Practical Implementation of Lattice-based cryptography

Máire O'Neill
Queen's University Belfast

This project has received funding from the European Union H2020 research and innovation programme under grant agreement No 644729

SAFEcrypto Project

4-year H2020 project: Jan 2015 - Dec 2018

SAFEcrypto provides a new generation of practical, robust and physically secure post-quantum cryptographic solutions that ensure long-term security for future ICT systems, services and applications.

Focus is on **lattice-based cryptography** and solutions demonstrated for:

1. Satellite communications
2. Municipal Data Analytics
3. IoT

Queen's University Belfast
CSIT CENTRE FOR SECURE INFORMATION TECHNOLOGIES
THALES
HWCommunications Cyber Security and Resilience
IBM
Inria INVENTEURS DU MONDE NUMÉRIQUE
DELL EMC
RUHR UNIVERSITÄT BOCHUM
RUB
Università della Svizzera italiana

Quantum-Safe Cryptography

Lattice-based Cryptography (LBC) emerging as a promising PQ candidate

- LBC encryption and digital signatures already practical & efficient
  - NTRUEncrypt exists since 1996 with no significant attacks to date
  - LBC schemes can match and outperform ECDSA/RSA schemes
- Underlying operations can be implemented efficiently
- Allows for other constructions/applications beyond encryption/signatures - Identity based encryption, Attribute-based encryption, Fully homomorphic encryption

Family	Signature	Encryption/ KEM	Total
Lattice-based	5	23	28
Code-based	3	17	20
Multivariate	8	2	10
Hash-based	3	0	3
Isogeny-based	0	1	1
Other	2	5	7
Total	21	48	69

Lattice Based Cryptographic Building Blocks

- **Matrix vector multiplication** for standard lattices
- **Polynomial multiplication** for ideal lattices
- **Error Sampling**
  - Bernoulli sampling
  - Cumulative Distribution Table (CDT) sampling
  - Knuth-Yao sampling
  - Ziggurat sampling
  - Micciancio-Walter Gaussian Sampler
  - …

Challenges for Practical LBC Implementations

- Need to be as efficient and versatile as classical Public Key systems, such as RSA and ECC
- Embedded devices are constrained
  - No large memories
  - Limited computational power
- Choice of parameters is crucial - long-term/QC-security
  - Larger Parameters directly affects performance
  - Scalability
- Choice of Sampler
  - Different choice for signatures Vs encryption
  - Different choice for high speed Vs compact design
- Need to consider vulnerability to Side Channel Analysis

SAFE crypto

Practical Implementation of Basic Primitives

This project has received funding from the European Union H2020 research and innovation programme under grant agreement No 644729

Standard Lattices vs Ideal Lattices

- Standard lattices
  - Require computations with large matrices
  - Matrix-vector multiplication with quadratic complexity

- Ideal lattices:
  - More efficent, smaller parameters
  - Polynomial multiplication (can use NTT)

But less trust in security due to structure!

Lattice-based Encryption on FPGA

- LWE (Standard) Vs Ring-LWE (Ideal) Encryption
  - Standard LBC shown to be practical – 1272 Ops/sec on Spartan 6 FPGA

Operation and Algorithm	Device	LUT/FF/SLICE	BRAM/ DSP	MHz	Cycles	Ops/s
LWE Encrypt ($\lambda = 128$)	S6LX45	6152/4804/1866	73/1	125	98304	1272
LWE Encrypt ($\lambda = 64$)	S6LX45	6078/4676/1811	73/1	125	98304	1272
LWE Decrypt	S6LX45	63/58/32	13/1	144	32768	4395
RLWE Encrypt (Pöppelmann & Güneysu (PG), 2014)*	S6LX16	4121/3513/-	14/1	160	6861	23321
RLWE Decrypt (PG 2014)*	S6LX16	4121/3513/-	14/1	160	4404	36331
RLWE Encrypt (PG 2014)*	V6LX75T	4549/3624/1506	12/1	262	6861	38187
RLWE Decrypt (PG 2014)*	V6LX75T	4549/3624/1506	12/1	262	4404	59492
RLWE Encrypt (PG 2014)	S6LX9	282/238/95	2/1	144	136212	1057
RLWE Decrypt (PG 2014)	S6LX9	94/87/32	1/1	189	66338	2849
RLWE Encrypt (Roy et al, 2014)*	V6LX75T	1349/860/-	2/1	313	6300	49751
RLWE Decrypt (Roy et al, 2014)*	V6LX75T	1349/860/-	2/1	313	2800	109890

SAFE crypto

Frodo KEM Implementation on ARM

FrodoKEM (standard lattices) has a number of design options:

- FrodoKEM-640 (~ AES-128 security) – **total execution time of 836ms**
- FrodoKEM-976 (~ AES-192 security) – total execution time of 1.84s

PRNG implemented using AES and cSHAKE

Implementation	Platform	Security Level	Cycle counts
FrodoKEM-640-AES	Cortex-M4	128 bits	140,398,055
FrodoKEM-976-AES	Cortex-M4	192 bits	315,600,317
FrodoKEM-640-cSHAKE	Cortex-M4	128 bits	310,131,435
FrodoKEM-976-cSHAKE	Cortex-M4	192 bits	695,001,098
FrodoKEM-640-cSHAKE [pqm]	Cortex-M4	128 bits	318,037,129
KyberNIST-768 [pqm]	Cortex-M4	192 bits	4,224,704
NewHopeUSENIX-1024 [AJS16]	Cortex-M4	255 bits	2,561,438
ECDH scalar multiplication [DHH+15]	Cortex-M0	pre-quantum	3,589,850

Cycle counts for ARM Cortex-M4 implementations (at 168 MHz)

SAFE crypto

Frodo KEM Implementation on FPGA

- FrodoKEM-640 (~ AES-128 security) – **total execution time of 60ms**
- FrodoKEM-976 (~ AES-192 security) – total execution time of 135ms

Cryptographic Operation	LUT/FF	Slice	DSP	BRAM	MHz	Ops/sec
FrodoKEM-640 Keypair	6621/3511	1845	1	6	167	51
FrodoKEM-640 Encaps	6745/3528	1855	1	11	167	51
FrodoKEM-640 Decaps	7220/3549	1992	1	16	162	49
FrodoKEM-976 Keypair	7155/3528	1981	1	8	167	22
FrodoKEM-976 Encaps	7209/3537	1985	1	16	167	22
FrodoKEM-976 Decaps	7773/3559	2158	1	24	162	21
cSHAKE*	2744/1685	766	0	0	172	1.2m
Error+AES Sampler*	1901/1140	756	0	0	184	184m
NewHopeUSENIX Server [OG17]	5142/4452	1708	2	4	125	731
NewHopeUSENIX Client [OG17]	4498/4635	1483	2	4	117	653
LWE Encryption [HMO+16]	6078/4676	1811	1	73	125	1272

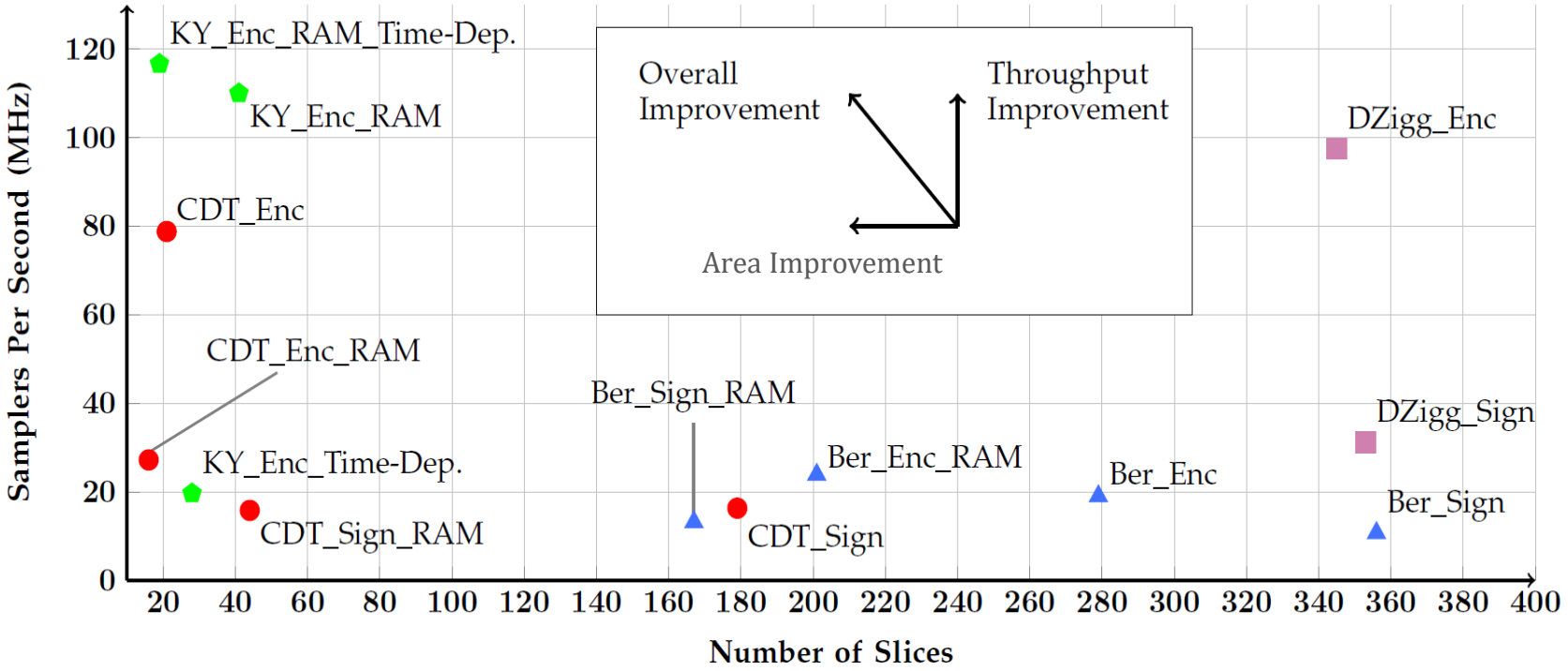
FPGA consumption and performance of designs on Artix-7

SAFE crypto

Error Sampling Evaluation in Hardware

Error Sampling is a key component in LBC - major bottleneck in practice

- *Comprehensive evaluation of Discrete Gaussian Samplers* - offers recommendations on most appropriate sampler to use for encryption, authentication, high-speed applications etc..
- Proposed *independent-time hardware designs* of a range of samplers offering security against side-channel timing attacks

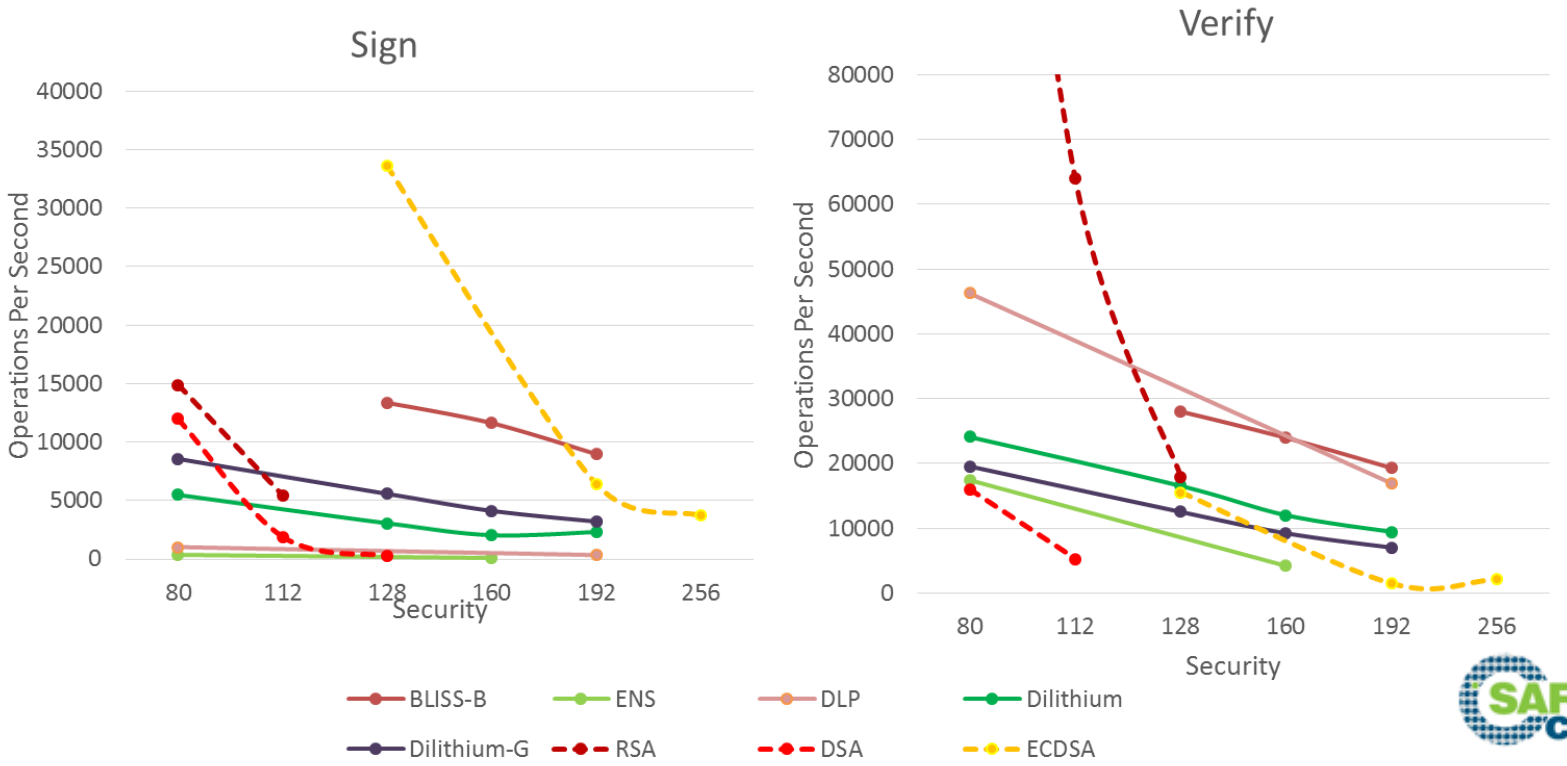


libsafecrypto: https://github.com/safecrypto/libsafecrypto

Open source software library enabling the development of lattice-based crypto solutions for commercial applications. Currently supports:

- **Signatures:** BLISS-B, Dilithium, Dilithium-G, Ring-TESLA, DLP, ENS
- **Encryption:** RLWE, Kyber
- **KEM:** ENS, Kyber

Digital Signatures: Classical vs LBC Signatures (Intel Core i7 6700 3.4 GHz)



Lattice-based Authenticated Key Exchange on ARM

Based on Generic AKE Construction:
"The Whole is less than the sum of its parts: constructing more efficient lattice-based AKEs, R. del Pino, V. Lyubashevsky, D. Pointcheval, SCN 2016

Generic AKE uses:

- KEM: **JarJar** ("*lightweight" NewHope)*
- Digital Signature: **BLISS-B**
- Hash Function: **SHA3-256**

Advantage: Common modulus q and dimension n for BLISS-B and JarJar (q = 12289 and n = 512 $\rightarrow$ synergies to reduce code size)

Lattice-based Authenticated Key Exchange on ARM

- **Implementation optimisations**:
  - Arithmetic: Barret reduction, Montgomery reduction, NTT
  - Randomness:
    - on-board TRNG for discrete Gaussian sampler (BLISS-B)
    - PRNG (ChaCha20) for binomial sampling (JarJar-Simple)
    - PRNG (ChaCha20) for delta sampling (BLISS-B)
  - Efficient hashing of long inputs: Reordering hash function inputs to make use of overlaps and avoid hashing the same inputs twice.

Algorithm	AKE (our work)	Kyber
Precomputations	517,377	6,590,440
KeyGen_A	3,900,854	7,354,193
Shared_B	5,333,723	11,940,641
Shared_A	1,124,200	7,598,468

Comparison with ported reference implementation of Kyber

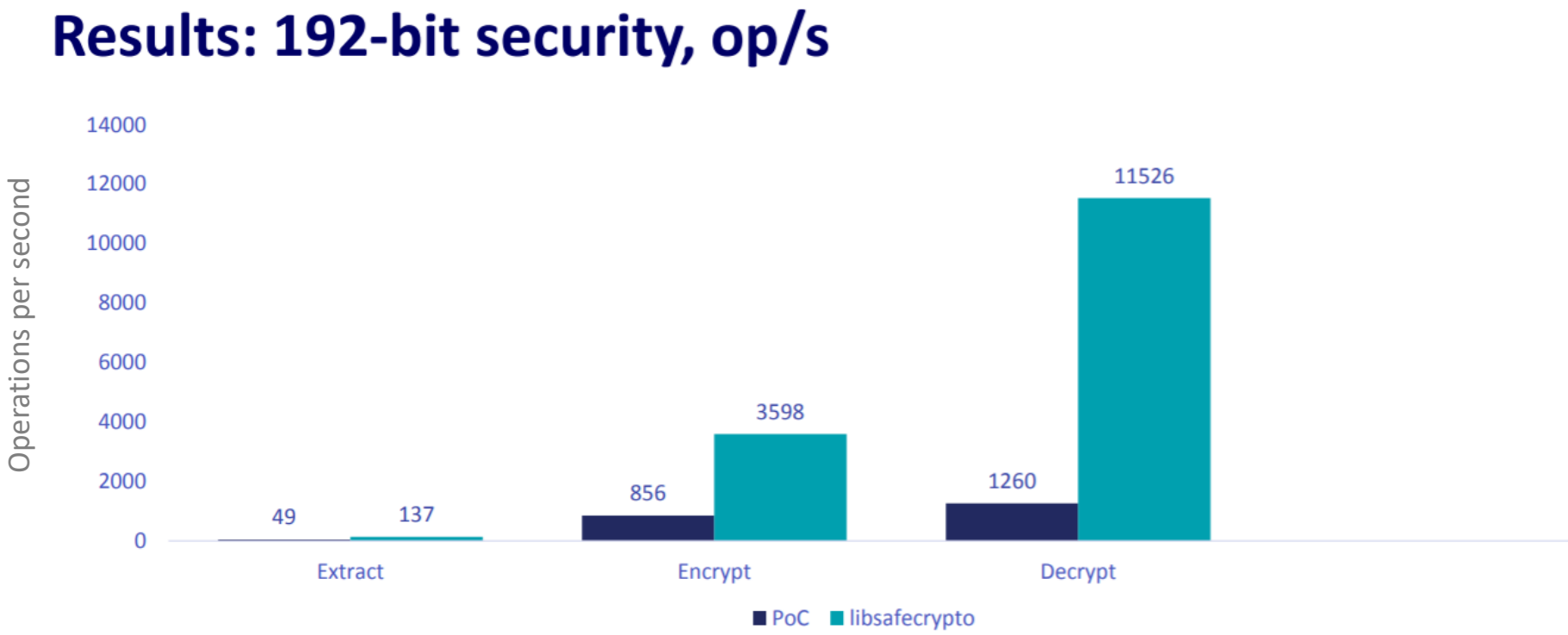
SAFE crypto

Practical Implementation of Advanced Primitives

This project has received funding from the European Union H2020 research and innovation programme under grant agreement No 644729

Practical lattice-based Identity-Based Encryption

First ANSI C Implementation of DLP-IBE Scheme[1]
(Intel Core i7 6700 3.4 GHz)



1. Ducas, L., Lyubashevsky, V., Prest, T.: Efficient identity-based encryption over NTRU lattices, pp. 22-41. Advances in Cryptology ASIACRYPT 2014, Springer

SAFE crypto

Practical lattice-based Identity-Based Encryption

Implementation of DLP-IBE Scheme on ARM Cortex-M

Operation/cycles	(512/16813057)		(1024/134348801)	
	Cortex-M0	Cortex-M4	Cortex-M0	Cortex-M4
Encryption	3,297,380	972,744	6,202,910	1,719,444
Decryption	1,155,000	318,539	2,171,000	557,015

80 bit security: 5.8ms per enc operation (Cortex-M4)

SAFE crypto

Practical lattice-based Identity-Based Encryption

Implementation of DLP-IBE Scheme on Spartan FPGAs

Implementation	Clock	(LUT|FF|BRAM|DSP)	Cycles
IBE (S6LX25, **this work**) (512/16813057)	174	(7,023 | 6,067 | 16 | 4)	13,958 9,530
IBE (S6LX25, **this work**) (1024/134348801)	174	(8,882 | 8,686 | 27 | 4)	28,586 19,535

80 bit security: 80μs per enc operation

- Results are 2 orders of magnitude faster than pairing-based IBE implementations
- **Results highlight that IBE is practical for IoT devices**

SAFE crypto

Side Channel Analysis (SCA) attacks

NIST Post-quantum Cryptography standardisation
In addition to **security**, candidates need to consider **practicality**:
1. Investigation of resistance to physical attacks
2. Development of Side Channel Attack (SCA) countermeasures

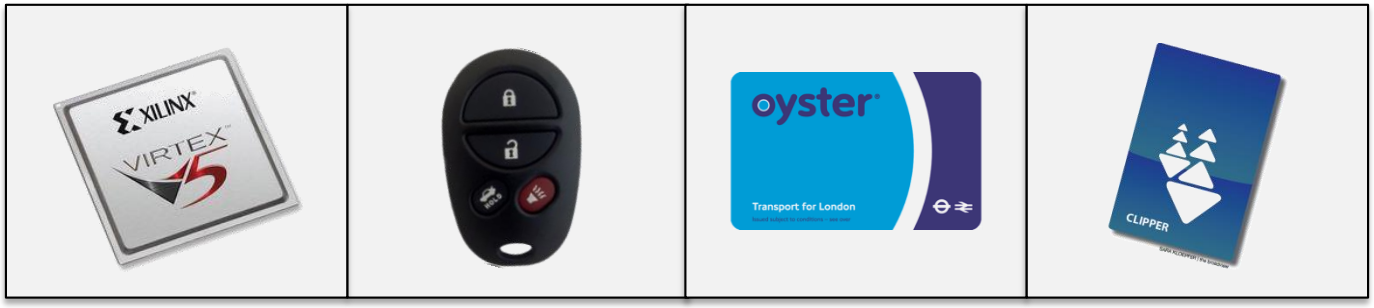
"Schemes that can be made resistant to side-channel attack **at minimal cost are more desirable** than those whose performance is severely hampered by any attempt to resist side-channel attacks"[1]

Physical security vulnerabilities of Lattice based constructions are *understudied*

1. http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/call-forproposals-final-dec-2016.pdf

SCA in the context of Lattice Based Cryptography

Side Channel Analysis (SCA) can be used to extract the secret key from electronic devices using power, EM, timing analysis, acoustics



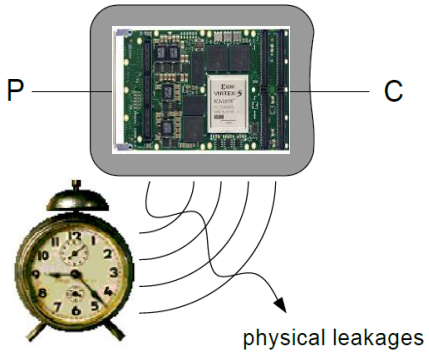
- SCA attacks and their countermeasures are an established field
  - *Why re-invent the wheel?*
- The underlying components of lattice-based schemes are *different compared to* today's prevalent symmetric/asymmetric cryptographic schemes

SAFE crypto

Timing Attacks on LBC

Timing attacks exploit the **differences in execution time** to perform an operation, e.g.,

- Different execution delays of different instructions, conditional branches
- Data fetch times due to cache memory hit/miss, attacks called *Cache attacks*



Attacks reported on lattice-based schemes target

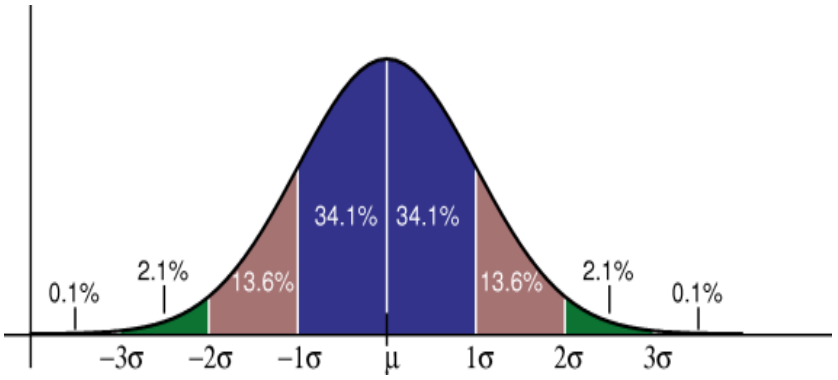
- Different number of calls to Hash function during decryption[1] (NTRU)
- Different response times for different CDT Gaussian samples[2] (BLISS)
- Attacking the shuffled Gaussian samples via a cache attack[3] (BLISS)

1. J H Silverman, W Whyte. Timing attacks on NTRUEncrypt via variation in the number of hash calls. CT-RSA, Springer, 208–224, 2007.
2. L G Bruinderink, A Hülsing, T Lange, Y Yarom. Flush, Gauss, and Reload–a cache attack on the BLISS lattice-based signature, CHES 2016, Springer, 323–345.
3. P Pessl. Analyzing the shuffling side-channel countermeasure for lattice-based signatures. INDOCRYPT 2016, Springer, 153–170

SAFE crypto

Countermeasures against Timing Attacks on LBC

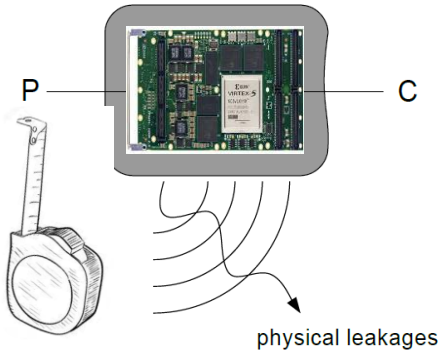
- Ensure fixed number of each function calls (hash, RNG etc.), independent of the secret values
- Ensure constant time execution times of all the functions (NTT, Gaussian Samplers)
- Randomly scramble sampler outputs, more than once.
  Multiple sampling and shuffling stages together with the use of different convolution parameters are recommended to ensure adequate protection [Pessel, 2016].



Power Analysis Attacks on LBC

Power analysis attacks extract secret information by **correlating power leakage of a device and the secret values processed** during the algorithm execution.

- Simple Power Analysis (SPA)
- Differential power analysis (DPA)
- First order DPA, Higher order DPA

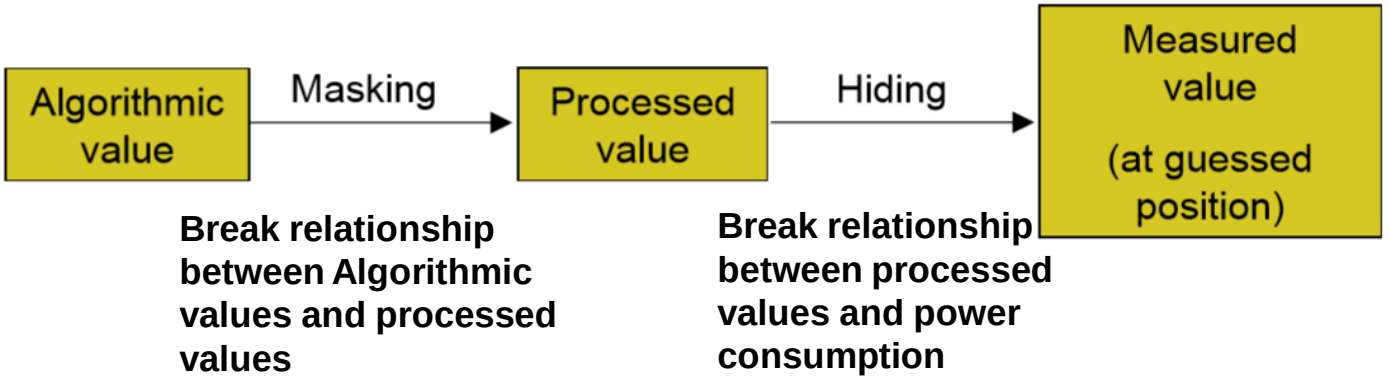


Attacks reported on lattice-based schemes target

- DIV instruction duration in ARM Cortex-M4 microcontrollers depends on the processed value[1] (RLWE)
- Difference in the hamming distance information, generated during the computation of the convolution product[2] (NTRU)

1. R Primas, P Pessl, S Mangard. 2017. Single-Trace Side-Channel Attacks on Masked Lattice-Based Encryption. CHES 2017, Springer, 513–533.
2. M-K Lee, J E Song, D Choi, D-G Han. 2010. Countermeasures against power analysis attacks for the NTRU public key cryptosystem. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 93, 1 (2010), 153–163

Countermeasures against SPA/ DPA Attacks on LBC



Masking: how to achieve?

- Masking splits the secret value into uniformly random shares and performs computations on each share individually so that an attacker needs to know every share to reconstruct the secret value.

- Masked CCA2-Secured Ring-LWE Decryption has been proposed on microcontrollers as well as on FPGAs.

Hiding: how to achieve?

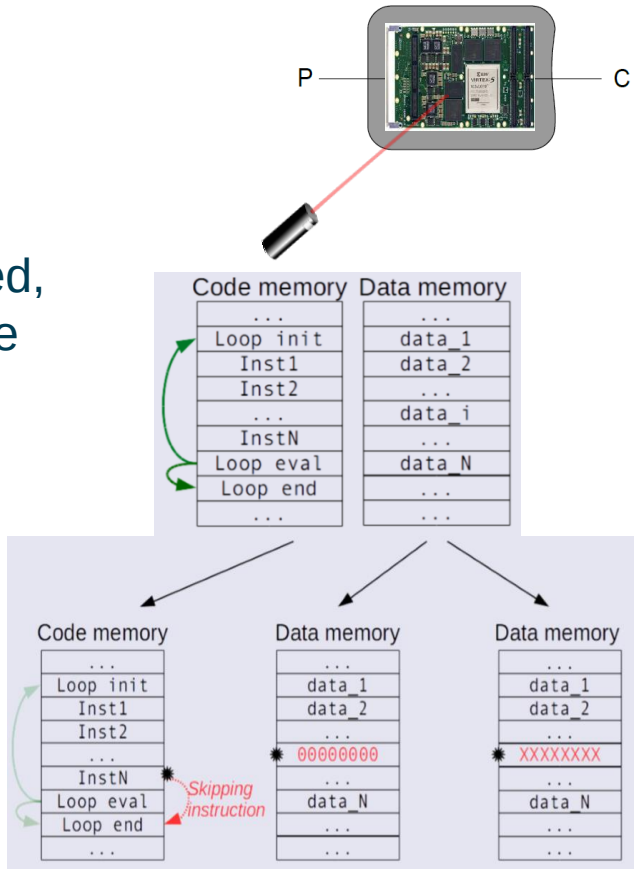
- Shuffle the order of the executed operations
- Randomly inserting instructions that don't affect the algorithm.

Fault Attacks on LBC

- **Fault attack** involves maliciously injecting an error into a device computing cryptographic operations
  - *Exploit the faulty behavior* to gather information about the secret key

- **How**: varying the supply voltage, system clock speed, ambient temperatures. Expensive and highly precise faults injected using dedicated laser beams

- **Effects:** faults shown to induce effects such as
  - changing the values of internal registers, e.g., **zeroing**
  - incorrect branching of the program, e.g., **randomization**
  - skipping of program instructions, e.g., **loop abort**



Fault Attacks on LBC

Fault attacks reported on lattice-based schemes

- Fault injection attacks have been applied to NTRU-Encrypt[1] & NTRU-Sign[2]
- A full recovery of the secret key value is possible by early loop termination of the random commitment vector and the Gaussian sample generation (BLISS,GLP,TESLA, GPV)[3]
- BLISS, ringTESLA and GLP signatures found to be vulnerable to[4]:
  - zeroing faults during the signing and verification,
  - skipping faults during the key generation and verification

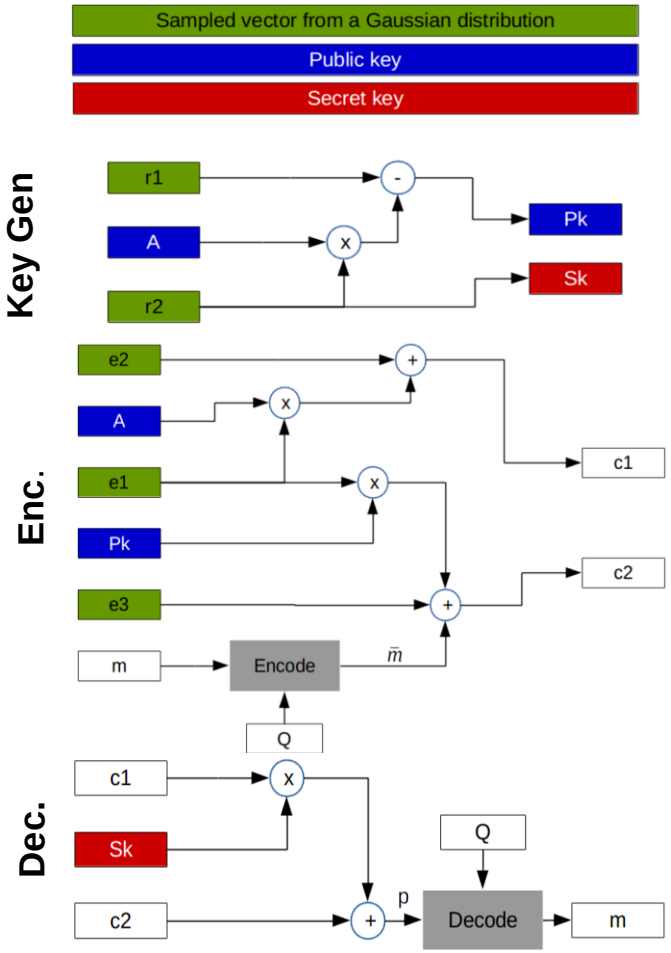
1. A. A Kamal, A M Youssef. 2011. Fault analysis of the NTRUEncrypt cryptosystem. IEICE transactions on fundamentals of electronics, communications and computer sciences 94, 4, 1156–1158, 2011
2. A. A Kamal, A M Youssef. 2012. Fault analysis of the NTRUSign digital signature scheme. Cryptography and Communications 4, 131–144, 2012.
3. T Espitau, P-A Fouque, B Gérard, M Tibouchi, Loop-abort faults on lattice-based Fiat-Shamir and hash-and-sign signatures. SAC 2016, Springer, 140–158.
4. N Bindel, J Buchmann, J Krämer. Lattice-based signature schemes and their sensitivity to fault attacks. FDTC 2016, pp. 63–77.

Fault Attacks on LBC

Fault attacks reported on lattice-based schemes

- Vulnerability of R-LWE encryption against fault injection effects:
  - *Single bit flips*
  - *Single bit zeroing*
  - *Skipping faults*

Phase	Fault	Result
Key Generation	$r_1 = 0$	Weak key generated
Key Generation	$r_2 = 0$	Weak key generated
Encryption	$e_1 = 0$	Message recovery
Encryption	$e_2 = 0$	Message recovery
Encryption	$pke_1 = 0$	Message recovery
Decryption	Zeroing secret key	Secret key recovery
Decryption	Zeroing the cipher text	Secret key recovery
Decryption	Zeroing during the NTT	Secret key recovery
Decryption	Randomization of the key	Secret key recovery



R-LWE encryption

Countermeasures against Fault Attacks on LBC

Concurrent error detection (CED) is carried out to detect Fault Injection Attacks (FIA). Two ways of achieving this:

- Duplication of hardware
- Re-computation on the same hardware

The first technique is **resource expensive**, the second one results in **performance penalty**.

Practical Case Studies

This project has received funding from the European Union H2020 research and innovation programme under grant agreement No 644729

Satellite Communications Case Study

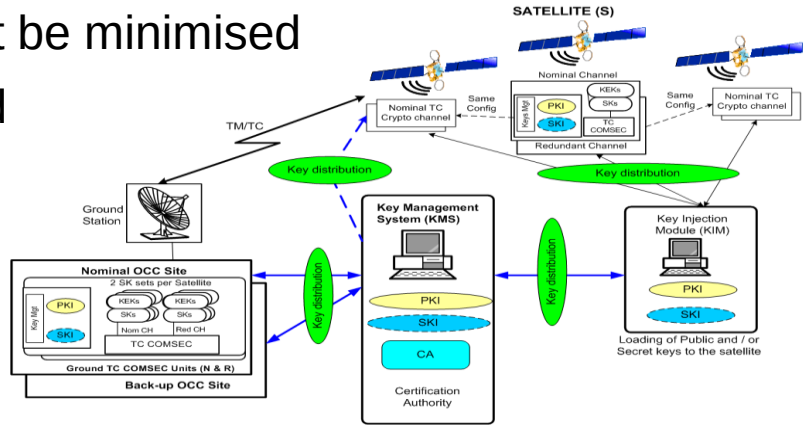
- **Currently systems tend to be owned and operated by one organisation, and built for one specific purpose**
  - Symmetric key cryptography is exclusively used
- **Cost pressures and the need for more flexibility in satellite missions is leading to repurposing of satellites and sharing of infrastructure**
  - Public key cryptography will be used, and studies have looked at this
- **Due to the longevity of satellites and associated infrastructure, any public key solution needs to be secure for a long period of time.**
  - Ideal case study for the use of Quantum Safe cryptographic solutions.

SAFE crypto

Scenario

Securing the Telecommand and Housekeeping Telemetry Channel between the Control Centre and Satellites. Main requirements:

- Due to bandwidth limitations and latency, size and no. of key management messages must be minimised
- Only “space-grade” FPGAs can be used on satellites.
  (hence gate count may be reduced)



IKEv2 (IPsec) selected as the best fit to the requirements. However:

- Current IETF standards do not support QS algorithms
  - In fact, built around DH or ECDH as key agreement mechanism
- Therefore, modifications to messages and implementations needed

We have also considered key establishment for groups of satellites

- Secure communications setup for “networks of space based entities”
- G-IKEv2 selected as best fit, and natural extension to QS IKEv2 work (basic demo produced)

SAFE crypto

Results

Thales UK have integrated SAFEcrypto implementations of lattice-based algorithms into StrongSwan (open source IPsec implementation)

- IKEv2 uses algorithms submitted to the NIST competition with SAFEcrypto contributions:
  - Kyber and Dilithium, using Software (ground) and FPGA (space-qualified)
- Demonstrated using simulated communications between ground & satellites
- Hybrid Kyber and ECDH also implemented
  - draft-tjhai-ipsecme-hybrid-qske-ikev2-01 implemented

IKEv2 Message	Message Size (bytes)	Transmission time (msecs)
Initiator INIT	970	1016
Responder INIT	1085	326.8
Initiator AUTH	5972	5018
Responder AUTH	5825	706
Total		7066.8 (~7 secs)

Figure 1 – IKEv2 transmission times for Kyber and Dilithium, assuming worst case 240ms latency, 10kbps uplink, 100kbps downlink

SAFE crypto

Results

Lessons learnt

- Straightforward to integrate LBC with IKEv2 and StrongSwan, by modifying messages
- Implementations fit in space-grade FPGAs
- Only significant issue is Dilithium signature size (and QS signature size more generally) compared to non-QS (e.g. ECDSA).
  - Has a significant effect on performance for this use case. Still meets satellite application requirements, but could be an issue in other low bandwidth use cases.
- Hybrid approach is attractive for risk averse customers, and could be first deployment

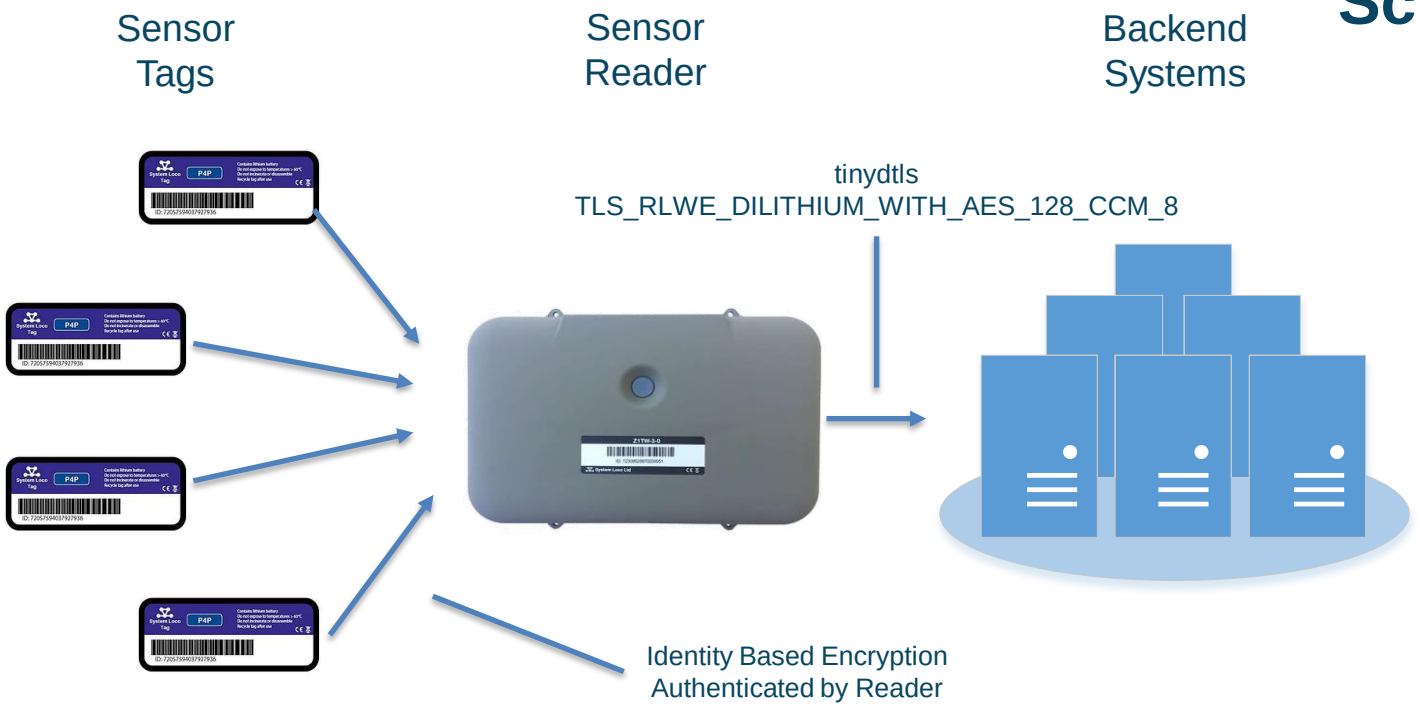
SAFE crypto

IoT Case Study

- **Concerns are often raised that the IoT is being developed rapidly without appropriate consideration of security**
  - Security is often added as an afterthought, or not at all, relying on pure transport stream mechanisms.
- **With a large number of potential devices to be configured with keys, key management can be problematic**
  - New schemes such as Identity Based Encryption (IBE) are considered, with demonstration on embedded hardware (ARM Cortex M series).
- **Efficiency is key on frequently battery powered IoT devices, when running on embedded platforms**
  - Implementation on low power sensor devices is shown, with post quantum identity and access management.

SAFE crypto

Scenario



DTLS used for communication with backend cloud services

- Constrained Application Protocol (CoAP) aimed at low power IoT devices.
  - Post quantum methods patched into an implementation of tinydtls, used within the LibCoAP library.

Identity based encryption used on tags, provisioned at device creation

- Allows sensor reader to be able to authenticate tags.
- IBE provisioned at device creation based on Tag Serial ID.

SAFE crypto

Results

HWC have integrated SAFEcrypto implementations of LBC algorithms into tinydtls (open source iot dtls library)

- Post Quantum key agreement protocols established extending the protocols for negotiation for establishing a DTLS connection.
- DILITHIUM and BLISS based methods both implemented.
- Elliptic Curve and PSK fallback options still remain as available algorithms.

IBE on Embedded device possible

- Viable option for sensor device authentication and provisioning.

Lessons learnt

- Adding post quantum algorithms into tinydtls relatively simple by appending the supported Cipher Suite during the Cipher Suite Negotiation stage.
- Post Quantum a viable option for IoT devices.

Municipal Data Analytics Case Study

Potential for big data analytics to help in the reduction of crime, improved health care efficiency and decreased cost of government

- Big data platforms increasingly moving to Cloud providers.

Increasing concern about privacy in municipal data sets

- Multi-tenant, shared environments pose new threats to the privacy of data. Encryption is one of the primary means by which this threat is mitigated.

Due to shared nature of the environment, communications over untrusted networks, and the longevity of some sensitive data relating to a municipality and its citizens, long term strategies to protect data are needed.

- Personally Identifiable information(PII)
- Privacy of behaviour
- Privacy of personal communication
- It is thus an ideal case study for the use of Quantum Safe cryptographic solutions.

SAFE crypto

Results

(1) A KMIP client supporting LBC keys was developed & integrated with Dell EMC's Key Trust Platform (KTP)

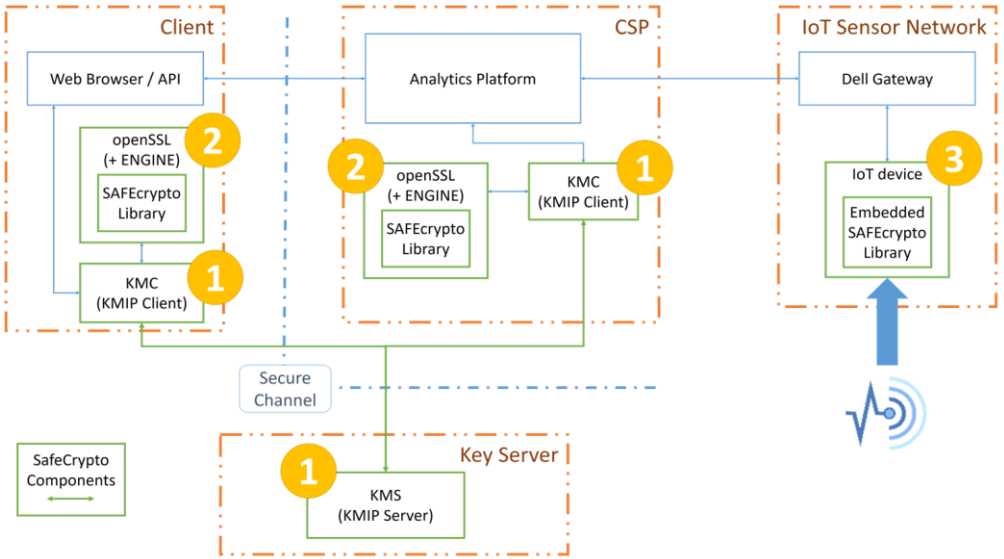
- Demonstrates the distribution of lattice key material

(2) An openSSL ENGINE was developed integrating libsafecrypto

- Demonstrates the generation of lattice key material, and availability of lattice key algorithms using industry standard library and API

(3) Environmental sensor PoC using Dilithuim to digitally sign the data

- PoC demonstrating lattice digital signatures on embedded devices



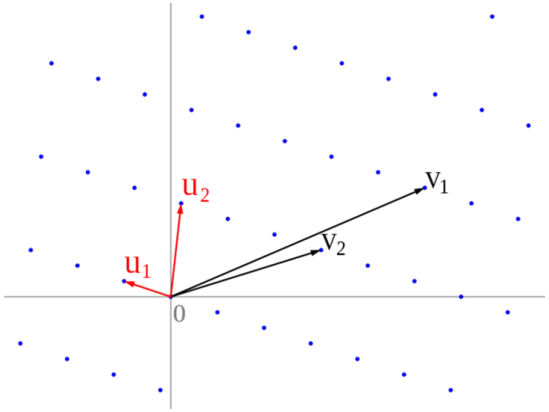
Results

Lessons learnt

- openSSL integration via ENGINE component is possible, but new cipher suites will require closer integration with core openSSL codebase
- Industry efforts to influence the KMIP standard need to continue to achieve lasting impact
  - Uncertainty over NIST process is an additional hurdle to moving standards

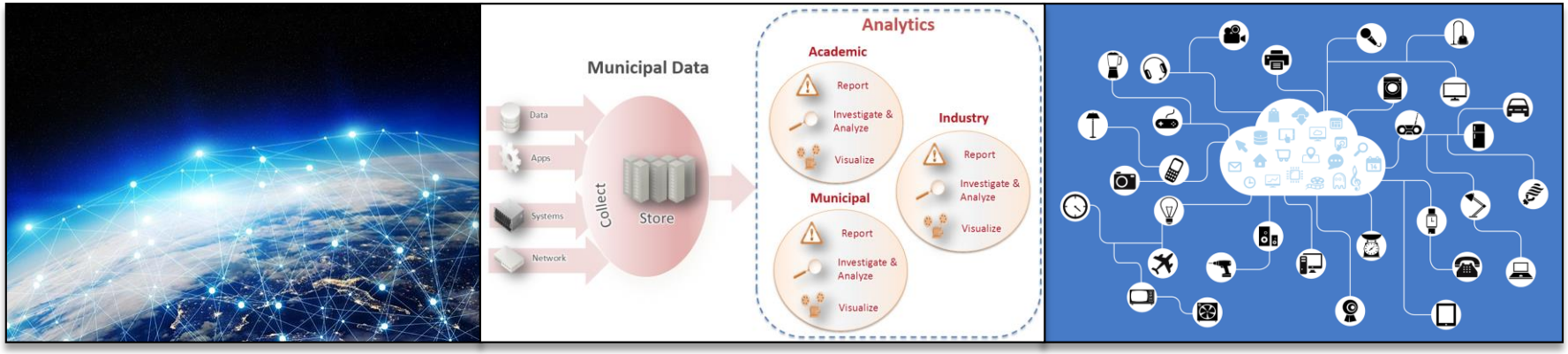
Conclusions

- Lattice-based cryptosystems are a **promising Post-Quantum cryptography solution** for long-term security applications
- LBC **offers versatility** in the range of cryptosystems it can support
- **Practical Implementations of lattice-based schemes** possible:
  - Standard LWE, RLWE Encryption
  - Frodo KEM
  - Dilithium, Kyber, RingTESLA, BLISS-B
  - Lattice-based AKE
  - Lattice-based IBE



Conclusions

- Important to **consider SCA countermeasures appropriate to LBC** and their effect on performance.
- SAFEcrypto outputs demonstrate that ***Lattice-based cryptography can meet the requirements of real world scenarios***.



Project Deliverables and Publications can be found at www.safecrypto.eu