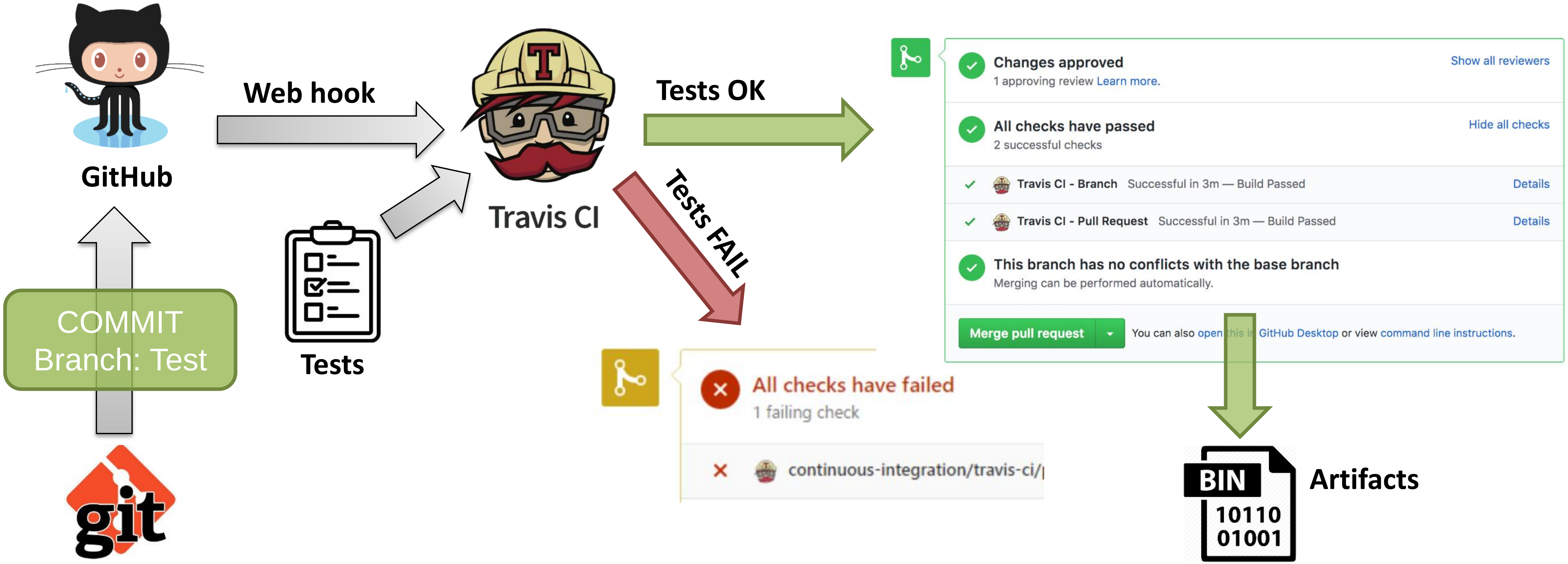
1. Better interpretation of existing CC and FIPS certificates
   - Learn more from the current database of certificates (4000+, 3000+ certs)
   - Understand what is certified when buying a product
   - Asses quickly your devices after some new vulnerability is published
2. Provide more information about device certification process
   - Ideally, user can independently replicate all certification steps
     - Requires freely available tooling (ideally open-source)
     - Requires complete log of tools and settings used
   - Ideally, "Continuous replicable certification" in the spirit of "Continuous Integration with Deterministic builds"

Some steps to improve certification transparency

3. Prepare for easy evaluation for (future) vulnerability tracking
   - Clear referencing of used components by the certified product
     - (ID + how, “pom.xml” => “dependabot-like” updates)
   - Clear references of vulnerability entries: CPE/CVE
     - Anticipate future vulnerabilities found => prefill CPE
4. Make all public data available
   - e.g., CC generates csv and html from some internal database – can we get it?
5. Make automatization of the whole process possible

How to make certification more transparent and reliable

- Inspiration from software development – continuous integration



Changes approved
1 approving review Learn more.
Show all reviewers

All checks have passed
2 successful checks
Hide all checks

✓ Travis CI - Branch Successful in 3m — Build Passed Details

✓ Travis CI - Pull Request Successful in 3m — Build Passed Details

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request ▾ You can also open this in GitHub Desktop or view command line instructions.

All checks have failed
1 failing check

× continuous-integration/travis-ci/

Take-home

seccerts

- Certificates contains trove of interesting data without NDA
- sec-certs tool released as open-source
  - Usable locally, many existing analyses, possibility for custom ones
- Ecosystem insight is possible
  - Trends in security, types of devices certified, parameters of vendors…
- Usable as tool for vulnerability analysis (both attacker and defender)
  - Assessing impact of known vulnerability, proactive monitoring
- Current certificates are written primarily for humans
  - Needs to change for automatic and more transparent certification