Rasta

Christoph Dobraunig, Maria Eichlseder, Lorenzo Grassi, Virginie Lallemand, Gregor Leander, Florian Mendel, Christian Rechberger

September 8, 2017

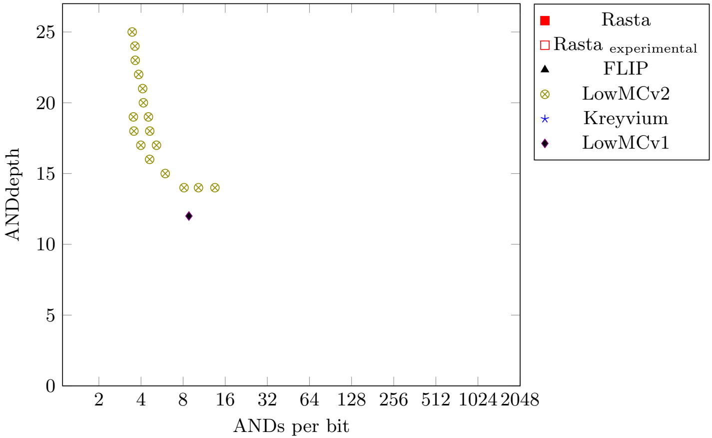
Motivation

Design cipher with low ANDdepth and few ANDs per bit

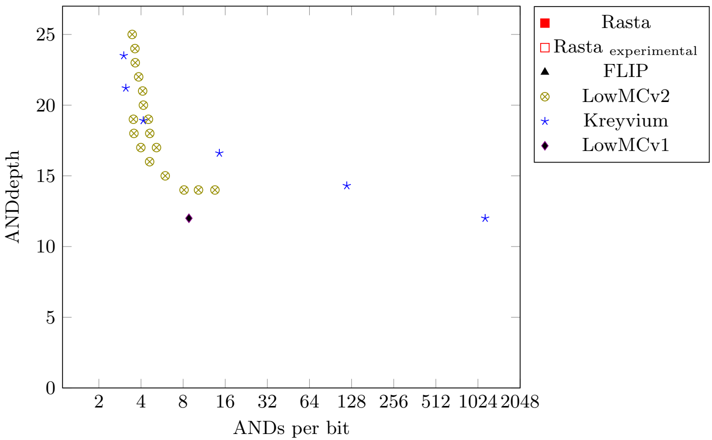
Remove huge ciphertext expansion in applications of FHE

In general interesting problem, e.g. for cheap side-channel attack countermeasures

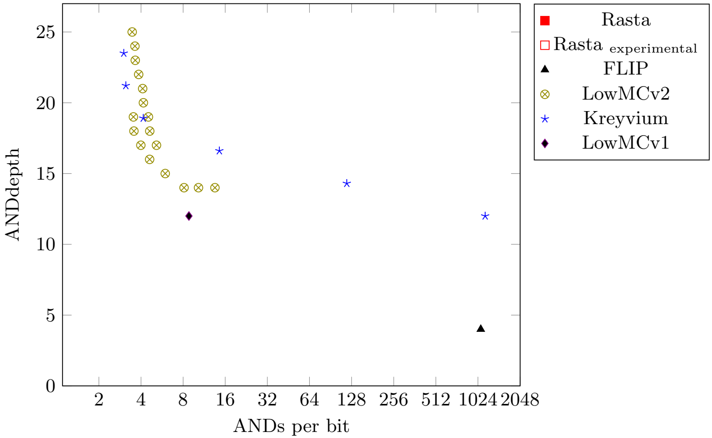
Comparison to Other Designs



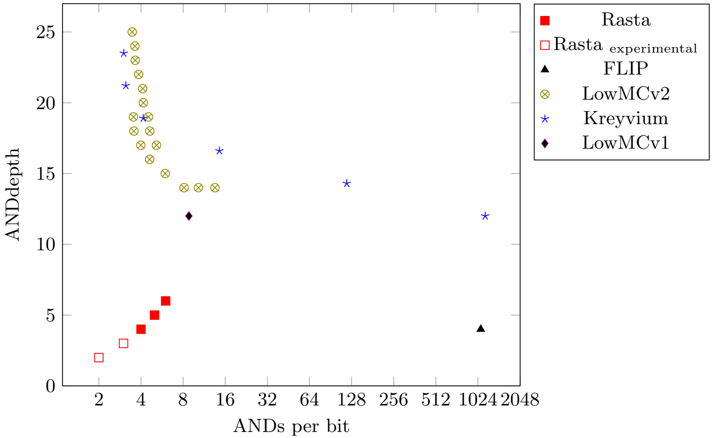
Comparison to Other Designs



Comparison to Other Designs



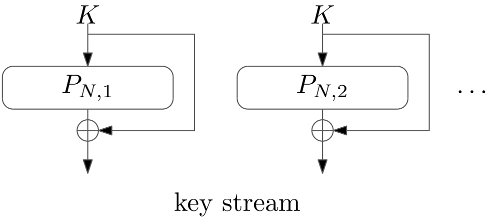
Comparison to Other Designs



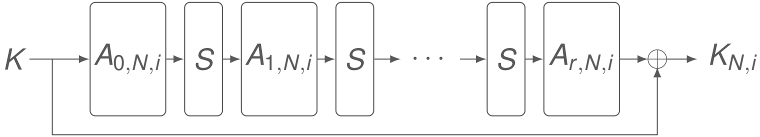
Rasta

Stream cipher based on public permutation

- Different permutations to generate key stream
- Each permutation evaluated once
- Choice of permutation depends solely on public parameters
- High-level idea to make relevant computations of the cipher independent of the key was first propsed by Méaux, Journault, Standaert and Carlet at Eurocrypt 2016.



Rasta



Seed PRNG with public values

- “Randomly” generate invertible matrix
- “Randomly” generate round constant

PRNG does not influence relevant AND metric

Design Rationale

- Changing affine layers against
  - Differential and impossible differential attacks
  - Cube and higher-order differential attacks
  - Integral attacks
- Wide permutation and secret key $\gg$ security level against
  - Attacks targeting polynomial system of equations
  - Attacks based on linear approximations
  - MitM attacks
  - Huge security margin despite very few rounds

Instances of Rasta, derived blocksizes

Security level	Rounds				
	2	3	4	5	6
80-bit	$2^{21.2}$	2^{12}	327	327	219
128-bit	$2^{33.2}$	2^{18}	1 877	525	351
256-bit	$2^{65.2}$	2^{34}	$2^{18.8}$	3 545	703

Instances of Rasta

Block sizes depend on bounds on

- The existence of good linear approximations
- Total number of different monomials

Block sizes are not based on attacks

Cryptanalysis

- SAT solver
  - Exhaustive search performs better for more than 1 round
- Various dedicated attacks
  - For various versions of SAS
  - Variants of 2-round Rasta where block size = security level
- Grobner bases and related algebraic attacks
  - Even no improvement for variants of 2-round Rasta where block size = security level
- Experiments with toy versions
  - No no-random behaviour

Agrasta: More agressive parameters

Security level	Rounds	Block size
80-bit	4	81
128-bit	4	129
256-bit	5	257

Closer to what we can attack, still large security margin

Benchmarking of FHE use-case

Implemented Rasta using Helib

Compared with
- LowMC
- Trivium/Kreyvium
- Flip

For Trivium, Kreyvium and FLIP no public Helib implementation available

Benchmarking 80-bit Cipher Security

Cipher	n	r	t_{total}	BGV slots	BGV lev.	BGV sec.
LowMC v1	128	11	2011.9	720	20	74.05
H. t. LowMC v2	256	12	1721.3	600	21	62.83
Trivium	57	12	∼1560.0	504	–	–
Trivium	136	13	∼4050.0	682	–	–
FLIP	1	4	∼3.5	600	12	–
Rasta	327	4	397.8	224	12	89.57
Rasta	327	4	609.6	600	13	62.83
Rasta	327	5	766.7	600	14	62.83
Rasta	219	6	610.6	600	14	62.83
Agrasta	81	4	98.9	600	12	81.41

Benchmarking 128-bit Cipher Security

Cipher	n	r	t_{total}	BGV slots	BGV lev.	BGV sec.
LowMC v1	256	12	3785.2	480	21	106.31
Kreyvium	12	42	∼1760.0	504	–	–
Kreyvium	13	124	∼4430.0	682	–	–
FLIP	1	4	∼39.0	720	13	–
Rasta	525	5	912.1	682	14	90.39
Rasta	351	6	2018.6	720	15	110.74
Agrasta	129	4	217.4	682	12	127.50

Benchmarking 256-bit Cipher Security

Cipher	n	r	t_{total}	BGV slots	BGV lev.	BGV sec.
LowMCv2	Too big to run					
Kreyvium	Not specified for this security level					
FLIP	Not specified for this security level					
Rasta	703	6	5543.2	720	16	89.93
Agrasta	257	5	1763.8	1800	15	210.68

Conclusion

New interesting design approach

Even conservative versions competitive in benchmark

Huge gap between known attacks and bounds

Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives

Christian Rechberger
Joint work with Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Daniel Slamanig, Greg Zaverucha

Tor's birthday MMC, Sept 8, 2017

IAIK, Graz University of Technology

Overview

Digital Signatures in a post-quantum world

- RSA and DLOG based schemes insecure

New schemes

- based on new structured hardness assumptions (lattices, codes, isogenies, etc.)
- based on symmetric primitives: Hash-based signatures

Other alternatives only relying on symmetric primitives?

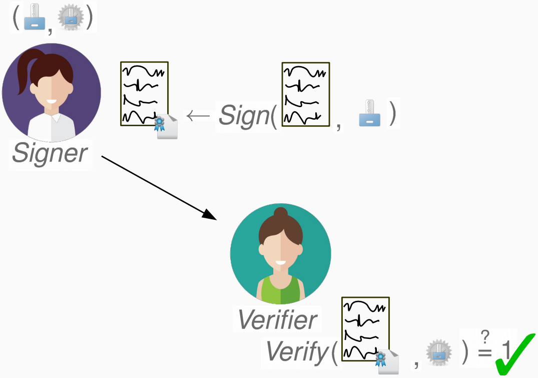
High-level View

Recent years progress in two areas

- Symmetric-key primitives with few multiplications
- Practical ZK-Proof systems over general circuits

New signature schemes based on these advances

Digital Signatures

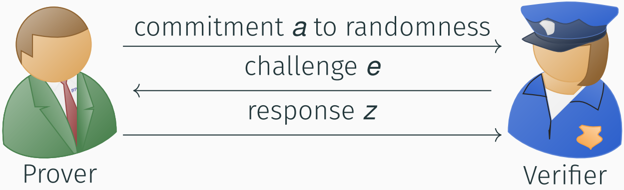


Existential Unforgeability under Chosen-Message Attacks

- Adversary may see signatures on arbitrary messages
- Still intractable to output signature for new message

Σ-Protocols

Three move protocol:



- Important that e unpredictable before sending a
- aka (Interactive) Honest-Verifier Zero-Knowledge Proofs

Non-interactive variant via Fiat-Shamir [FS86] transform

Digital Signatures from Σ-Protocols

Well known methodology

One-way function $f_k : D \to R$ with $k \in K$

- $sk \stackrel{R}{\leftarrow} K$
- $y \leftarrow f_{sk}(x)$, $pk \leftarrow (x, y)$

Signature

- Σ-protocol to prove knowledge of sk so that $y = f_{sk}(x)$
- Use Fiat-Shamir transform to bind message to proof
  $e \leftarrow H(a \| m)$

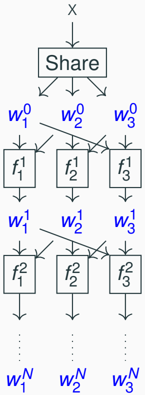
ZKBoo [GMO16]

Efficient Σ-protocols for arithmetic circuits

- generalization, simplification, + implementation of "MPC-in-the-head" [IKOS07]

Idea

1. (2,3)-decompose circuit into three shares
2. Revealing 2 parts reveals no information
3. Evaluate decomposed circuit per share
4. Commit to each evaluation
5. Challenger requests to open 2 of 3
6. Verifies consistency



Efficiency

- Heavily depends on #multiplications

ZKB++

Improved version of ZKBoo:

- Remove redundant information from views
- Remove redundant checks
- Proof size reduction to less than half the size
- But without extra computational cost

LowMC [ARS+15, ARS+16]

Substitution-permutation-network design

- Very lightweight S-box with one AND gate per bit
- S-box layer is only partial
- Very expensive affine layer with $n/2$ XOR gates per bit.
- Allows selection of instances minimizing, e.g.
  - ANDdepth,
  - number of ANDs, or
  - ANDs / bit

Blocksize n	**S-boxes** m	**Keysize** k	**Data** d	**ANDdepth** r	**# of ANDs**	**ANDs/bit**
256	2	256	256	232	**1392**	5.44
512	66	256	256	**18**	3564	6.96
1024	10	256	256	103	3090	**3.02**

Table 1: LowMC parameters for 128-bit PQ-security

Fish

Fish:

- Turn ZKB++ and OWF into signature scheme
- via Fiat-Shamir Transform
- Instantiate OWF with LowMC v3
- $\Rightarrow$ EUF-CMA security in the ROM

Picnic

Picnic:

- Turn ZKB++ and OWF into signature scheme
- via Unruh Transform
- Instantiate OWF with LowMC v3
- $\Rightarrow$ EUF-CMA security in the QROM

Unruh Transform incurs overhead in signature size

- But careful tweaking reduces overhead to factor 1.6

Signature Size

- Recall: OWF $f_k : D \to R$, $sk \xleftarrow{R} K$, $pk \leftarrow (x, f_{sk}(x))$
- Security parameter κ

OWF represented by arithmetic circuit with

- ring size λ
- multiplication count a

Signature size: $|\sigma| = c_1 + c_2 \cdot (c_3 + \lambda \cdot a)$ where c_i are polynomial in κ

OWF with few multiplications?

Build OWF from

name	security	$\lambda \cdot a$	
AES	128	5440	$\mathbb{F}_2$ approach
AES	128	4000?	$\mathbb{F}_{2^4}$ approach
AES	256	7616	$\mathbb{F}_2$ approach
SHA-2	256	> 25000	
SHA-3	256	38400	
Noekeon	128	2048	
Trivium	80	1536	
PRINCE		1920	
Fantomas	128	2112	
LowMC v3	128	< 800	
LowMC v3	256	< 1400	
Kreyvium	128	1536	
FLIP	128	> 100000	
MIMC	128	10337	
MIMC	256	41349	

Signature Size Comparison

name	security	$\|\sigma\|$
AES	128	339998
AES	256	473149
SHA-2	256	1331629
SHA-3	256	2158573
LowMC v3	256	108013

Example of Exploration of Variation of LowMC Instances

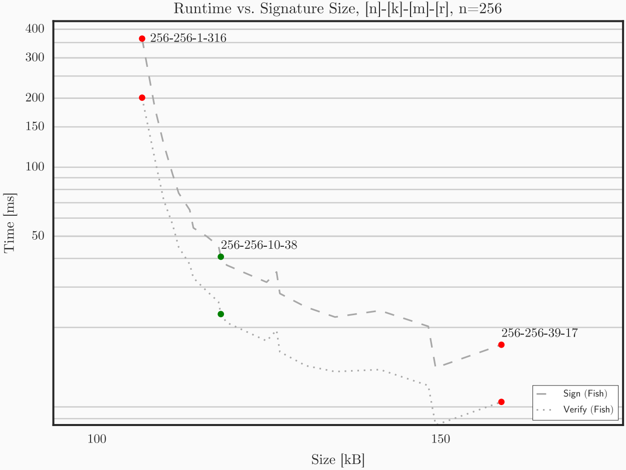


Figure 1: Measurements for instance selection (128-bit PQ-security).

Comparison with other recent proposals

Scheme	Gen	Sign	Verify	$\lvert sk \rvert$	$\lvert pk \rvert$	$\lvert \sigma \rvert$	M
Fish-10-38	0.01	29.73	17.46	32/64		$116K$	ROM
Picnic-10-38	0.01	31.31	16.30	32/64		$191K$	QROM
MQ 5pass	1.0	7.2	5.0	32	74	$40K$	ROM
SPHINCS-256	0.8	1.0	0.6	$1K$	$1K$	$40K$	SM
BLISS-I	44	0.1	0.1	$2K$	$7K$	$5.6K$	ROM
Ring-TESLA	$17K$	0.1	0.1	$12K$	$8K$	$1.5K$	ROM
TESLA-768	$49K$	0.6	0.4	$3.1M$	$4M$	$2.3K$	(Q)ROM
FS-Véron	n/a	n/a	n/a	32	160	$\geq 126K$	ROM
SIDHp751	16	$7K$	$5K$	48	768	$138K$	QROM

Table 2: Timings (ms) and key/signature sizes (bytes)

Conclusion

ZKB++: Improved ZK proofs for arithmetic circuits

Fish/ Picnic: Two new efficient post-quantum signature schemes in ROM and QROM

Applications beyond signatures: NIZK proof system for arithmetic circuits in post-quantum setting

Outlook and Future Work

- Alternative symmetric primitives with few multiplications
  - Something new with even less multiplications than LowMC?
  - 256-bit secure variant of Trivium/Kreyvium?
- More LowMC cryptanalysis
  - More aggressive LowMC parameters with very low allowable data complexity, e.g. only 1 or 2 texts.
- Analysis regarding side-channels
- Unruh Transform with constant overhead?

Thank you.

- To appear in ACM CCS'17.
- Preprint: `https://ia.cr/2017/279`

Supported by: prisma cloud, CryptoAction, PQCRYPTO ICT-645622

References i

[ARS+15] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner.
Ciphers for MPC and FHE.
In *EUROCRYPT*, 2015.

[ARS+16] Martin Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner.
Ciphers for MPC and FHE.
Cryptology ePrint Archive, Report 2016/687, 2016.

[FS86] Amos Fiat and Adi Shamir.
How to prove yourself: Practical solutions to identification and signature problems.
In *CRYPTO*, pages 186–194, 1986.

[GMO16] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi.
ZKBoo: Faster zero-knowledge for boolean circuits.
In *USENIX Security*, 2016.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai.
Zero-knowledge from secure multiparty computation.
In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 21–30, 2007.