

# On FHEMPCZK-friendly symmetric crypto

---

Christian Rechberger, TU Graz

June 14, 2023

# A Zoo of FHEMPCZK-friendly concretely-efficient symmetric crypto: How many designs?

2013: -

2014: -

2015: 1

2016: 4

2017: -

2018: 3

2019: 5

2020: 5

2021: 8

2022: 10

2023: 4 until April

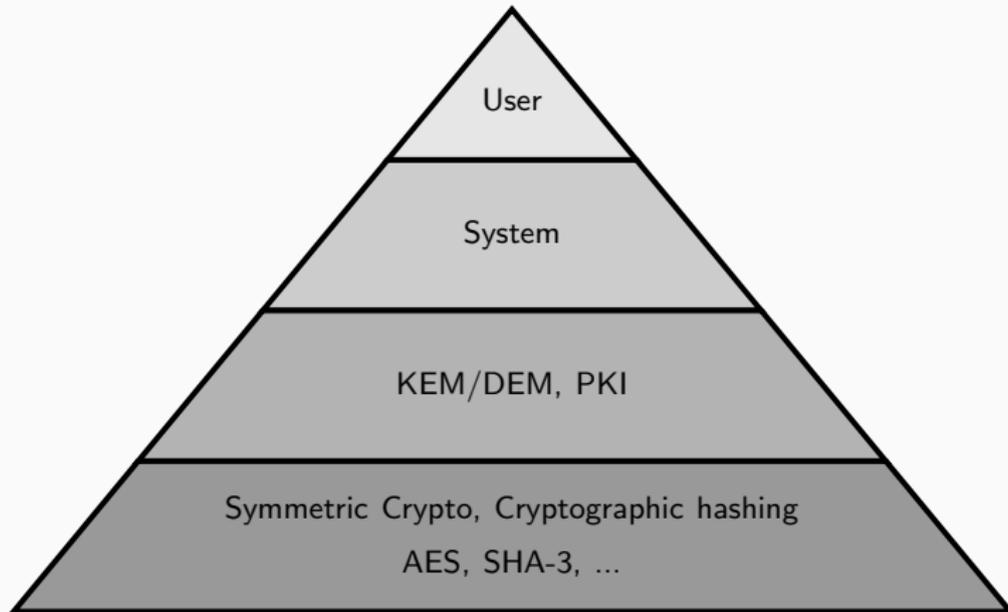
source: mostly IACR eprint, plus selection from IEEE Access, ToSC, arxiv

How did we get here?

Efficiently provide confidentiality, authenticity, integrity

- **until 1980s**: dedicated machines, hardware implementing DES, LFSR-based approaches
- **since 1990s**: software implementations become more relevant in addition to hardware, see e.g. AES
- **since 2010s**: another boost for software-environments due to virtualization

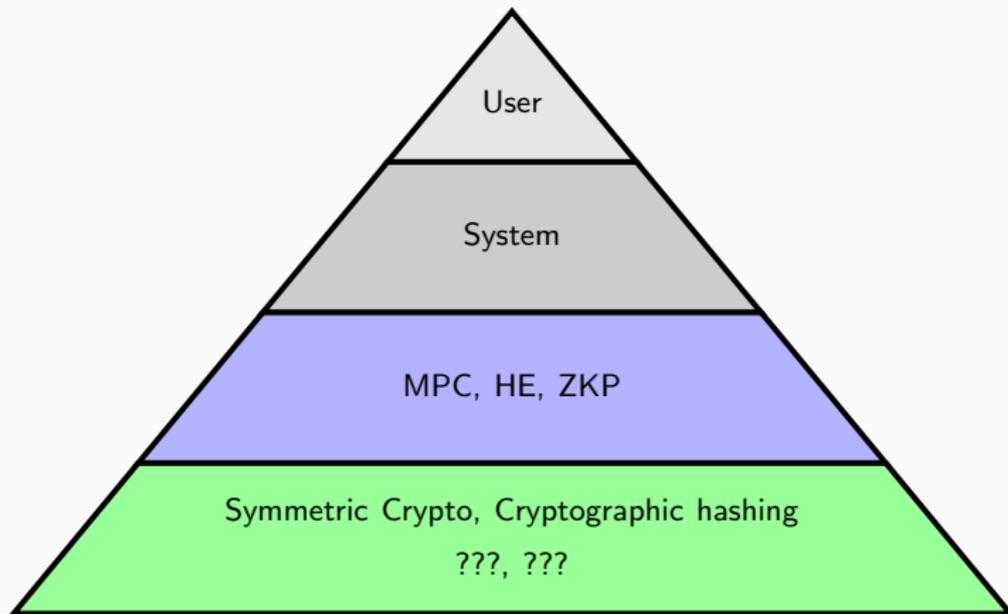
# Role of symmetric-key crypto and hashing in systems



# New cryptographic functionalities are new applications of symmetric cryptography

- **FHE:** Reducing ciphertext expansion, OPRFs, ...
- **MPC:** Distributed databases, private set intersection, data analytics, but also new public-key signature schemes
- **ZKP:** Use-cases of zero-knowledge proofs:
  - Set Membership Proofs (“I know a private key of one of the public keys of this Merkle tree”)
  - Data Commitments (“Here is the Merkle tree of the execution trace of my program, I can open it at any point”).

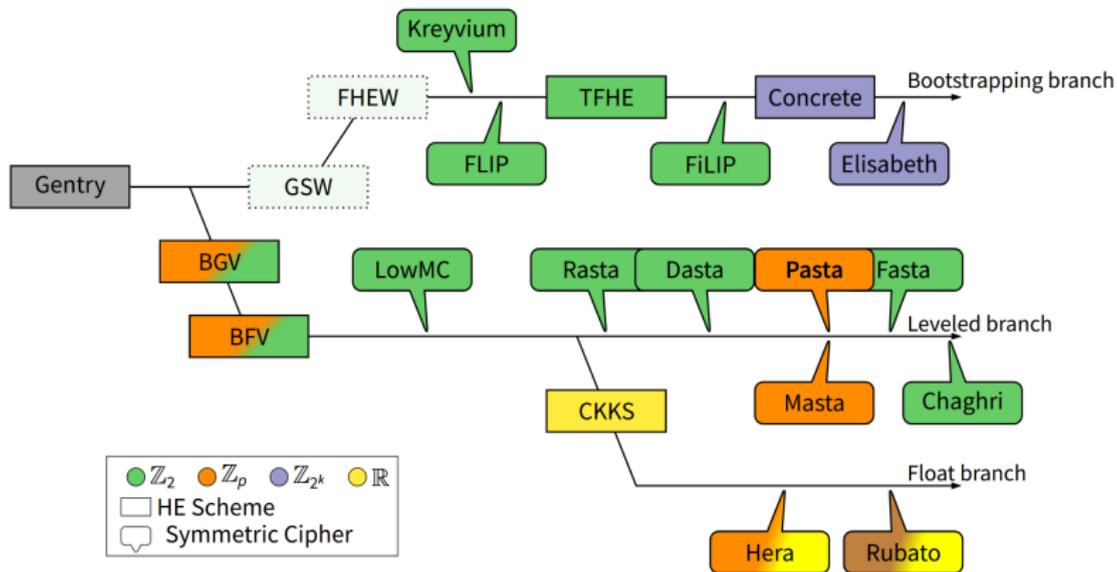
# Role of symmetric-key crypto and hashing in systems



# Transitions of use-cases in (symmetric) cryptography

- in the 1980s and 90s, there was a transition from hardware to software.
  - Hardware grew, but software grew much more.
- since the mid 2010s: we seem to be in a transition phase **from direct implementations to indirect implementations** within protocols aiming for "high functionality cryptography"
  - direct hardware and software implementations of course remain relevant, but the area of indirect implementations is growing fast.
  - new "virtual machines", new "metrics", co-developments of symmetric crypto with "higher/more functional" crypto layers

# A Zoo of Ciphers for Hybrid Homomorphic Encryption, a.k.a. Transciphering



# The ZK-friendly Hash Function Zoo

## Type 1

*"low degree only"*

- Low-degree

$$y = x^d$$

- Fast in Plain
- Many rounds
- Often more constraints
- MiMC(16),  
GMiMC(19),  
POSEIDON(19),  
NEPTUNE (21),  
Poseidon2 (23)

# The ZK-friendly Hash Function Zoo

## Type 1

*"low degree only"*

- Low-degree

$$y = x^d$$

- **Fast in Plain**
- **Many rounds**
- **Often more constraints**
- MiMC(16),  
GMiMC(19),  
POSEIDON(19),  
NEPTUNE (21),  
Poseidon2 (23)

## Type 2

*"non-procedural", "fluid"*

- Low-degree  
equivalence

$$y = x^{1/d} \Rightarrow x = y^d$$

- **Slow in Plain**
- **Fewer rounds**
- **Fewer constraints**
- Friday(18), Vision  
(19), *Rescue*(19),  
Grendel(21),  
GRIFFIN (22),  
ANEMOI (22),  
Arion(23)

# The ZK-friendly Hash Function Zoo

## Type 1

*"low degree only"*

- Low-degree

$$y = x^d$$

- **Fast in Plain**
- **Many rounds**
- **Often more constraints**
- MiMC(16),  
GMiMC(19),  
POSEIDON(19),  
NEPTUNE (21),  
Poseidon2 (23)

## Type 2

*"non-procedural", "fluid"*

- Low-degree  
equivalence

$$y = x^{1/d} \Rightarrow x = y^d$$

- **Slow in Plain**
- **Fewer rounds**
- **Fewer constraints**
- Friday(18), Vision  
(19), *Rescue*(19),  
Grendel(21),  
GRIFFIN (22),  
ANEMOI (22),  
Arion(23)

## Type 3

*"lookups"*

- Lookup tables

$$y = T[x]$$

- **Very fast in Plain**
- **Even fewer rounds**
- **Constraints depend on proof system**
- Reinforced  
Concrete (21),  
Tip5 (23), Tip4  
(23),  $RC_p$ (23)

# The MPC/Sharing-friendly Symmetric Crypto Zoo

2015: LowMC

2016: MiMC, LegendrePRF

2018: CryptoDarkMatter

2019: GMiMC

2020: HadesMiMC

2021: Ciminion, "CryptoDarkMatter++"

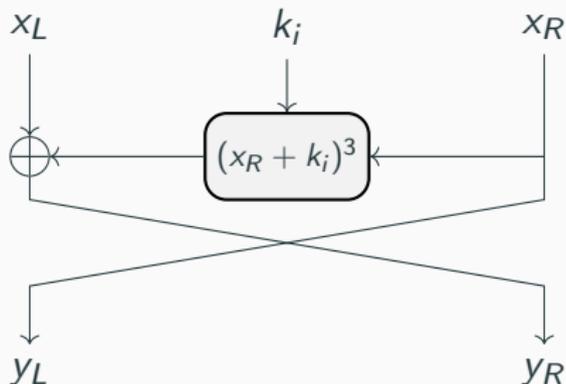
2022: Rain, AIM

2023: Hydra

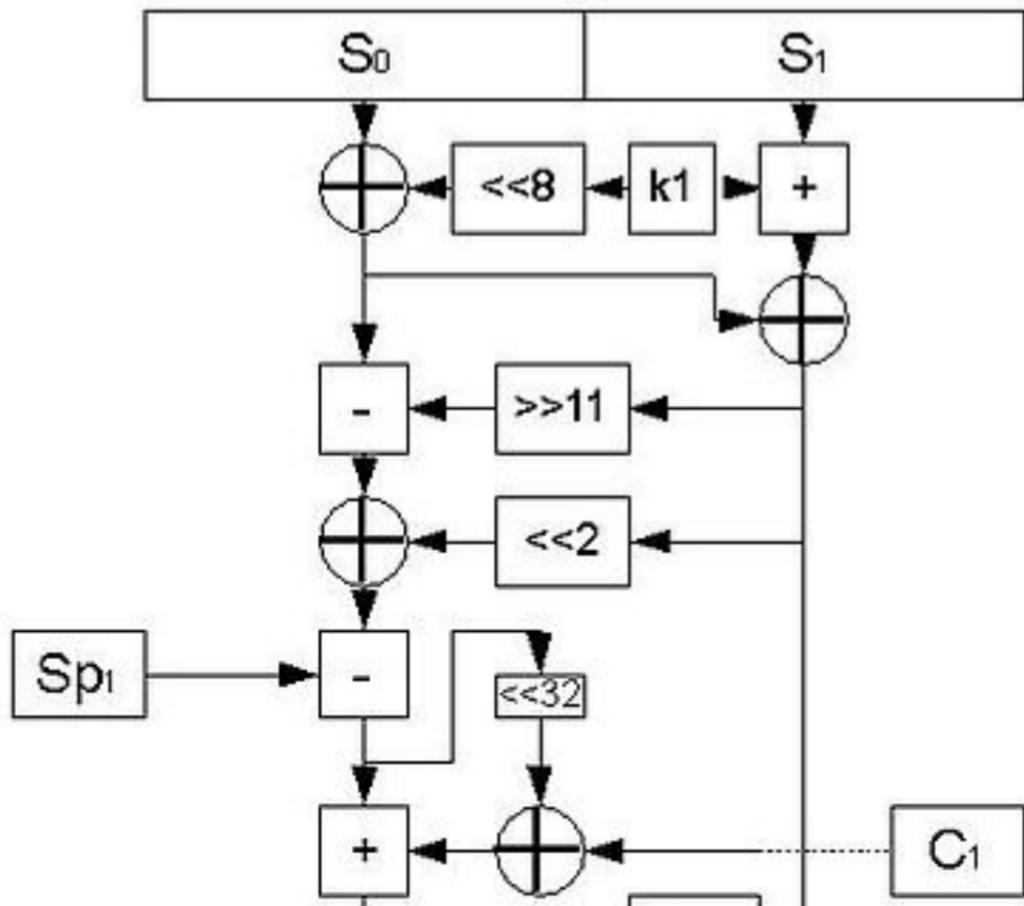
- Hybrid HE Use-Case:
  - Extensive benchmarks in different HE libraries including use-cases
  - 16 implementations (various ciphers for various HE libraries), before the count was 1.
- MPC Use-Cases:
  - Implementations of MiMC, GMiMC, HadesMiMC, Rescue, Ciminion, Hydra
  - More elaborate framework allowing for various libraries, access structures, still to come
- Zero-Knowledge Use-Cases:
  - Zoo of plain implementations (8)
  - Proof knowledge of preimages of hash functions (6)
  - Proof membership witness in Merkle tree accumulators (6)

## Prior art for (Feistel) MiMC

- *PURE* cipher [JK97] based on the  $\mathcal{KN}$  Feistel cipher [NK95]



## More prior art, for F(p) ciphers (1/2)



## More prior art, for $F(p)$ ciphers (2/2)

Richard Schroepel: "The Hasty Pudding Cipher", submission to the NIST AES Competition, 1998.

First(?)  $F(p)$  cipher.

First tweakable block cipher

Flexible parameterization (blocksize, keysize), maybe a first too?



Ok. Where do we go from here?

## On the "stability" of symmetric crypto and hashing

- MPC-friendly: Seems the most stable. Focus cryptanalysis efforts in standardization process/competition?
- HE-friendly: 4-5 underlying HE schemes are under standardization at ISO. Most, but not all schemes have a matching transciphering proposal.
- ZK-friendly: Most dynamic development at the moment, almost most immediate use in industry.

In general, more cryptanalysis is definitely useful and needed.

## Underexplored directions?

- MPC-friendly hashing? Brought up by Luis Brandao in recent NIST call.
- FHE-friendly PRFs.
- Hardware-friendly Sharing-friendly  $F(p)$  ciphers.  
Also relevant for cheap side-channel countermeasures.
  - Mathias Oberhuber(2021): MiMC+ECC synergies in e.g. hash+sign HW implementations. Both use same-sized multiplier in  $GF(2^n)$  or  $GF(p)$ .
  - FX Standaert et al. (2023): AES-like  $F(p)$  ciphers

## Classes of open problems

- How far can we go with signature schemes based symmetric crypto *only*? Signature size, computation effort?
- How far can we go with reducing computational overhead of hybrid homomorphic encryption?
- Holy grail in ZK-friendly hash function design: *Simultaneously* good performance in both plain and ZK
- Cryptanalysis of various new schemes in this domain

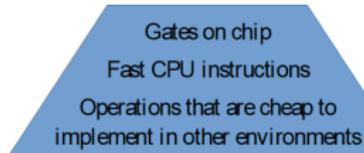
# Thoughts on "Theory" vs. "Practice"

- Provable Security?
  - Modes of operation: do proofs carry over from  $F_2$  to  $F_p$ ?
  - SPN vs. Partial-SPN: First positive results by Guo, Standaert, Wang, Wang, Yu (FSE 22)
    - Stronger model, like indifferenciability?
- "Asymptotic analysis" / "asymptotic designs".
  - Input: blocksize, security level
  - Output: concrete design with security claim
  - Some designs allow for it, e.g. HPC, LowMC, MiMC, Poseidon, ...
  - Pros: Flexibility.
  - Cons: Less focused cryptanalysis.

# Thoughts on "Theory" vs. "Practice": A vision



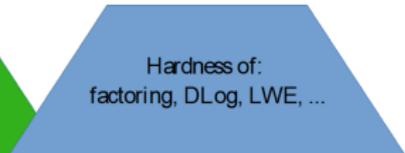
Design simple round function...  
Then iterate until no  
longer attackable



What statements  
can be proven?  
  
What can be attacked?



What statements  
can be proven?



# Conclusions

- Lots of exciting new developments in "high functionality cryptography" - some are likely here to stay
- ... leading to lots of exciting research for design and analysis of symmetric crypto and hashing
- Industry interest is growing, demand for standards to support interoperability and increase trust

# On FHEMPCZK-friendly symmetric crypto

---

Christian Rechberger, TU Graz

June 14, 2023



Thomas Jakobsen and Lars R. Knudsen. “The Interpolation Attack on Block Ciphers”. In: *FSE*. Vol. 1267. Lecture Notes in Computer Science. Springer, 1997, pp. 28–40.



Kaisa Nyberg and Lars R. Knudsen. “Provable Security Against a Differential Attack”. In: *J. Cryptology* 8.1 (1995), pp. 27–37.