Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity

Arnab Roy[1]

(joint work with Martin Albrecht[2], Lorenzo Grassi[3], Christian Rechberger[1,3] and Tyge Tiessen[1])

Technical University of Denmark[1]

Royal Holloway, University of London[2]

TU Graz[3]

Background

In recent years significant progress in - MPC, FHE, ZK

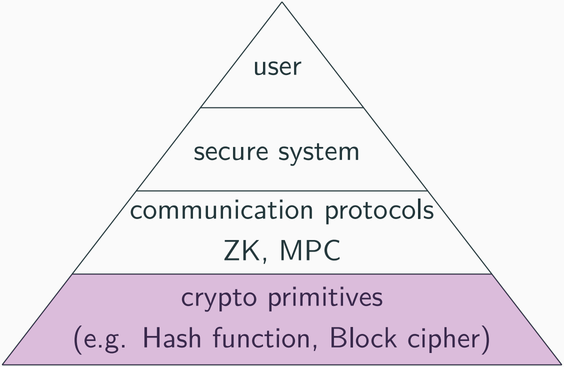
Communication protocol (Theory $\rightarrow$ Practice)

Many applications are being developed

Examples include

- Private set intersection, privacy preserving search
- Statistical computation on sensitive data
- Verifiable computation
- Cloud computation

Security of systems



Performance of symmetric-key algorithms can improve the efficiency of protocols

Motivation

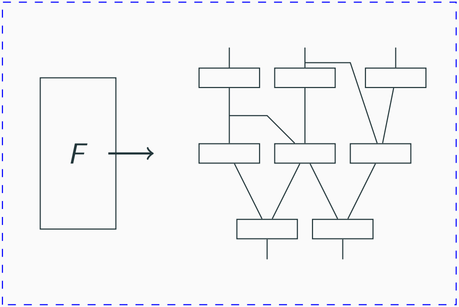
Our focus: Verifiable computation based on **SNARK** [BSCG+13]

Recently developed application around SNARK - ZeroCash [SCG+14]

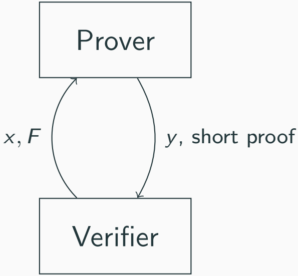
Motivation: constriction of performance due to *private-key crypto*

Our focus: constriction due to **Hash function**

SNARK



arithmetic circuit C for F, witness - w for input x



check $F(x) = y$
without computing F

Let $L_C = \{x \in \{0,1\}^n : \exists w \in \{0,1\}^h, C(x,w) = 0\}$
Prover knows w, keeps it secret

Rank-1 constraints

- An $\mathbb{F}$-arithmetic circuit $\mathcal{C} : \mathbb{F}^n \times \mathbb{F}^h \to \mathbb{F}^\ell$
- The Arithmetic Circuit Satisfiability (ACS) of $\mathcal{C}$ is given by relation $\mathcal{R} = \{(x, a) \in \mathbb{F}^n \times \mathbb{F}^h : \mathcal{C}(x, a) = 0\}$
- The circuit consists of bilinear gates only
- The SNARK algorithm generates the proof for satisfiability of a system of rank-1 quadratic constraints over the field $\mathbb{F}$.
- The systems looks like

$$\langle A_i, w\rangle \cdot \langle B_i, w\rangle = \langle C_i, w\rangle$$

where $i = 1, \ldots, N_c$ and $w \in \mathbb{F}^{N'}$.
$N_c \to$ no. of constraints; $N' \to$ no. of variables.

Computational model

Cost of computation - (**MULT**, **ADD**); (**AND**, **XOR**)

Cost of single XOR (or ADD) is negligible *compared* to single MULT/AND

Caution: Very large number of XORs (or ADDs) influences the cost

Similar cost model, less extreme: Masking (for side-channel attack resilient crypto)

General idea

- Linear/Affine functions, Mult with a constant (almost free)
- Non-linear functions (expensive)

Computation cost: symmetric-key primitives

The well-known primitives use operations over $\mathbb{F}_2$ or (and) $\mathbb{F}_{2^n}$

Example

- SHA-256 over $\mathbb{F}_2$, $\mathbb{Z}_{2^{32}}$
- SHA-3 over $\mathbb{F}_2$
- AES over $\mathbb{F}_{2^8}$
- PRINCE over $\mathbb{F}_{2^4}$ and $\mathbb{F}_2$

MULT or AND - $x \cdot y$

Typical examples

- Linear: XOR, ADD, Rotation
- Non-linear: S-box, modular addition, bitwise AND

MPC/FHE/ZK friendly

Protocols usually require computations over $\mathbb{F}_p$

Symmetric-key computations: Embed the circuit in $\mathbb{F}_p$

- Operations over $\mathbb{F}_2$ are expressed over $\mathbb{F}_p$
- Operations over $\mathbb{F}_{2^n}$ are expressed over $\mathbb{F}_2$, then embedded in $\mathbb{F}_p$
- Example: XOR over $\mathbb{F}_2$ changes over $\mathbb{F}_p$

FHE friendly - Low circuit depth

MPC friendly - Low circuit depth and/or Low number of multiplications

SNARK friendly - Low number of multiplications

Recent results - FLIP [MJSC16] , LowMC [ARS$^+$15], Legendre symbol based PRF [GRR$^+$16]

SNARK friendly design

Mixing different fields is NOT useful

Embedding PRP/PRF circuit over $\mathbb{F}_2$ into $\mathbb{F}_p$ has cost issues

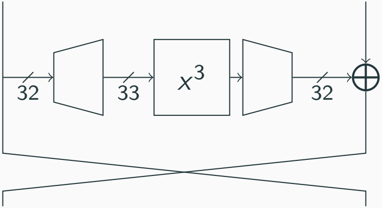
Efficient design over $\mathbb{F}_p$? **MiMC** family

Block cipher: MiMC-n/n, MiMC-$2n/n$

Hash function: MiMC-Hash (uses **sponge mode**)

An old design: KN cipher

- Knudsen-Nyberg cipher: Round function uses APN function over finite field
- 64-bit block cipher using Feistel mode of operation



- Broken with **Interpolation Attack** (algebraic)
- This way of design was abandoned

MiMC block-cipher: MiMC-n/n

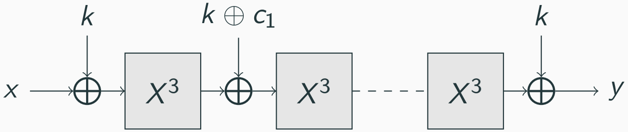


Figure 1: MiMC in Even-Mansour mode

- Note: $n = odd$ so that x^3 is a permutation
- Randomly chosen round constants (fixed)
- Round key
  - Single k in $\mathbb{F}_{2^n}$
  - $(k_1, k_2) \in \mathbb{F}_{2^n}^2$ on alternate rounds ($k_1 \neq k_2$)
- Number of rounds: $\frac{n}{\log 3}$ or $\frac{\log p}{\log 3}$
- Same design strategy over $\mathbb{F}_{2^n}$ and $\mathbb{F}_p$

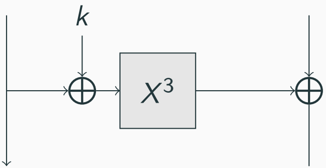


Figure 2: MiMC in Feistel mode

Uses x^3 over $\mathbb{F}_{2^n}$ with Feistel mode (No linear layer)

Number of rounds: $\frac{2n}{\log 3}$ or $\frac{2\log p}{\log 3}$

Round key and round constants: same as MiMC-n/n.

Hash function

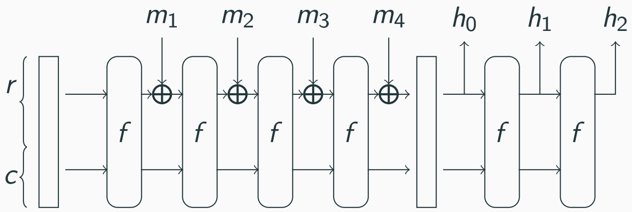


Figure 3: Sponge mode

Sponge mode instantiated by MiMC permutation with a fixed key

In the SNARK setting we use MiMC-n/n

It is possible to use MiMC-$2n/n$ for large block size

Cryptanalysis

- Optimal differential property for - x^3
- Simple differential attack is not possible for full rounds
- The degree of the polynomial $P(x)$ representing the cipher has full degree over $\mathbb{F}_{2^n}$
- **Interpolation attack** requires $\approx 2^n - 1$ plaintexts

Cryptanalysis

- Consider two polynomials $E(K, x_1) - y_1$ and $E(K, x_2) - y_2$ over $\mathbb{F}_q[K]$
- The GCD of these two polynomials is $(K - k)$ where k is the unknown secret key
- **GCD attack** recovers the unknown key
- **Complexity** is $\mathcal{O}(d \log^2 d)$

Note: GCD attack assumes that adversary can compute the necessary polynomial(s)

Cryptanalysis

- **Higher-order differential** attack requires 2^n plaintexts
- APN function provides security against **linear** attacks
- **Invariant subfield attack**: Poor choice of round constants allows this attack
- In this attack subsequent states following the input value belong to the same subfield
- Randomly chosen round constants thwart this attack
- Over $\mathbf{F}_p$ this attack does not apply

MiMC in SNARK setting

- Each round can be expressed with

$$X + \underbrace{k_i + C_i}_{\alpha} + U = 0, U \cdot U = Y$$

$$Y \cdot U = Z$$

- The equations are combined to obtain

$$(X + \alpha)(X + \alpha + Y) = Y + Z$$

- These equations represent the **rank-1 constraints**
- Each round has **two** multiplications (for witness generation)

Experimental results

- We implemented a part of the SNARK algorithm to generate the circuit and witness
- Compared it with SHA-256 (libsnark implementation)
- SHA-256 takes $\approx$ **73 ms** while MiMC takes $\approx$ **7.8 ms**
- SHA-3 takes almost the same time as SHA-256
- Also compared with the LowMC and Keccak (SHA-3)

Comparison

	MiMC	LowMC		Keccak-$[1600, 24]$
		$\#r = 16$	$\#r = 55$	
		$m = 196$	$m = 20$	
total time	7.8ms	90.3ms	271.2ms	75.8ms
constraint generation	6.3ms	13.5ms	9.2ms	65.2ms
witness generation	1.5ms	76.8ms	262.0ms	10.6ms
# addition	646	8420888	28894643	422400
# multiplication	1293	9408	3300	38400
# rank-1 constraint	646	4704	2200	38400

MiMC and LowMC permutations have block size 1025

Our C++ implementation is available on

`https://github.com/byt3bit/mimc_snark.git`

New Results

- **Motivation**: Construct secure hash function over smaller prime fields
- **MiMC limitation**: ≈ 1024 bit permutation can be constructed over 1024 bit prime or 512 bit prime fields
- We construct block cipher(s) using Generalized Unbalanced Feistel
- Use the cipher with fixed key in Sponge mode
- New construction shows significant improvement in performance over MiMC
- **Example**: Secure (as SHA-256) hash function over 128 bit prime field

Conclusion

New efficiency criteria $\rightarrow$ Resurrection of an abandoned design strategy

MiMC also shows competitive performance in MPC setting when used as PRF ([GRR$^+$16])

Metric: Effect of large number XOR/ADD is clear from experimental results but *How to quantify* ?

Can we use polynomial instead of monomial ?

Thank you!

Remarks

Monomial with exponent $2^t + 1$

Problem: Resulting polynomial becomes sparse $\implies$ efficient attack

Monomial with exponent $2^t - 1$

Problem: Number of multiplication increases

Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner.
Ciphers for MPC and FHE.
In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Computer Science*, pages 430–454. Springer, 2015.

Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza.
***SNARKs for C: Verifying Program Executions Succinctly and in Zero Knowledge*, pages 90–108.**
Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

Lorenzo Grassi, Christian Rechberger, Dragos Rotaru, Peter Scholl, and Nigel P. Smart.

Mpc-friendly symmetric key primitives.

In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 430–443, New York, NY, USA, 2016. ACM.

Pierrick Méaux, Anthony Journault, François-Xavier Standaert, and Claude Carlet.

Towards stream ciphers for efficient fhe with low-noise ciphertexts.

In *Proceedings of the 35th Annual International Conference on Advances in Cryptology — EUROCRYPT 2016 - Volume*

9665, pages 311–343, New York, NY, USA, 2016. Springer-Verlag New York, Inc.

E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza.
Zerocash: Decentralized anonymous payments from bitcoin.
In *2014 IEEE Symposium on Security and Privacy*, pages 459–474, May 2014.