COSIC

imec embracing a better life

KU LEUVEN

New developments in cryptography land

SecAppDev

Bart Preneel
COSIC KU Leuven and imec
Bart.Preneel(at)esat.kuleuven.be @bpreneel1
14 June 2022

Definitions

1970s

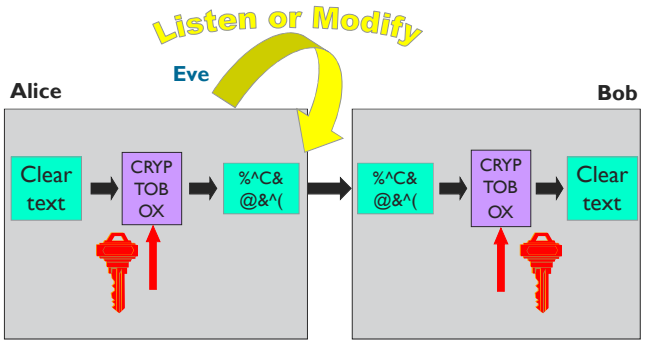
Confidentiality
Integrity
Availability

	data	entities
confidentiality	encryption	anonymity
authentication	data authentication	"identification"

Don't use the word authentication without defining it

- Authorisation
- Non-repudiation of origin, receipt
- Contract signing
- Notarisation and Timestamping

Cryptology: principle



Outline

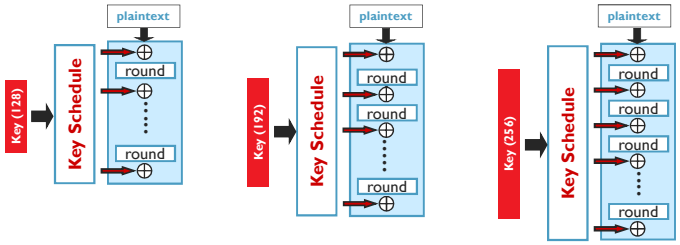
- Authenticated encryption
- Post-quantum cryptography
- Computing on Encrypted Data

AES variants (2001)

AES-128
10 rounds
Sensitive/classified (SECRET)

AES-192
12 rounds
Classified (TOP SECRET)

AES-256
14 rounds
Classified (TOP SECRET)

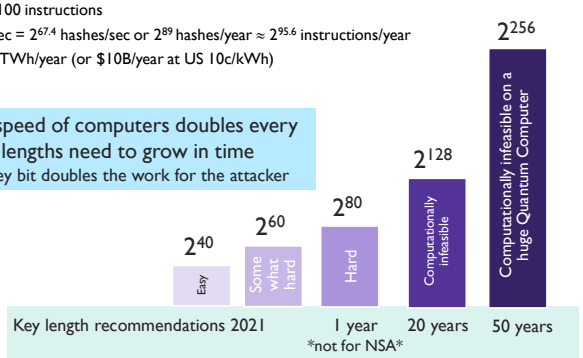


Exhaustive key search

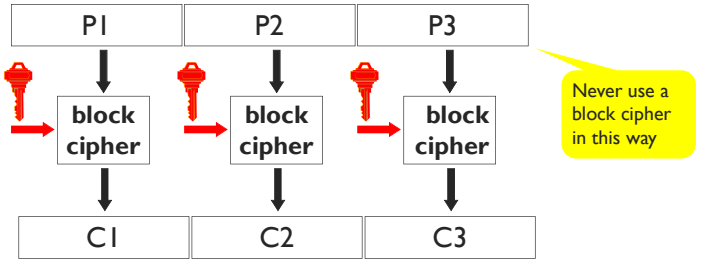
2022: 1 million machines with 16 cores @ 4 GHz can do 2^{56} instructions/sec or 2^{80} instructions/year
- trying 1 key ≈ 100 instructions

Bitcoin: 200 Exahashes/sec = $2^{67.4}$ hashes/sec or 2^{89} hashes/year ≈ $2^{95.6}$ instructions/year
- Electricity: 100 TWh/year (or $10B/year at US 10c/kWh)

Moore's "law": speed of computers doubles every 18 months: key lengths need to grow in time
but adding 1 key bit doubles the work for the attacker



Block cipher

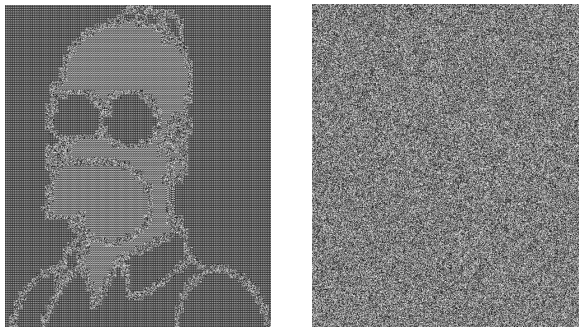


Never use a block cipher in this way

- larger data units (blocks): 64…128 bits
- memoryless
- repeat simple operation (round) many times

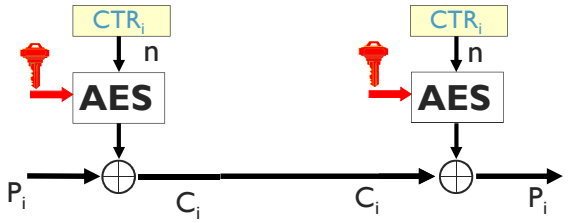
An example plaintext

Encrypted with AES in ECB and CTR mode



CounTer Mode (CTR)

$C_i = P_i \oplus E_K(CTR_i), CTR_i ++$



state initialized $CTR_0 = IV$

all 0? or random? or nonce? or ….?

CTR: properties

- different IV necessary; otherwise insecure (Venona)
- uses only encryption
- key stream independent of plaintext: can be pre-computed
- no error propagation: errors are only copied
- random access on decryption
- optimal for hardware:
  - parellellism: one can process multiple counter values at the same time
  - pipelining: no need to know the ciphertext block corresponding to the current plaintext block to start processing the next plaintext block
- risk: what if counters are (accidentally) reset to same value?

Encryption limitations

- Typically does not hide the **leng**th of the plaintext (unless randomized padding but even then…)
- Ciphertext becomes random string: "normal" crypto does not encrypt a credit card number into a (valid) credit card number
- Does **not** hide existence of plaintext (requires steganography)
- Does **not** hide that Alice is talking to Bob (e.g. Tor)
- Does **not** hide traffic volume (requires dummy traffic)
- Does **not** protect against modifications

Data authentication: the problem

- Bob wants to know:
  - the **source** of the information (data origin)
  - that the information has not been **modified**
  - (optionally) the **destination** of the information
  - (optionally) **timeliness** and **sequence**

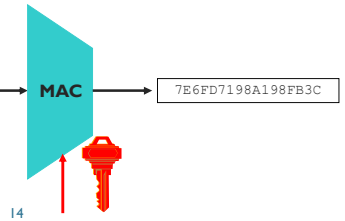
There are no applications that require encryption without data authentication (but this can still be found in legacy applications with as excuse performance)

Data authentication: MAC algorithms

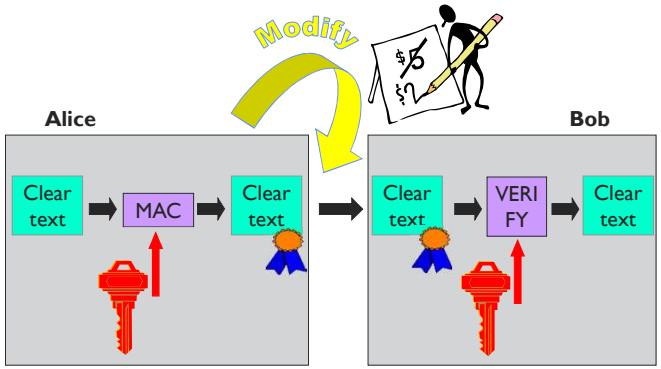
- Replace protection of authenticity of (long) message by protection of secrecy of (short) key
- Append MAC to the plaintext

CBC-MAC (CMAC/LMAC)
HMAC
GMAC

This is an input to a MAC algorithm. The input is a very long string, that is reduced by the hash function to a string of fixed length. There are additional security conditions: it should be very hard for someone who does not know the secret key to compute the hash function on a new input.



Data authentication: MAC algorithms



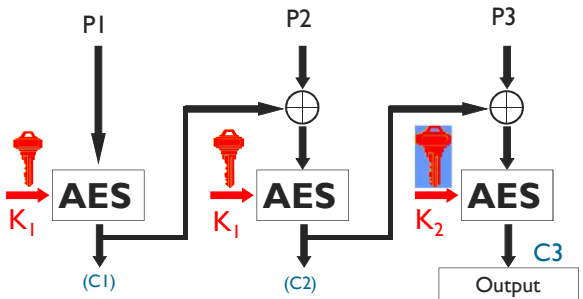
Data authentication: MAC algorithms

- typical MAC lengths: (32)..64..96 bits
  - forgery attacks: 2^m steps with m the MAC length in bits
- typical key lengths: (56)..112..160 bits
  - exhaustive key search: 2^k steps with k the key length in bits
- birthday attacks: security level smaller than expected

MAC algorithms

- Banking: CBC-MAC based on triple-DES
- Internet: HMAC and CBC-MAC based on AES
- information theoretic secure MAC algorithms (authentication codes): GMAC/Poly1305
  - rather efficient
  - part of the key refreshed per message

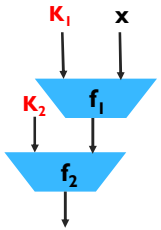
CBC-MAC based on AES (LMAC)



Secure up to 2^{50} message blocks

MAC based on a hash function

- **HMAC**:
  - $h_K(X) = h(h(K_1||x)||K_2)$
  - not secure with MD4/MD5
  - Ok with SHA-1/SHA-2/SHA-3



GMAC: polynomial authentication code
(NIST SP 800-38D 2007 + 3GSM)

keys $K_1, K_2 \in GF(2^{128})$

input x: $x_1, x_2, \ldots, x_t$, with $x_i \in GF(2^{128})$

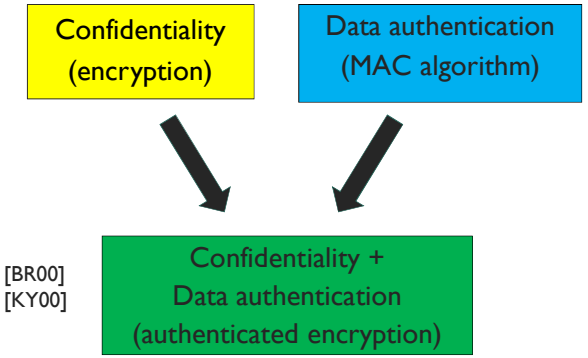
$$g(x) = K_1 + \Sigma_{i=1}^{t}\ x_i \bullet (K_2)^i$$

compute $K_1 = AES_K(n)$ (CTR mode)

properties:

- fast in software and hardware (support from Intel/AMD)
- not very robust w.r.t. nonce reuse, truncation, MAC verifications, due to reuse of K_2 *(not in 3G/4G!)*
- versions over GF(p) (e.g. Poly1305-AES) is more robust as key depends on nonce and keystream

Authenticated Encryption



[BR00]
[KY00]

Authenticated Encryption

Generic composition [BN'00][NRS'14]

- Encrypt-then-MAC with 2 independent keys
  - IPsec, TLS 1.2, 1.3
- MAC-then-Encrypt with 2 independent keys
  - TLS 1.1 and older, 802.11i WiFi
- MAC-and-Encrypt with 2 independent keys

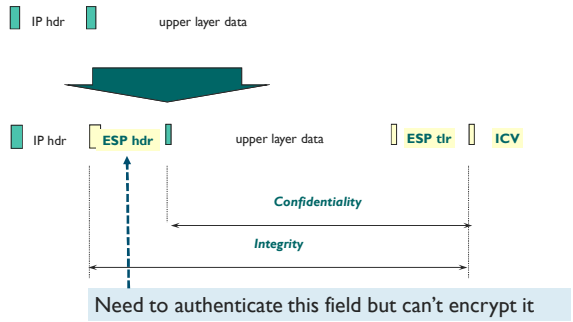
Design "from scratch"

- Integrated authenticated encryption schemes: combined operation with 1 key: see next slide

Authenticated Encryption: properties wish list

- Associated data
- Parallelizable
- Online for encryption
- Security reduction
- Resistance to nonce reuse
- Incremental tags
- Fragmentation
- No release of unverified plaintext
- Flexible implementation sizes
- Performance: speed/size
- Secure implementations: constant time/power analysis/EM analysis/fault attacks…

IPsec - ESP Transport mode



Need to authenticate this field but can't encrypt it

Long history to achieve AE based on block ciphers

- Ad hoc schemes
  - Jueneman ('80s)
  - PCBC (Kerberos)
  - iaPCBC
  - WEP for WiFi 802.11
- 1st schemes with proofs
  - RPC [Katz-Yung'00]
  - IAPM [Jutla'01]
  - XECB and XCBC [GD'01]
  - OCB1, OCB2, OCB3 [RBBK'01]
- 2nd generation
  - CCM
  - GCM
  - EAX
  - CWC
- 3rd generation
  - Chacha20-Poly1305
  - GCM-SIV (RFC 8452)
  - BTM
  - McOE-G
  - Aegis

AE: block cipher based

	# passes	//	Online (encr)	Nonce Misue	Patents (but all expired)
IAPM	1	√	√		√
XECB	1	√	√		√
OCB	1	√	√		√
CCM	2				
GCM	1*	√	√		
EAX	2		√		
CWC	2	√	√		
AEGIS	1	√	√		
GCM-SIV	2			√	
BTM	1	√		√	
McOE-G	1*		√	√	

Nonce reuse in practice

- H. Böck, A. Zauner, S. Devlin, J. Somorovsky, P. Jovanovic, Nonce-Disrespecting Adversaries: Practical Forgery Attacks on GCM in TLS, Black Hat 2016
  - Affects 184 https servers
  - Affects Ruby if nonce is set before the key
- Samsung Galaxy: ARM Trustzone implementation
  - Exporting key encrypted under Hardware Derived Key in GCM mode
  - IV provided by the application
  - A. Shakevsky, E. Ronen A. Wool, Trust Dies in Darkness: Shedding Light on Samsung's TrustZone Keymaster Design, https://eprint.iacr.org/2022/208.pdf

- We should make cryptographic primitives as robust as possible
- If developers can't follow instructions and understand constraints, will they be able to properly generate and manage keys?

Caesar competition for Authenticated Encryption

2013-2019 https://competitions.cr.yp.to/caesar.html

	Name	Designers
Lightweight	Ascon	C. Dobraunig, M. Eichlseder, F. Mendel, M. Schläffer
	ACORN	H. Wu
High speed	Aegis	H. Wu, B. Preneel
	OCB	T. Krovetz, P. Rogaway
Robust	COLM	J. Jean, I. Nikolić, T. Peyrin, Y. Seurin
	AES-COPA	E. Andreeva, A. Bogdanov, N. Datta, A. Luykx, B. Mennink, M. Nandi, E. Tischhauser, K. Yasuda

Selected from 52 submissions – a 5-year effort

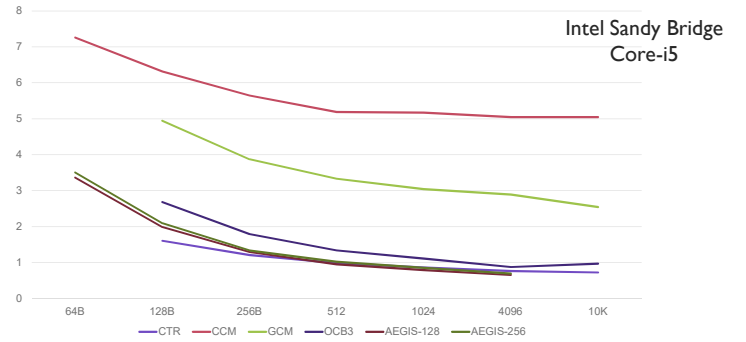
OCB2 has been broken at Crypto 2019 (bug in security proof) – but OCB3 is still ok

AEGIS: nonce-based Authenticated Encryption

- stream cipher using AES instruction
- 2x faster than AES-GCM: 0.287 cycles/byte
- multiple implementations available (including in Linux kernel)

AEGIS Performance:

0.287 cycles/byte on Skylake, 0.66 cycles/byte on Sandy Bridge



Lightweight cryptography competition (2015-2022)

https://csrc.nist.gov/projects/lightweight-cryptography

Authenticated Encryption with Associated Data (AEAD)

- AEAD and hashing for constraint environments
- AEAD for hardware environments

Status

Start: 2015

Feb. 2019: Round 1: regular submissions 56

Aug. 2019: Round 2: candidates left 32

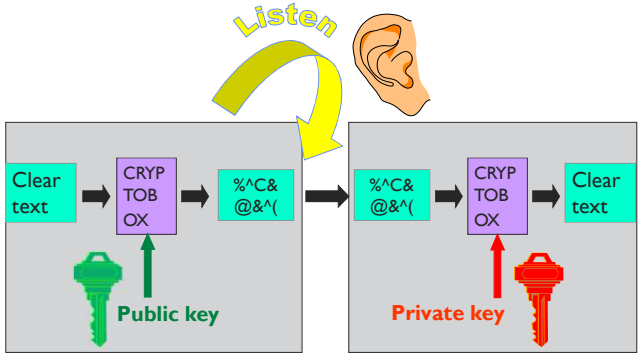
Mar. 2021: 10 finalists

ASCON, Elephant, GIFT-COFB, Grain128-AEAD, ISAP, Photon-Beetle, Romulus, Sparkle, TinyJambu, Xoodyak

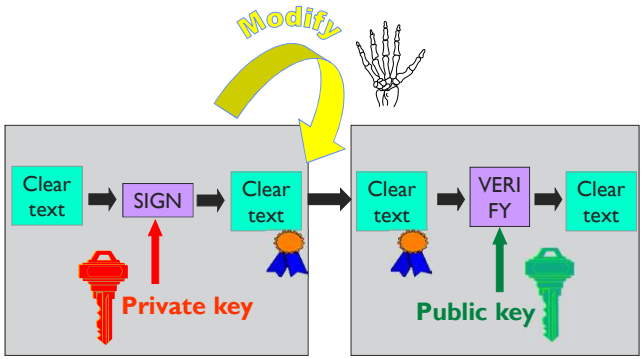
Outline

- Authenticated encryption
- Post-quantum cryptography
- Computing on Encrypted Data

Public key cryptology: encryption



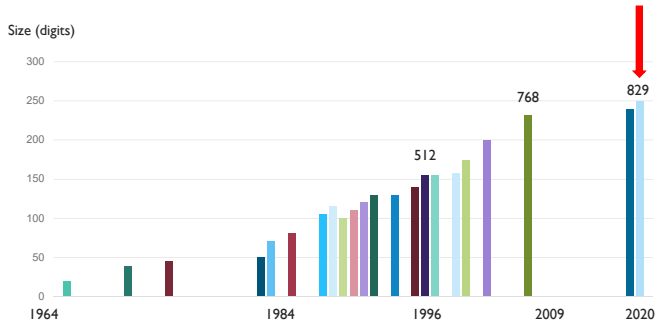
Public key cryptology: digital signature



Factorisation records

https://en.wikipedia.org/wiki/RSA_Factoring_Challenge

total computation time: 2700 core-years
(Intel Xeon Gold 6130 2.1 GHz)
- sieving: 2450 physical core-years
- matrix: 250 physical core-years



The advent of quantum computers

Yuri Manin 1980

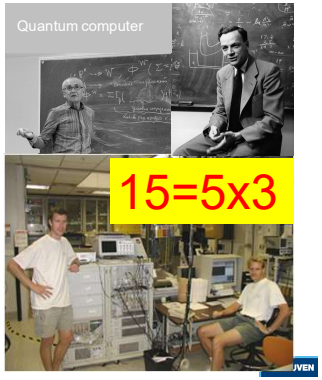
Richard Feynman 1981

Exponential parallelism

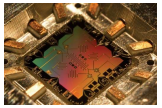
First trials in the 1990s

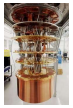
7-bit quantum computer in 2001

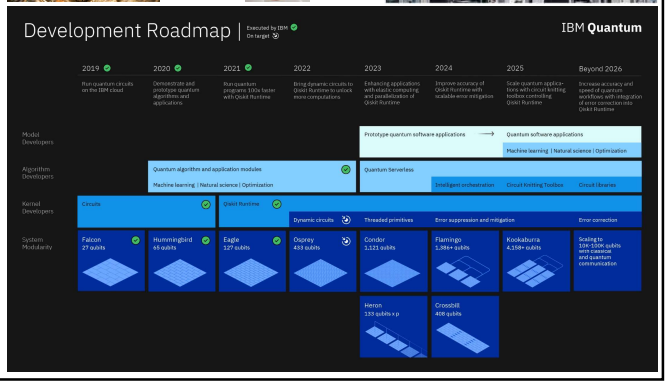
Jan. 2014: NSA has spent 85 M$ on research to build a quantum computer



	#qubits
IBM '17:	50
Intel '18:	49
Google '18:	72
Google '19:	54
IBM '19:	53
USTC China '20:	76
IBM 2021:	127

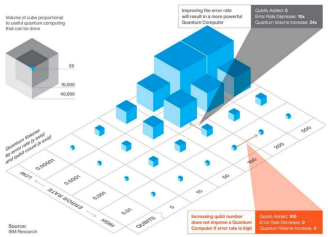


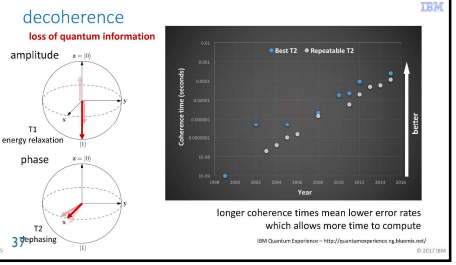




It's not only about qubits: error rate and decoherence

Quantum Volume





If a large quantum computer can be built

public-key cryptography algorithms have to be replaced [Shor'94]

RSA, Diffie-Hellman (including elliptic curves)

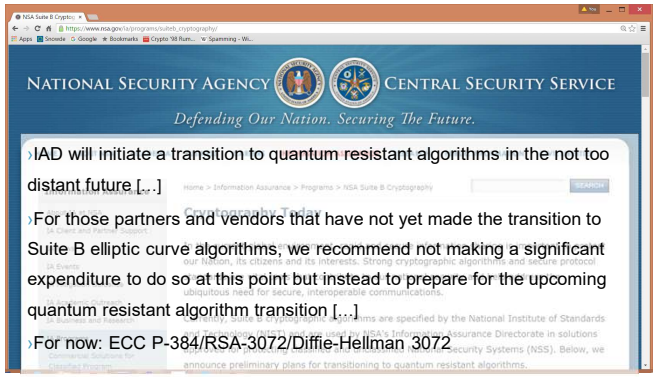
Breaking RSA-2048 requires 4096 ideal qubits or 20 million real qubits

symmetric crypto: key sizes: x2 [Grover'96]

but huge quantum devices needed

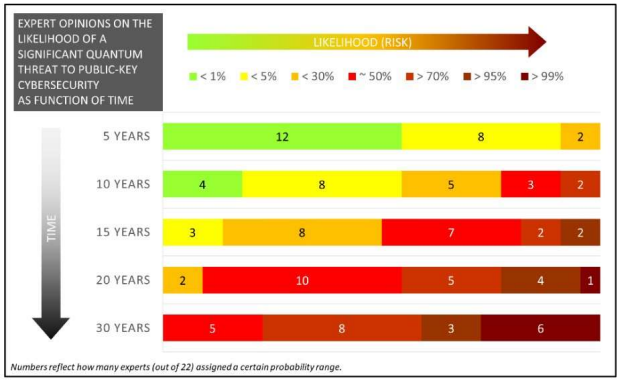
KU LEUVEN

What did the NSA say? August 19 2015: do not switch to Suite B



KU LEUVEN

What do the experts say today?

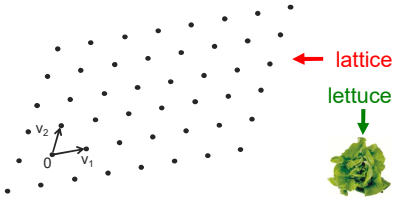


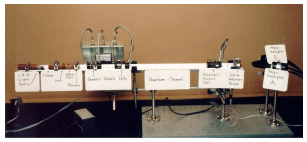
KU LEUVEN

Post-quantum Cryptography ≠ Quantum Key Distribution

Find new cryptographic algorithms that resist attacks on quantum computers

Use quantum physics to agree on secret keys





Original Quantum Cryptographic Apparatus built in 1989 transmitted information secretly over a distance of about 30 cm.

Sender's side produces very faint green light pulses of 4 different polarizations.

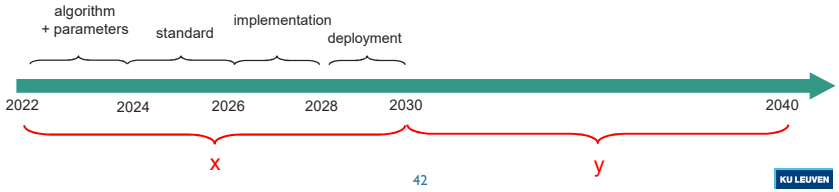
Quantum channel is an empty space about 30 cm long. There is no Eavesdropper, but if there were she would be detected.

Calcite prism separates polarizations. Photomultiplier tubes detect single photons.

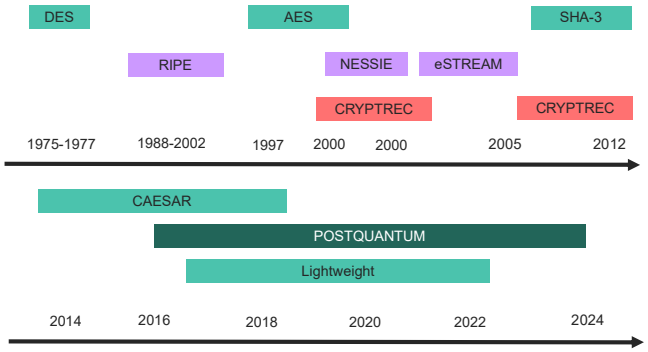
When to switch to post-quantum cryptography? [Mosca]

Q = #years until first large quantum computer
x = #years it takes to switch (3-12 years)
y = #years data needs to be confidential (10 years)

Need to start switching in the year 2022 + Q – x – y
e.g. Q = 18, x=8, y=10: today!



Open competitions



http://csrc.nist.gov/groups/ST/post-quantum-crypto/index.html

NIST Post-Quantum Standardization Effort

http://csrc.nist.gov/pqcrypto

Fall 2016		Formal call for proposals – NISTIR 8105
Nov 2017	69	Deadline for submissions (82 attempts)
Apr 2018		Workshop - Submitter's presentations
Jan 2019	26	Second round candidates announced – NISTIR 8240
Aug 2019		Second conference
July 2020	7	Third round finalist announced – NISTIR 8309
June 2021		Third conference
June 2022	?	Winners announced
2022-2023		Release draft standard
2024		Parameters chosen and standard published

Submissions to NIST Post-Quantum Competition

https://en.wikipedia.org/wiki/Post-Quantum_Cryptography_Standardization

13 rejected as incomplete; 25 broken in first year

	Signatures	Encryption/KEM	TOTAL
Lattice	4	24	28
Code	5	19	24
Multivariate	7	6	13
Hash	4	0	4
Other	3	10	13
TOTAL	23	59	82

45

KU LEUVEN

Submissions to NIST Post-Quantum Competition

https://en.wikipedia.org/wiki/Post-Quantum_Cryptography_Standardization

Reduction also by mergers

	Signatures	Encryption/KEM	TOTAL
Lattice	4/3	24/9	28/12
Code	5/0	19/7	24/7
Multivariate	7/4	6/0	13/4
Hash	4/1	0/0	4/1
Other	3/1	10/1	13/2
TOTAL	23/9	59/17	82/26

46

KU LEUVEN

Submissions to NIST Post-Quantum Competition

https://en.wikipedia.org/wiki/Post-Quantum_Cryptography_Standardization

7 finalists (1 of which broken)

	Signatures	Encryption/KEM	TOTAL
Lattice	4/3/2	24/9/3	28/12/5
Code	5/0/0	19/7/1	24/7/1
Multivariate	7/4/1	6/0/0	13/4/1
Hash	4/1/0	0/0/0	4/1/0
Other	3/1/0	10/1/0	13/2/0
TOTAL	23/9/3	59/17/4	82/26/7

47

KU LEUVEN

Submissions to NIST Post-Quantum Competition

https://en.wikipedia.org/wiki/Post-Quantum_Cryptography_Standardization

7 finalists (of which one broken) + 8 alternates

	Signatures	Encryption/KEM	TOTAL
Lattice	2+0	3+2	5+2
Code	0	1+2	1+2
Multivariate	~~1~~0+1	0	0+1
Hash	0+2	0	0+2
Isogeny	0	0+1	0+1
TOTAL	~~3~~2+3	4+5	6 *+8*

48

KU LEUVEN

Security levels

Level	Classical	
I	AES 128	2^{170}/MAXDEPTH quantum gates or 2^{143} classical gates
II	SHA3-256	2^{146} classical gates
III	AES192	2^{233}/MAXDEPTH quantum gates or 2^{207} classical gates
IV	SHA3-384	2^{210} classical gates
V	AES256	2^{298}/MAXDEPTH quantum gates or 2^{272} classical gates

Criticism: too vague
- circuit depth
- cost of memory
- type of quantum gates

Evaluation Criteria

Security
Security levels offered
Confidence in proofs
Attacks
Classical/quantum complexity

Performance
Size of parameters
Speed of Keygen Enc/Dec
Sign/Verify
Softw./hardw. benchmarks

Algorithm and implementation
IP issues
Decryption failures
Side channel resistance
Simplicity and clarity of docs
flexibility

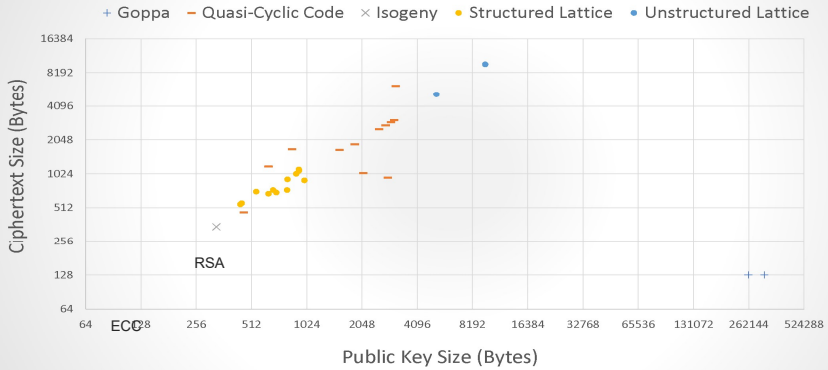
Other
Comments received
Academic papers published

Public-key Encryption: 9 KEMs (Key Encapsulation Mechanism)

Saber has best performance
NTRU has not decryption failures

- 5 Lattice based schemes:
  - **Finalists: Kyber (module LWE), SABER (module LWR), NTRU (1996)**
  - Alternates: FrodoKEM, NTRUprime
  - FrodoKEM is only scheme with unstructured lattice (larger keys)
- 3 Code-based schemes
  - **Finalist: Classic McEliece (1978) – large public keys but small ciphertexts**
  - Alternates: BIKE, HQC: structured codes thus smaller keys
- 1 Isogeny-based scheme
  - Alternate: SIKE: small key and ciphertext but very slow

Public Key vs Ciphertexts, Category 1



https://csrc.nist.gov/CSRC/media/Presentations/Round-2-of-the-NIST-PQC-Competition-What-was-NIST/images-media/pqcrypto-may2019-moody.pdf

6 Digital Signature schemes

- 2 Lattice based schemes:
  - **Finalists: Dilithium (Fiat-Shamir), Falcon (hash and sign)**
  - Alternates: FrodoKEM, NTRUprime
  - FrodoKEM is only scheme with unstructured lattice (larger keys)
- Hash-based signatures
  - Alternate: SPHINCS+ stateless
- Zero-Knowledge proof + block cipher
  - Alternate: Picnic
- Multivariate
  - **Finalist: Rainbow**
  - Alternate: GEMSS

Falcon has better performance

NIST update Spring 2022: 4th round and a new call for digital signatures

Overall goal: more diversified portfolio

Round 4
- Non-winners in round 3
- Duration: 18-24 months

Digital signatures: general purpose scheme not based on structured lattices
- January 2023: call for submissions
- Separate track from Round 4

Challenges

- Patents
- Larger keys/ciphertexts/signatures
- Most robust schemes have worse performance: hash-based signature and Classic McEliece
- Lattice based schemes
  - Good performance
  - Uncertainty about parameters for structured lattices
  - Decryption failure, floating point, noise sampling
- Side channel resistance: KEM in Fujisaki-Okamoto mode: FO-calyps [Azouaoui et al., Surviving the FO-CALYPS: Securing PQC Implementations in Practice, RWC'22]

Other standards

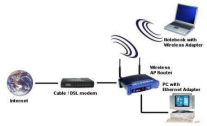
- IEEE P1363.3 (2008), X9.98: NTRU
- IETF: hash-based signatures
  - IETF RFC 8554 Leighton-Micali signatures (stateful)
  - IETF RFC 8391 XMSS eXtended Merkle (stateful)
- ISO/IEC JTC1/SC27: study period

Outline

- Authenticated encryption
- Post-quantum cryptography
- Computing on Encrypted Data

Changing role of cryptography

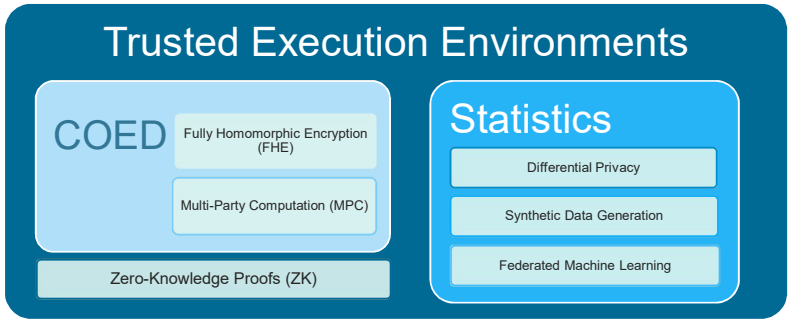
communications



storage

during computation

Computing on Encrypted Data (COED)



Computing on Encrypted Data

Overhead	Trusted Server	MPC (Multi Party Computation)	FHE (Fully Homomorphic Encryption
	ARM TrustZone Intel SGX AMD SEV	all parties engage in a protocol to compute the function securely	the parties encrypt their data, a server computes the function in the encrypted domain, a designated party gets the output
Computation	Fast	Relatively fast	Very very slow
Communication	Relatively low	Expensive	Relatively low
Applications	Yes	Growing range of options	Simple functions
Security	Need to trust hardware manufacturer (+ infrastructure)	Very high (can even be unconditional) (trust cryptographers)	High (trust cryptographers)

Conclusion

- Cryptography keeps changing
- Cryptographic agility is challenging
- Secure implementations

Bart Preneel, COSIC, KU Leuven and imec

ADDRESS: Kasteelpark Arenberg 10, 3000 Leuven
WEBSITE: homes.esat.kuleuven.be/~preneel/
EMAIL: Bart.Preneel@esat.kuleuven.be
TWITTER: @bpreneel1
TELEPHONE: +32 16 321148

Selected books on cryptology

A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997. The "bible" of applied cryptography. Thorough and complete reference work but slightly outdated– not suited as a first text book. http://www.cacr.math.uwaterloo.ca/hac

D. Boneh, V. Shoup, A Graduate Course in Applied Cryptography, https://toc.cryptobook.us/ Draft. Rather advanced course with interesting applications.

N. Smart, *Cryptography Made Simple*, Springer, 2015. Solid and up to date but on the mathematical side.

D. Stinson, M. Peterson, *Cryptography: Theory and Practice*, CRC Press, 4th Ed., 2018. Solid introduction, but only for the mathematically inclined.

J. Katz and Y. Lindell, *Introduction to Modern Cryptography,* Chapman & Hall, 2014. Rigorous and theoretical approach.