Feistel Structures for MPC, and More

Arnab Roy[1]

(joint work with Martin Albrecht[2], Lorenzo Grassi[3], Léo Perrin[4], Sebastian Ramacher[3], Christian Rechberger[3], Dragos Rotaru[1,5] and Markus Schofnegger[3])

University of Bristol, Bristol, UK[1]

Royal Holloway, University of London[2]

TU Graz, Austria[3]

Inria, Paris, France[4]

KU Leuven, Belgium[5]

Motivation and background

Background

- In recent years significant progress in the areas of MPC, FHE, ZK
- Communication protocol (Theory $\rightarrow$ Practice)
- Many new applications are being developed
- Examples include
  1. Private set intersection, privacy preserving search
  2. Statistical computation on sensitive data
  3. Verifiable computation
  4. Cloud computation

Background

- The role of symmetric-key primitives - Hash function, PRF, PRP
- Specific requirements from the protocols
- Examples of typical conditions
  - Low number of multiplications (over integers): MPC, ZK
  - Low number of AND: MPC
  - Low multiplicative depth: FHE, MPC
- Designs must be secure

Need for new design

- Aren't there secure symmetric-key designs (AES, SHA2, SHA3, Blake, ...) ?

Need for new design

- Aren't there secure symmetric-key designs (AES, SHA2, SHA3, Blake, ...) ?
- Yes, but they are not enough

Need for new design

- Aren't there secure symmetric-key designs (AES, SHA2, SHA3, Blake, ...) ?
- Yes, but they are not enough
- Example:
    - SHA2 **optimized**: $\approx$ 25000 AND gates (per compression function)
    - AES is not efficient for MPC

Need for new design

- Aren't there secure symmetric-key designs (AES, SHA2, SHA3, Blake, ...) ?
- Yes, but they are not enough
- Example:
  - SHA2 **optimized**: $\approx$ 25000 AND gates (per compression function)
  - AES is not efficient for MPC
- Uses many XOR: non-linear after embedding into $\mathbb{F}_p$

New design endeavours

- New type of symmetric-key designs
- New design challenges: Minimize
  - AND depth and/or No. of ANDs (per bit)
  - multiplicative complexity and/or depth (per bit)
- Can we design primitives which minimize one or more of these metrics?
- Example
  - MiMC, Feistel-MiMC [ZKP, MPC friendly]
  - Flip, Rasta [FHE friendly]
  - LowMC, Legendre PRF [MPC friendly]
  - GMiMC [ZKP, MPC friendly]
  - More recent designs
- Present new cryptanalysis challenges

ZKP friendly

Hash function for Zero-knowledge proof system

- Finite field (large) friendly hash
- Different from the designs optimized for x86 (binary rings)
  - operations over $\mathbb{Z}_2$ or $\mathbb{F}_2$ makes them very slow for ZK system
  - Can not use BLAKE2b, SHA2, SHA3
- First new designs: MiMC, Feistel-MiMC
- Recent designs
  - GMiMC (SNARK friendly)
  - Poseidon(SNARK friendly), Starkad (STARK friendly)
  - Vision(STARK friendly), Rescue (SNARK friendly)

MiMCHash

- We work over a field $\mathbb{F}$

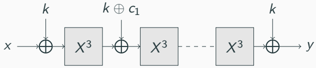


Figure 1: MiMC

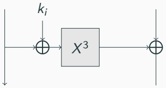
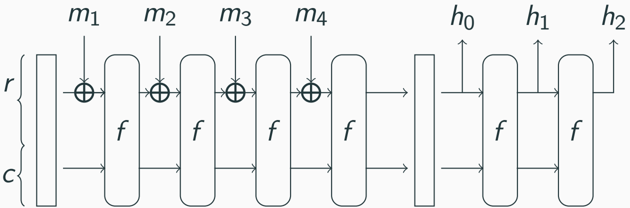


Figure 2: Feistel-MiMC

- Simple design idea:
    1. Add (round) key
    2. Add round constant
    3. Repeat
- Uses Sponge mode
- **Problem**: expanding to $>$ 512-bit $=$ 2 elements in $\mathbb{F}$, for 128 bit security

Sponge



- f is a bijection
- $c = 256$; One $\mathbb{F}$ element

GMiMC: Extension of MiMC

- Uses Generalized Unbalanced Feistel with

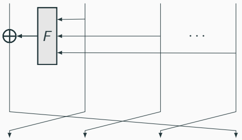


Figure 3: Contracting round function

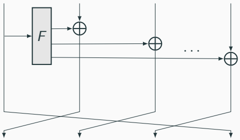
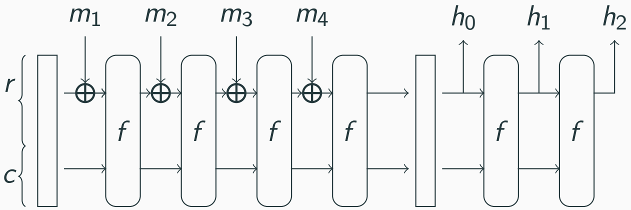


Figure 4: Expanding round function

- Round function $F_i(x_2, \ldots, x_t, k_i) = (\sum x_j + rc_i + k_i)^3$ for CRF
- Round function $F_i(x, k_i) = (x + rc_i + k_i)^3$ for ERF
- ~~$k_i = k$~~ $\quad k_i = (i+1)k$
- No. of branches $t << \log_2(|\mathbb{F}|)$

GMiMCHash

- Uses the sponge mode with capacity $c = 256$;
- No. of branches $t > 2$
- **Security Goal**: 128-bit security



Cryptanalysis

- Use of APN function ($x \to x^3$) protects against differential and other statistical attacks
- Security relies mostly on algebraic cryptanalysis
  - Interpolation, GCD, Groebner basis
  - Interpolation analysis (with root finding) for Hash function
- Mostly exploiting the degree of the output polynomial
- No weakness found in the GMiMCHash beyond birthday bound (to the best of our knowledge)

Performance and application

- In **SNARK**: GMiMCHash is faster ($\approx 1.2x$) than MiMC/Fesitel-MiMCHash
- Main advantage is the expansion
- Application examples: ZCash (ZKSNARK), Smart contract, STARK application etc.
- StarkWare Hash challenge (https://starkware.co/hash-challenge/)
  - GMiMCHash, Feistel-MiMCHash
  - Poseidon and Starkad (SNARK and STARK friendly resp.)
  - Vision and Rescue (STARK and SNARK friendly resp.)

MPC Friendly

MPC friendly encryption

- Ciphers optimized for $x86$ are not suitable for MPC
- **Security aim**: Secure block cipher
- First new design: LowMC (over $\mathbb{F}_2$)
- Other: Legendre PRF (over integers)
- Legendre PRF is secure only upto birthday bound
- In **SPDZ**: MiMC turned out to be efficient in mode of operation (e.g. Authenticated Encryption) (!!)
- What about GMiMC?

GMiMC in MPC

- **Securty Goal**: At least 128-bit key security
- Efficiency in MPC: preprocessing + online computation
- $\mathrm{GMiMC}_{\mathsf{erf}}$ and $\mathrm{GMiMC}_{\mathsf{crf}}$ have very fast preprocessing phase
- **Reason**: Least no. of multiplications per (encryption) round
- Avoids linear scaling with increased blocks (only known case)
- Example: $\mathrm{GMiMC}_{\mathsf{erf}}$ is 5.5x faster than MiMC (with 16 blocks)
- Gain in **throughput**

Yet another application

PQ signature

- A new application
- **Picnic**: Uses ZKB++; ZKP-based signature scheme
- Minimize: No. of multiplications $\times \log_2(|\mathbb{F}|)$
- Current best option: LowMC
- Can we use GMiMC?

GMiMC in Picnic

- Pushing the MiMC design strategy for small field
- **Security Goal**: 256-bit key security with 256-bit input

Scheme	(n, t, R)	Sign	Verify	View size
MiMC	$(256, 1, 162)$	333.97 ms	166.28 ms	83456 bits
	$(272, 1, 172)$	92.45 ms	46.32 ms	94112 bits
$\text{GMiMC}_{\text{erf}}$ over $\mathbb{F}_{2^n}$	$(33, 8, 56)$	**3.34 ms**	**2.29 ms**	1848 bits
LowMC-$(256, 10, 38)$	-	3.74 ms	3.52 ms	1140 bits
LowMC-$(256, 1, 363)$	-	9.55 ms	7.12 ms	1089 bits

- GMiMC is comparable to LowMC

Conclusion and open questions

- Finite field friendly designs
- Design space exploration
- Open questions in design and analysis
  - Cryptanalysis methods over $\mathbb{F}_p$ (completely unknown)
  - New design principle?
  - Bounds on multiplicative complexity
  - How far can we extend current cryptanalysis techniques?
  - Can we obtain generic (algebraic) complexity results for security?

Updates on MiMC, GMiMC and similar designs on https://byt3bit.github.io/primesym/ (new, still under construction)

Thank you!