Error Correction Codes, from Communication to Cryptography... Two sides of one *chip*

Dr. Cyrille CHAVET - Dr. Bertrand LE GAL

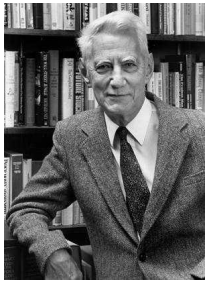
Pr. Philippe COUSSY, Mael TOURRES, Syed FAHIMUDDIN ALAVI

June 12, 2024

Agenda

1. **A brief history of ECC & Cryptography**
2. **Face of the chip - Error Correction Codes**
3. **Tails side of chip - Cryptographic algorithms**
4. **Conclusion**

Once upon a time...

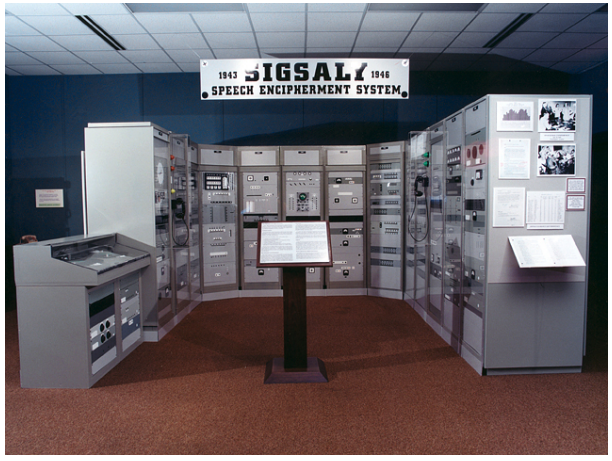


- Efforts to create a secure voice system had existed since the 1920s
- During World War II, C. Shannon works on SIGSALY, the first Secure Digital Voice Communications
- Mathematical definition of information and encrypted transmission over a noisy channel

- Shannon's paper "*A Mathematical Theory of Communication*[1]" is considered as the founding work of information theory
- Error Correction Codes are an integral part of encryption

[1] *Shannon, C. E. (1948). A mathematical theory of communication, The Bell System Technical Journal, 27(3), 379-423*; and the second part, *in The Bell System Technical Journal, 27(4), 623-656.*

Once upon a time...



SIGSALY[2] was a digital speech encryption system, developed by Bell Telephone Laboratories

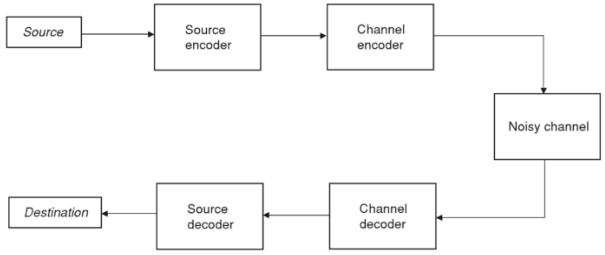
[2] Source: National Cryptologic Museum

In our everyday lives

- The world's most important asset is information
- Protecting information is crucial to ensure a trusted global economy (E-commerce, online banking, social networking or emailing, online medical results, mobile phone communications, stock exchange...)
- These constraints generate two huge challenges:
  - Ensuring the same quality of information for any communication channel ⇒ **Error Correction Codes**
  - Ensuring the security of information for any communication channel ⇒ **Cryptography**
  - For a "**reasonable**" cost

Error Correction Codes Principles

- Error detection/correction on digital communications channels
- Two types of encoding methods
  - **Block encoding** - Each codeword is generated from one block of k message symbols
  - **Convolutional encoding** - Each codeword is generated from several consecutive message blocks



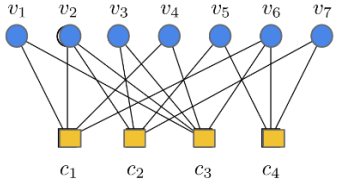
Some typical Error Correction Codes

- Hamming
- Reed-Salomon
- Goppa
- McEliecce
- LDPC
- Turbo-Codes
- NB-LDPC
- Polar codes
- ...

Some typical Error Correction Codes

- Hamming
- Reed-Salomon
- Goppa
- McEliecce
- **LDPC**
- Turbo-Codes
- NB-LDPC
- Polar codes
- ...

$$\mathbf{H} = \begin{array}{c} \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array} \begin{matrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{matrix}$$



Some typical Error Correction Codes

- Hamming
- Reed-Salomon
- Goppa
- McEliecce
- **LDPC**[a]
- Turbo-Codes
- NB-LDPC
- Polar codes
- ...

SW LDPC - Intel XEON GOLD 6148

LDPC code	#cores	favg	Throughput	Latency	Power
(16384, 4096)	1	3403 MHz	0.48 Gbps	34 μs	169 W
(16384, 4096)	40	2194 MHz	11.25 Gbps	60 μs	343 W
(64800, 29160)	1	3484 MHz	0.32 Gbps	201 μs	173 W
(64800, 29160)	40	2194 MHz	7.60 Gbps	345 μs	370 W
(22528, 6144)	1	3485 MHz	0.35 Gbps	64 μs	173 W
(22528, 6144)	40	2194 MHz	7.55 Gbps	122 μs	345 W

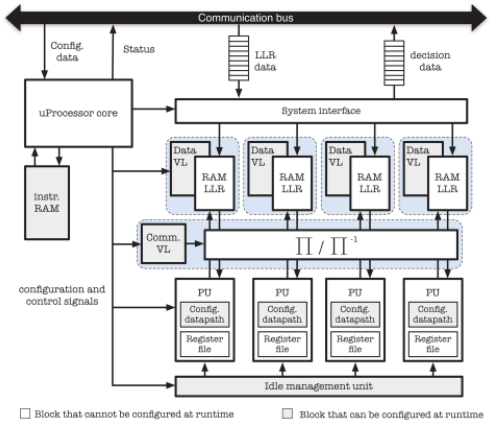
HW LDPC - Xilinx Ultrascale+ XCZU9EG-3FFVB1156E

LDPC code	#cores	FPGA usage	fmax	Throughput	Latency	Power
(16384, 4096)	1	17%	274 MHz	2.34 Gbps	7.7 μs	3.29 W
(16384, 4096)	5	78%	274 MHz	11.65 Gbps	7.7 μs	13.80 W
(64800, 29160)	1	22%	271 MHz	1.36 Gbps	47.5 μs	4.74 W
(64800, 29160)	3	68%	270 MHz	4.08 Gbps	47.7 μs	12.92 W
(22528, 6144)	1	31%	273 MHz	4.02 Gbps	6.1 μs	3.63 W
(22528, 6144)	2	61%	273 MHz	8.04 Gbps	6.1 μs	6.62 W

[a] V. Pignoly *et al.*, "Fair comparison of hardware and software LDPC decoder implementations for SDR space links", Int. Conf. ECS, 2020

Some typical Error Correction Codes

- Hamming
- Reed-Salomon
- Goppa
- McEliecce
- **LDPC**[ab]
- Turbo-Codes
- NB-LDPC
- Polar codes
- ...



[a] V. Pignoly *et al.*, "Fair comparison of hardware and software LDPC decoder implementations for SDR space links", Int. Conf. ECS, 2020

[b] B. Le Gal and C.Jégo, "GPU-Like On-Chip System for Decoding LDPC Codes",ACM TECS, 2014

Some typical Error Correction Codes

- Hamming
- Reed-Salomon
- Goppa
- McEliecce
- **LDPC**[ab]
- Turbo-Codes
- NB-LDPC
- Polar codes
- ...

Performance Comparison of the Proposed GPU-Like Decoder Vs GPU-Based Decoders

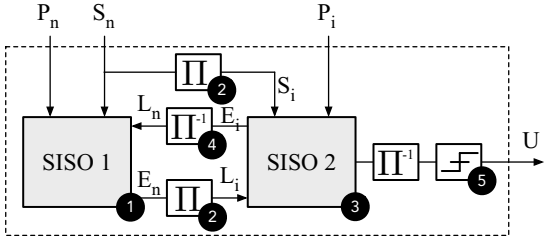
Code	It.	Work	Device	# PU	Freq.	*Mbps*	*T NDC*
2304 × 1152	10	[Wang et al. 2011a]	GTX-470	448	1215	48.74	0.089
	5	our work	GPU-like	64	100	307.4	47.97
4000 × 2000	20	[Chang et al. 2011]	Tesla C1060	240	1300	1.20	0.004
	10	our work	GPU-like	64	100	178.1	27.83
4896 × 2448	50	[Falcao et al. 2011b]	8800-GTX	128	1350	7.70	0.045
	25	our work	GPU-like	64	100	73.94	11.55
8000 × 4000	20	[Chang et al. 2011]	Tesla C1060	240	1300	2.27	0.007
	10	our work	GPU-like	64	100	181.1	28.29
8000 × 4000	50	[Falcao et al. 2011b]	8800-GTX	128	1350	10.10	0.058
	25	our work	GPU-like	64	100	73.71	11.51
20k × 10k	50	[Falcao et al. 2011b]	8800-GTX	128	1350	9.50	0.055
	25	our work	GPU-like	64	100	75.71	11.83
64k × 32k	20	[Falcao et al. 2011a]	Fermi C2050	448	1150	107.8	0.209
	10	our work	GPU-like	64	100	159.8	24.97

[a] V. Pignoly *et al.*, "Fair comparison of hardware and software LDPC decoder implementations for SDR space links", Int. Conf. ECS, 2020

[b] B. Le Gal and C.Jégo, "GPU-Like On-Chip System for Decoding LDPC Codes",ACM TECS, 2014

Some typical Error Correction Codes

- Hamming
- Reed-Salomon
- Goppa
- McEliecce
- LDPC
- **Turbo-Codes**
- NB-LDPC
- Polar codes
- ...



Some typical Error Correction Codes

- Hamming
- Reed-Salomon
- Goppa
- McEliecce
- LDPC
- **Turbo-Codes**[a]
- NB-LDPC
- Polar codes
- ...

[a] B. Le Gal and C. Jégo, "Low-latency and high-throughput software turbo decoders on multi-core architectures", Annals of Telecommunications, 2019

	Platform	Cores or SM	Freq Ghz	Γ Mbps	Δ μs	NThr. Mbps	TNDC	E_d nJ
[22]	GTX 550Ti	6	1.80	85	72*	47	1.5	227
[23]	GTX 580	16	1.54	4	1660	1	0.0	10991
[24]	GTX 480	15	1.40	123	50*	35	1.1	339
[25]	GTX 680	8	1.01	37	2657	27	0.1	878
[26]	GTX 580	16	1.54	107	230*	22	0.7	458
[25]	i7-3770K	4	3.50	76	323	33	4.1	168
[27]	i7-4960HQ	4	3.20	143	2731	67	4.2	41
[27]	2.E5-2680v3	24	2.50	716	3293	67	4.2	41
[27]	i7-4960HQ	4	3.20	51	7693	24	1.5	114
[27]	2.E5-2680v3	24	2.50	237	9971	24	1.5	169
[27]	i7-4960HQ	4	3.20	51	7693	24	0.7	114
[27]	2.E5-2680v3	24	2.50	457	10312	46	1.4	87
LeG	i7-4960HQ	4	3.20	238	103	112	7.0	25
LeG	2.E5-2680v3	24	2.50	908	162	91	5.7	44
LeG	i7-4960HQ	4	3.20	225	108	105	6.6	26
LeG	2.E5-2680v3	24	2.50	894	165	89	5.6	45
LeG	i7-4960HQ	4	3.20	466	52	218	6.8	13
LeG	2.E5-2680v3	24	2.50	1735	84	174	5.4	23

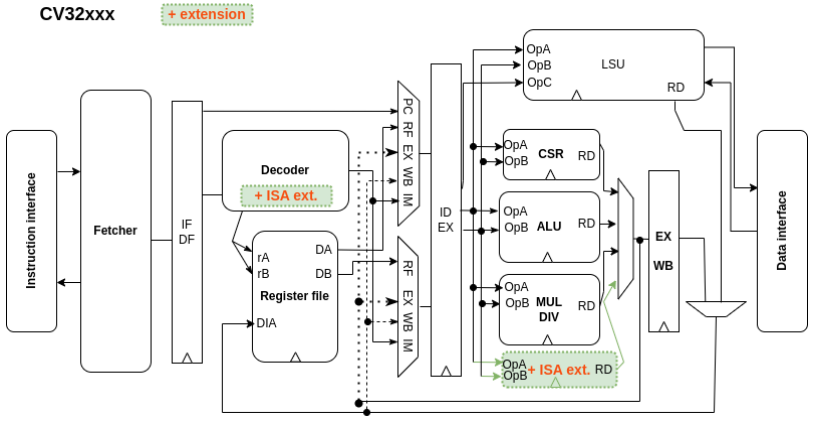
* Authors do not include CPU to/from GPU data transfers
NThr. = (Thr. × Iters) / (Freq. × Cores)
TNDC = (Thr. × Iters) / (Cores × Freq. × SIMD)
E_d = [TDP/(Thr. × Iters)] × 10^3

PhD thesis of Mael Tourres[3](2019-2024)

- Director: Pr. P. Coussy
- Advisors: C. Chavet and B. Le Gal
- Propose an approach to generate processor ISA extension dedicated to ECC
- Selected ECC for experiences : *LDPC*, *Turbo-Codes*, *NB-LDPC* and *Polar Codes*
- Objectives
  - Taking advantage of both worlds ASIC for speed and consumption, and CPU/GPU for adaptability
  - Taking advantage of data parallelism, to further enhance performance
  - Preserve the clock frequency
  - Do not degrade final chip area and power consumption

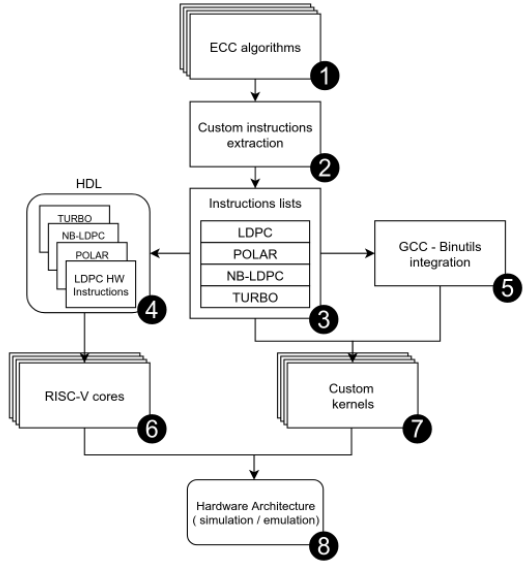
[3] M. Tourres *et al.*, "Extended RISC-V hardware architecture for future digital communication systems", 2021 IEEE 4th 5G World Forum (5GWF)

Architectural target



- Selected RISC-V cores: *PicoRV32*, *RISCY*, *IBEX* and *SCR1* *(and CVA6 from OHG)*

New Instruction Design Flow



Last Experimental Results *to be published*[4]

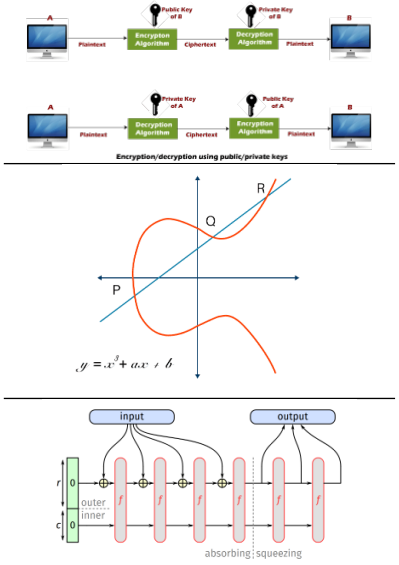
	Mnémonique	Mode SISD 2R	Mode SISD 3R	Mode SIMD inter-trames 2R	Mode SIMD inter-trames 3R	Mode SIMD intra-trame 2R	Mode SIMD intra-trame 3R
Instr. à 2 reg.	i8_max_pi8	✓	✓	✓	✓	✓	✓
	i8_scale_pi8	✓	✓	✓	✓		
	i8_sign_pi8			✓	✓		
	i8_add_pi8			✓	✓	✓	✓
	i8_sub_pi8			✓	✓	✓	✓
	i8_addsrl_pi8			✓	✓		
	i8_srl_pi8			✓	✓		
	i8_shuffle_pi8					✓	✓
	i8_add_div2_pi8					✓	✓
	i8_sub_div2_pi8					✓	✓
	i8_sat_sub_pi8					✓	✓
	i8_scale_add_pi8					✓	✓
Instr. à 3 reg.	i8_accumax_pi8		✓		✓		✓
	i8_maxpm_pi8		✓		✓		
	i8_accupp_pi8		✓		✓		✓
	i8_accump_pi8		✓		✓		✓
Total (2R|3R)		2	6(2|4)	7	11(7|4)	8	11(8|3)

inter-trames			LDPC OMS	LDPC SC	Polaire F-SC	LDPC-NB MS	Turbo MLM
int8	❶ Baseline	cycles	95955	454637	239292	1715625	1585822
int8_t×4 (32 bits)	❷ ISA 2R	cycles	61443	373854	215509	1292853	1051247
	❸ ISA 3R	cycles	54027	292678	167095	969990	804471
	Gain (❶→❸)	cycles	41928	161959	72197	745635	781351
	Gain (❶→❸)	(%)	43.7%	35.6%	30.2%	43.5%	49.2%
	Gain (❷→❸)	cycles	7416	81176	48414	322863	246776
	Gain (❷→❸)	(%)	12.0%	21.7%	22.4%	25.0%	23.4%
	Débit ❶	Kbits/s	141	900	1711	7	390
	Débit ❷	Kbits/s	885	4382	7602	39	2356
	Débit ❸	Kbits/s	1006	5598	9805	52	3078
int8_t×8 (64 bits)	❹ ISA 2R	cycles	57898	388220	243988	1359186	1207724
	❺ ISA 3R	cycles	52546	335032	213560	1015731	1035964
	Gain (❹→❺)	cycles	43409	119605	25732	699894	549858
	Gain (❹→❺)	(%)	45.2%	26.3%	10.8%	40.8%	34.6%
	Gain (❷→❺)	cycles	5352	53188	30428	343455	171760
	Gain (❷→❺)	(%)	9.2%	13.7%	12.4%	25.7%	14.2%
	Débit ❹	Kbits/s	141	900	1711	7	390
	Débit ❺	Kbits/s	1879	8440	13430	75	4101
	Débit ❺	Kbits/s	2070	9780	15343	100	4781
Accélération							
	❶ → ❸	32 bits	7.1×	6.2×	5.7×	7.1×	7.9×
	❶ → ❸	64 bits	14.6×	10.9×	9.0×	13.5×	12.2×
	❸ → ❸	32 → 64 bits	2.1×	1.7×	1.6×	1.9×	1.6×

[4] M. Tourres *et al.*, "Specialized Scalar and SIMD instructions for Error Correction Codes Decoding on RISC-V processors", journal paper in progress

Classical Cyphers

- Symmetric cyphering: DES, **AES**
- Asymmetric cyphering: PGP, **SSL**, **RSA**
- Elliptic Curve cyphering: **ECDSA**
- Signature: MD5/SHA1, **SHA2**, **SHA3**-*Keccak*



Standardization of RISC-V Cryptographic Extensions[5]

Scalar Cryptography

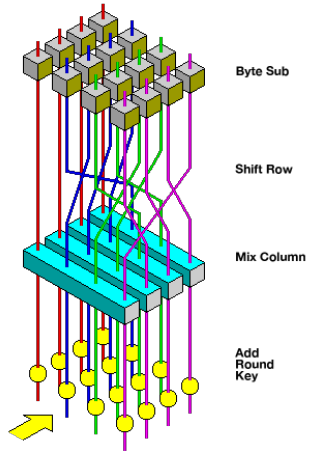
- Bit manipulation, Data independent execution, AES enccryption/decryption, SHA2...
- Public review ended in $Oct.17^{th}, 2021$
- **Ratified**

Vector Cryptography

- Same algorithms, but for element that are group of scalars
- Public review ended last year
- **Ratified**

[5] see https://github.com/riscv/riscv-crypto

Example of an AES extension on a RISC-V[6]



Source: J.Savard, wikipedia

- Hardware assisted T-table approach for 32 and 64-bit architectures
- 32-bit architectures
  - Byte-wise round instructions
- Results on SCARV and Rocket
  - Limited area increase - x1,03 vs SCARV and 1,001 vs Rocket
  - SW memory footprint divided by a factor 2,8
  - Speedup upto x3,6 on SCARV and x2,5 on Rocket

[6] B.Marshall *et al.*, "The design of scalar AES Instruction Set Extensions for RISC-V", https://doi.org/10.46586/tches.v2021.i1.109-136

Hypothetical (?) weaknesses

- Mathematical problem that are really complex to solve (prime factorization, log computation...)
- The computing power required to crack them is titanic
- However, the literature is full of side-channel attacks... and ways of countering them
  - CPA, DPA, clock glitching...
  - Masking[7], randomization...

All this could collapse the day a real quantum computer arrives in the laboratories

[7] F. Lozachmeur and A. Tisserand, "A RISC-V Instruction Set Extension for Flexible Hardware/Software Protection of Cryptosystems Masked at High Orders", 66th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS 2023)

A solution - Post-Quantum Cryptography

- The advent of post-quantum encryption has seen the return of ECC to favour
  - McEliecce
  - NTRU MDPC
  - Bike
  - **Crystal Kyber/Dilithium**
- Expensive and complex algorithms
- For small system, a hardware accelerator could be a good option

Recent works

- Many studies explore these aspects
  - Processor that are able to change between classical and post-quantum cryptography
  - Dedicated PQC coprocessors design
  - Definition of PQC extension of ISA *(cf. PhD thesis of M. Tourres)*

PhD Thesis of Rémy Fumeron (2019-xx)

- Founded by PEC Cyber
- Director: Pr. Philippe Coussy
- Advisors: C. Chavet and K. Martin
- Subject: *Exploring an agile CPU that could handle both generic computation and PQC algorithm - Study of ECC code from the NIST on the Pulp platform*
- Research abandoned by the PhD student at the end of 2020

PhD Thesis

- Founded by PEPR - ARSENNE
- Thesis director: Dr Arnaud Tisserand (DR CNRS)
- Subject: *Exploring the design of an architecture that could handle both classical and PQC*

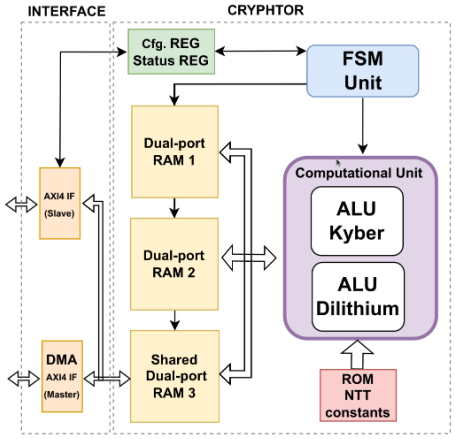
Master Thesis of Ivan SARNO

CRYPHTOR[8] (**CRY**stals **P**olynomial **H**W accelera**TOR**)

- This work proposes a coprocessor to handle Crystal PQC algorithms
- Integration on a RISC-V target

Collaboration with TIMA

A PhD thesis founded by CEA will start



[8] S. Di Matteo *et al.*, "CRYPHTOR: A Memory-Unified NTT-Based Hardware Accelerator for Post-Quantum CRYSTALS Algorithms", IEEE Access, 2024, DOI:10.1109/ACCESS.2024.3367109

Master Thesis of Syed FAHIMUDDIN ALAVI

- Supervisor: Dr. C. Chavet
- Exploring RISCV ISA extension for Post-Quantum Cryptography
- Target architecture: PicoRV32
- Test case: AES for classical cryptography and Crystal and McEliecce for PQC
- The idea is to show that it is possible to extract some complex parts of a PQC algorithm, and to replace it by some dedicated new RISCV instruction

Master Thesis of Syed FAHIMUDDIN ALAVI

- Current results
  - For AES, the PicoRV32 with custom instruction is upto twice as faster compared to the reference
  - For CRYSTAL Kyber 1024:
    - $\sim$ 8.5% and $\sim$ 18.5% of latency improvement in PQC encryption and decryption resp.
    - FMAX legacy: 58,27MHz, FMAX custom ext.: 82,85 MHz
    - Area overcost: 6,5% for our enriched PicoRV32
- Paper is coming...

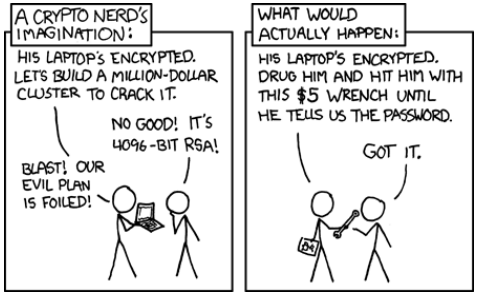
Conclusion

- The worlds of error-correcting codes and cryptography are part and parcel of each other
- But, currently 'simple' mathematical problems (compared to ECC) made it possible to guarantee, a level of security that make encryption algorithms virtually unbreakable
- However, it is possible that in the medium/long term the emergence of quantum machines will disrupt all this
- As part of the NIST competition, ECC have made a comeback in the field of cryptography (McElliecce, NTRU MDPC, Bike...)
- As it was the case when LDPCs were first discovered, these algorithms offer excellent performance, but at prohibitive cost
- To solve these problems, the solution may once again come from computer engineering...

Conclusion

- The worlds of error-correcting codes and cryptography are part and parcel of each other
- But, currently 'simple' mathematical problems (compared to ECC) made it possible to guarantee, a level of security that make encryption algorithms virtually unbreakable
- However, it is possible that in the medium/long term the emergence of quantum machines will disrupt all this
- As part of the NIST competition, ECC have made a comeback in the field of cryptography (McElliecce, NTRU MDPC, Bike...)
- As it was the case when LDPCs were first discovered, these algorithms offer excellent performance, but at prohibitive cost
- To solve these problems, the solution may once again come from computer engineering...

Thanks



https://xkcd.com/538 - Security