RUHR-UNIVERSITÄT BOCHUM

Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems

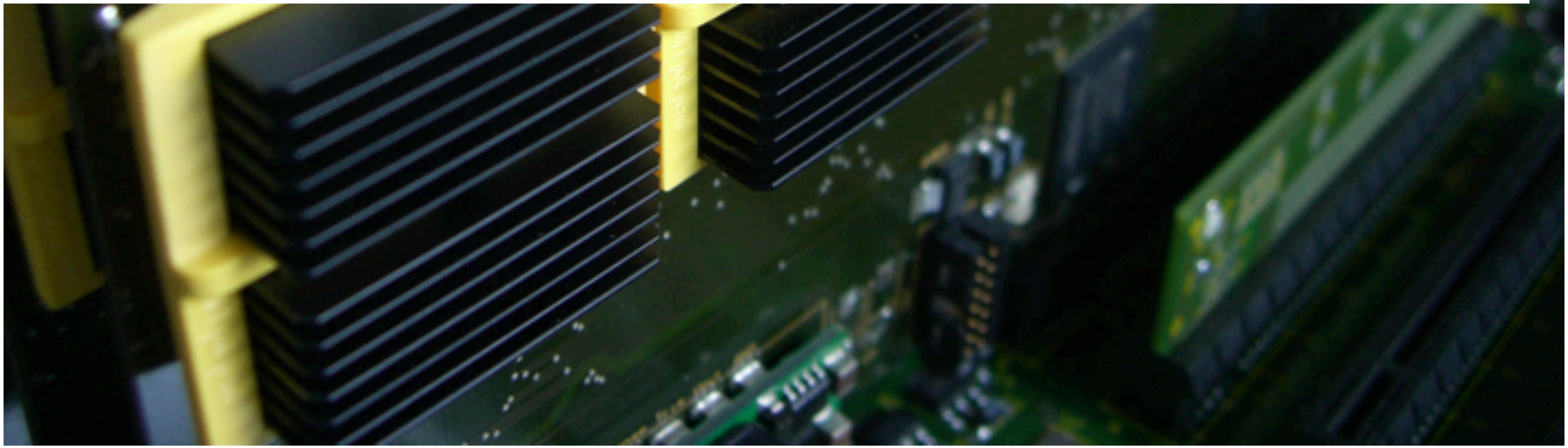
CHES 2012, Leuven, Belgium

Tim Güneysu[1], Vadim Lyubashevsky[2] and Thomas Pöppelmann[1]

[1] Horst Görtz Institute for IT-Security, Ruhr-University Bochum, Germany

[2] INRIA / ENS, Paris

12.09.2012



Outline

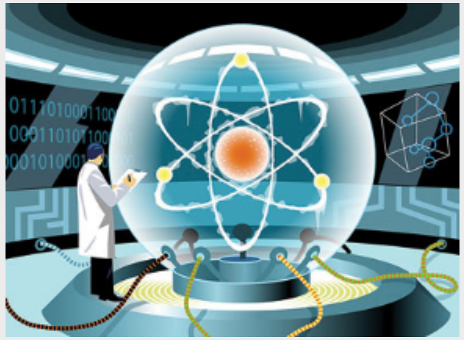
- Introduction
- Proposed Scheme
- FPGA Implementation
- Results
- Future Work

Outline

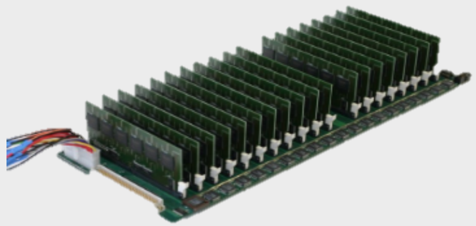
- **Introduction**
- Proposed Scheme
- FPGA Implementation
- Results
- Future Work

Motivation: Quantum Computers/Diversity

- Current asymmetric schemes rely on similar hard problems
  - RSA: Factoring
  - DSA/ECDSA: Discrete logarithm



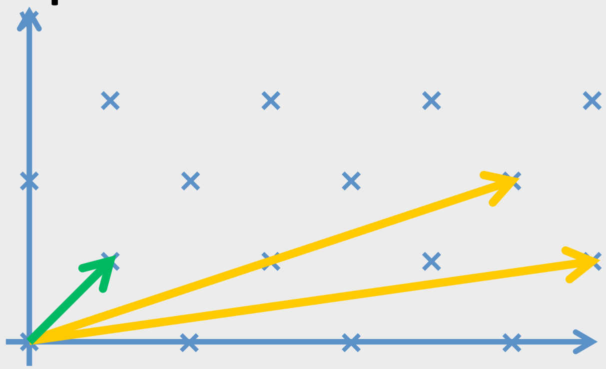
- Threats
  - Quantum computers (IBM: ~15 years?)
  - Mathematical/Cryptanalysis breakthrough



- We are in need for
  - New post-quantum secure schemes
    - ➢ Task of cryptographers
  - Efficient and secure implementations in hard- and software
    - ➢ Task of security engineers

Cryptographers' view: Lattice-Based Crypto

- Worst-case to average-case reductions
- Well-studied and (presumably) quantum secure problems
  - SVP, CVP, LWE …
  - Allow security reductions



- Classical (asymmetric) primitives: signature or encryption
- More versatile constructions: hash functions, PRFs, identity-based encryption, homomorphic encryption

Engineers' view: Lattice-Based Crypto

- Lattice-based does not always mean there are lattices inside
  - Arithmetic on polynomials (ideal lattices) or matices
  - Parallelizable: multi-core/hardware
  - FFT/NTT for high-performance

- **Current issues**
  - Large key sizes or ciphertext expansion
  - Selection of secure parameters still a challenge
  - First results are promising (you should have seen one already) but few implementations are published

Outline

- Introduction
- **Proposed Scheme**
- FPGA Implementation
- Results
- Future Work

Proposed Scheme: Preliminaries

- Ring $\boldsymbol{R} = Z_p[x]/(x^n+1)$
  - *p* is a prime (*p*=1 mod 2*n*)
  - *n* is a power of two
  - Coefficients in range [-(*p*-1)/2, (*p*-1)/2]

- Subset $\boldsymbol{R_k}$ ={polynomial in R with coefficients in the range [-*k*, *k*] }

- We always pick **uniformly random** out of *R* or R_k

Proposed Scheme: Efficient Variant of [Lyu12]

- Signature scheme by Lyubashevsky proposed at EUROCRYPT [Lyu12] provable secure in random oracle model (ROM)

- Efficiency improvement by a different hardness assumption: (Decisional) Ring-LWE with "aggressive" parameters

  – Decisional Compact Knapsack (DCK) problem requires to distinguish one sample (a,t) between
    A. Uniform distribution over $R \times R$
    B. $(a, t = as_1 + s_2)$, with uniformly random $a \in R$, $s_1, s_2 \in R_1$

  – Values s_1, s_2 only have -1/0/1 coefficients instead of Gaussian distribution (like in [LPR10])

Proposed Scheme: Key Generation

- **GEN**
  - Pick $\mathbf{s_1,s_2}$ from subset R_1
  - Pick **a** from $R = Z_p[x]/(x^n+1)$
  - Compute $\mathbf{t = as_1+s_2}$

  - Secret key: sk = (s_1,s_2)
  - Public key: pk = (a, t)

Proposed Scheme: Signing

- **SIGN**(*m*,*sk*)
  1. Pick y_1,y_2 from R_k
  2. c=H(Transform(r=ay_1+y_2),m)
  3. $z_1=s_1c+y_1$, $z_2=s_2c+y_2$
  4. If z_1, z_2 not in $R_{k\text{-}32}$ goto 1.
  5. z_2'=Compress(ay_1+y_2-z_2,z_2,p,k-32)
  6. Return σ=(z_1, z_2', c)

Proposed Scheme: Verification

- **VER**(σ=(z_1,z_2‘,c),*pk=(a,t), m*)
  1. If z_1,z_2‘ not in R_{k-32} **reject**
  2. If c=H(Transform(az_1+z_2‘-tc), m)

     then **accept**

     else **reject**

- **Correctness**: $az_1+z_2-tc=a(s_1c+y_1)+s_2c+y_2-(as_1+s_2)c=ay_1+y_2$

Proposed Scheme: Efficiency

- Transform/Compression cuts off parts of the signature that are neither needed for correctness nor for the proof (“higher-order bits”)

Parameters for 100 bit security p=8383489, n=512, k=2^{14}		
Signature: 8954 bit	Secret key: 1624 bit	Public key: 11776 bit

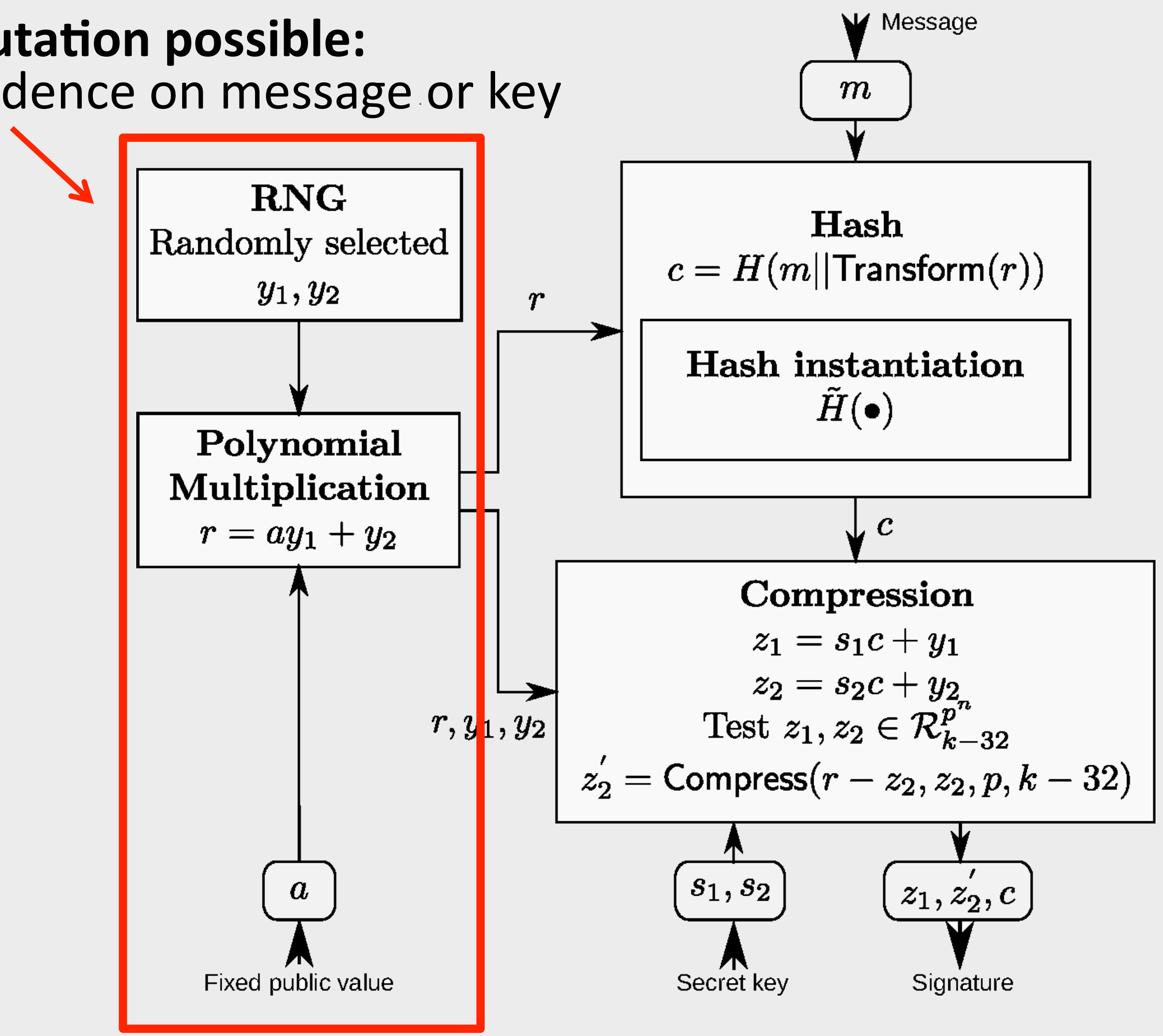
- **Rejection sampling step**
  - Success probability of 13,5 %
  - On **average 7 tries** until a valid signature is produced
  - Tradeoff between signature size/runtime/security

Outline

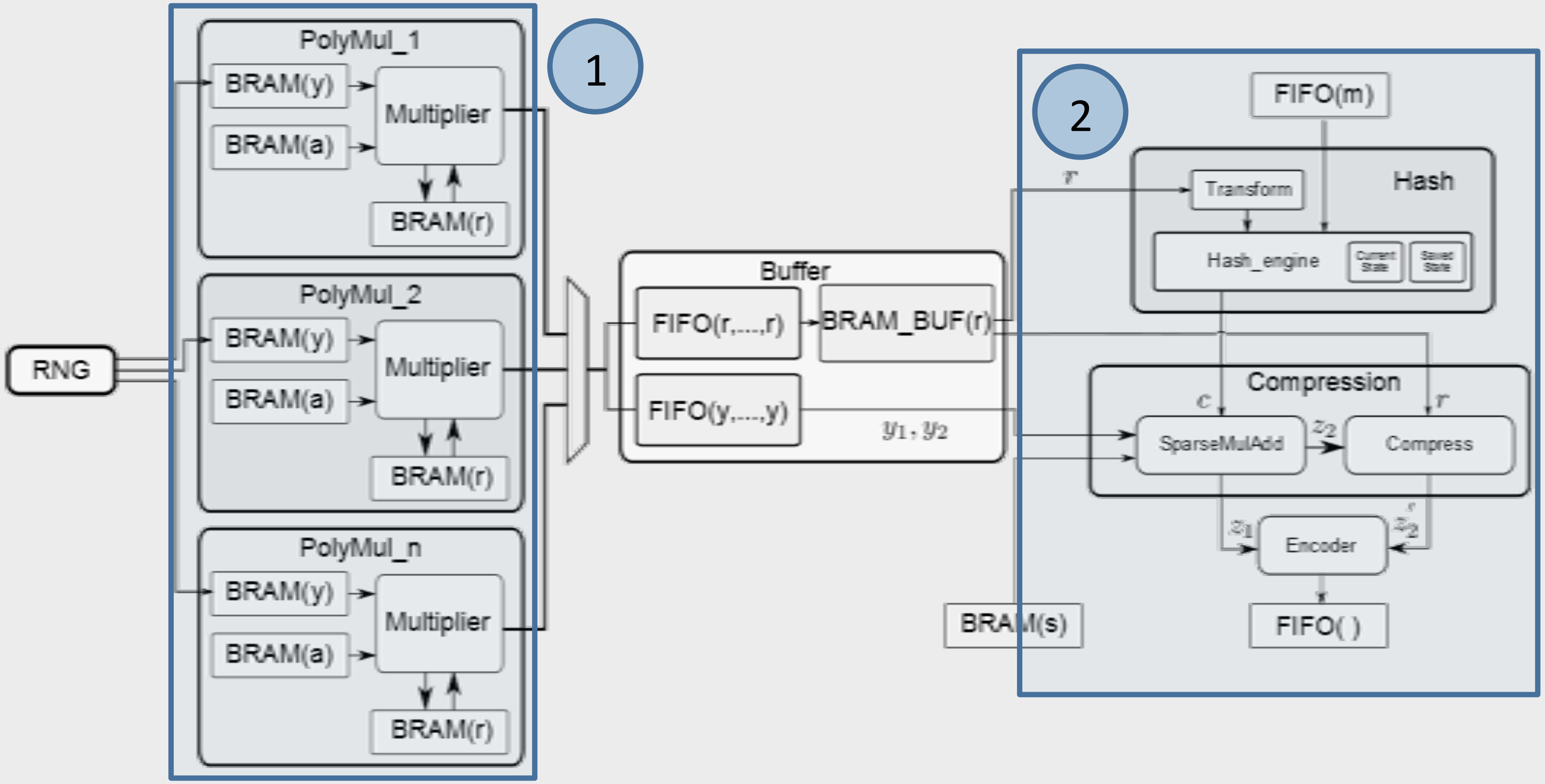
- Introduction
- Proposed Scheme
- **FPGA Implementation**
- Results
- Future Work

Implementation: Parallelization

Precomputation possible:
No dependence on message or key

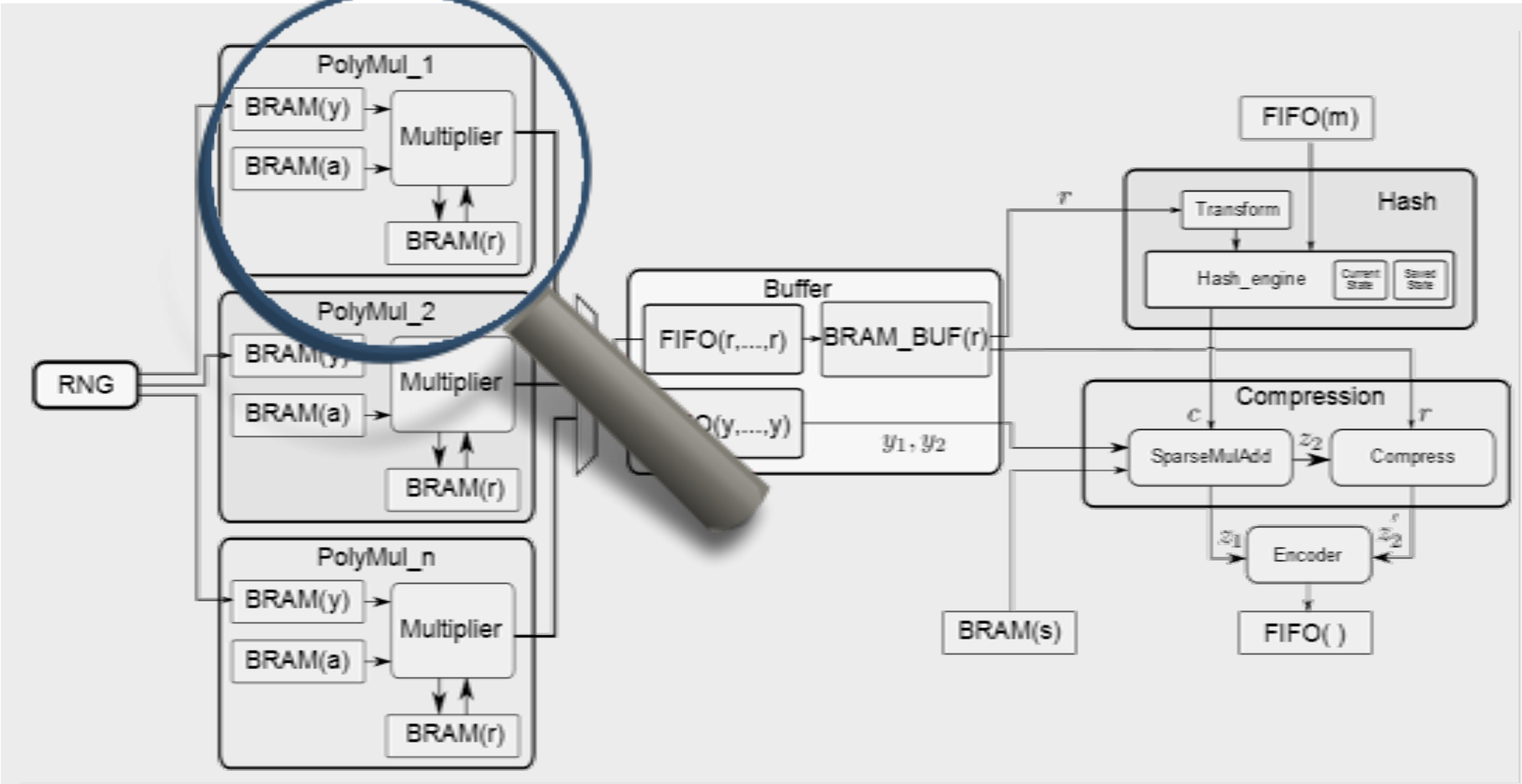


Implementation: FPGA Design

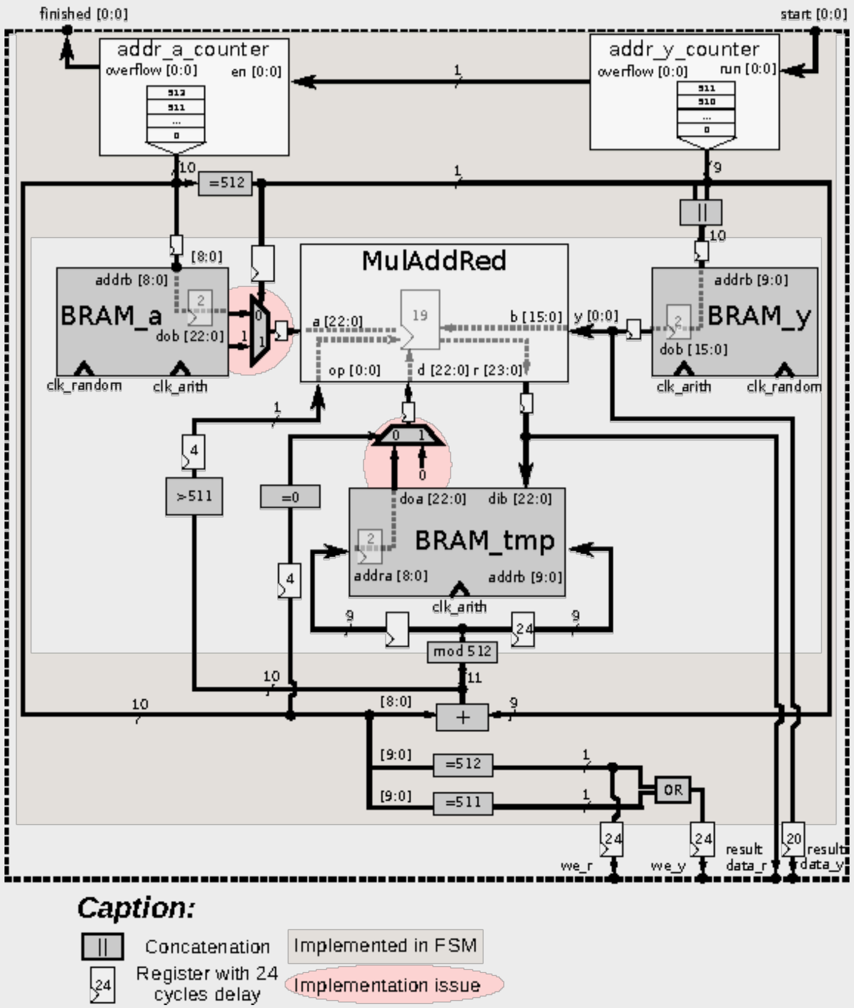


(1) Computation of ay_1+y_2 with multiple polynomial multipliers
(2) Further steps of the signing algorithm (Hash/Compression)

Next in Focus: Polynomial Multiplier

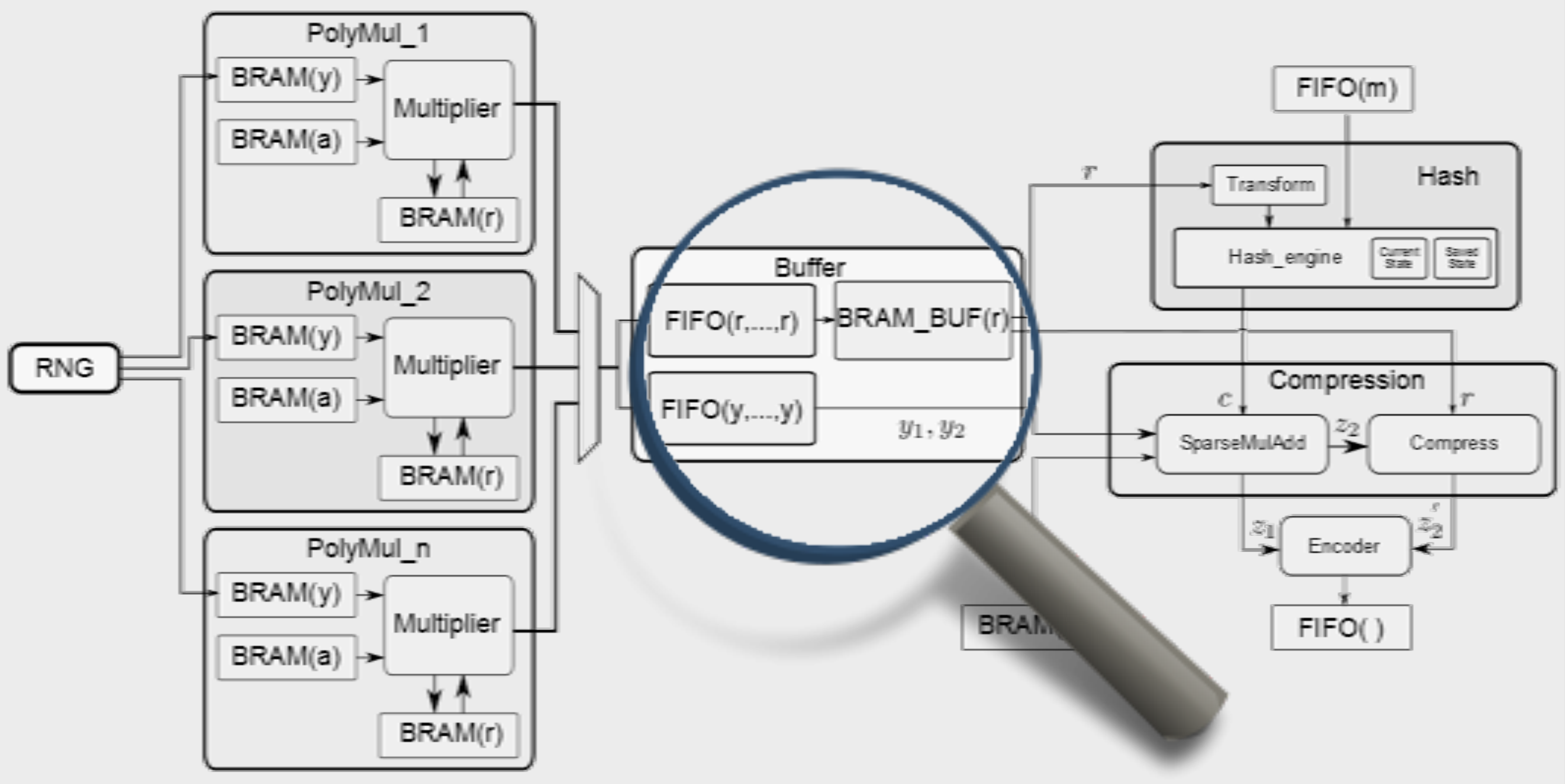


Implementation: Precomputation Core

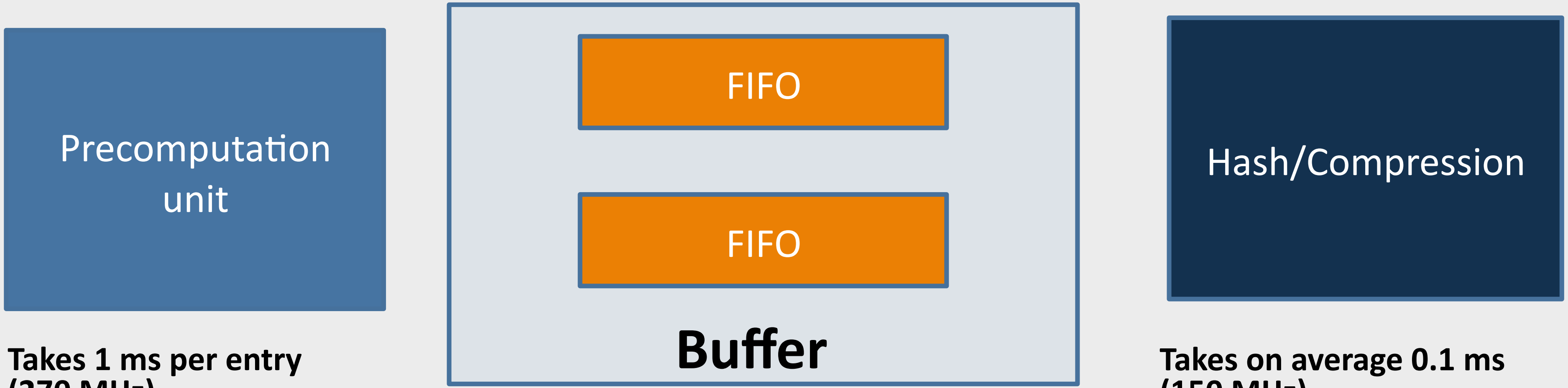


- Schoolbook multiplier with integrated adder to compute ay_1+y_2
- $n^2+n = 512^2+512 = 262656$ cycles
- High-frequency (270 MHz)
- 4 internal DSPs
- 23 pipeline stages
- **Can do approx. 1000 multiplications/s**

Next in Focus: Buffer Component



Implementation: Buffer

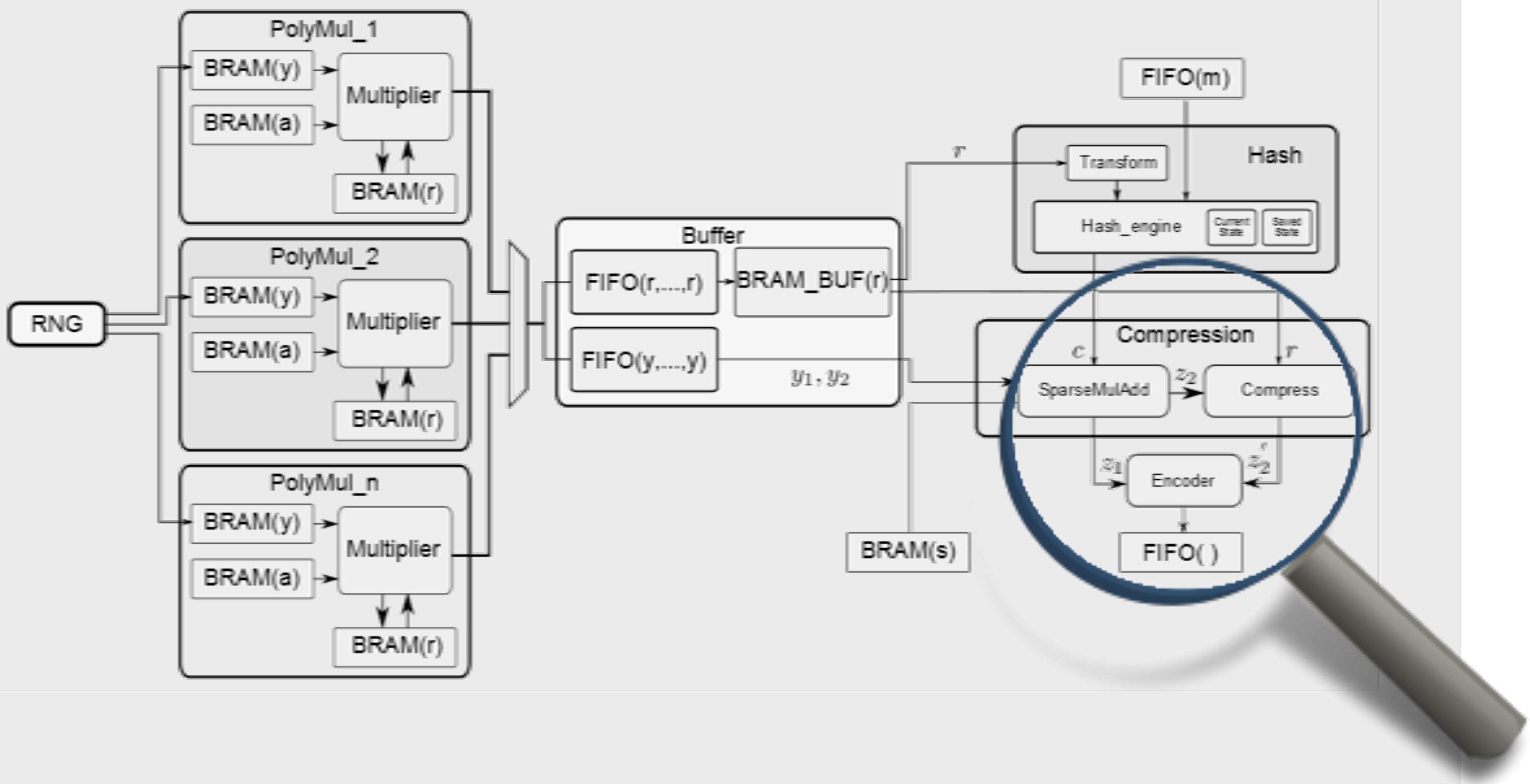


Takes 1 ms per entry (270 MHz)

Takes on average 0.1 ms (150 MHz)

- Values generated by the precomputation core can be buffered
  - Reduces the (non-deterministic) delay when a signature is requested (rejection sampling step)
- The final steps are 10x faster than the precomputation core

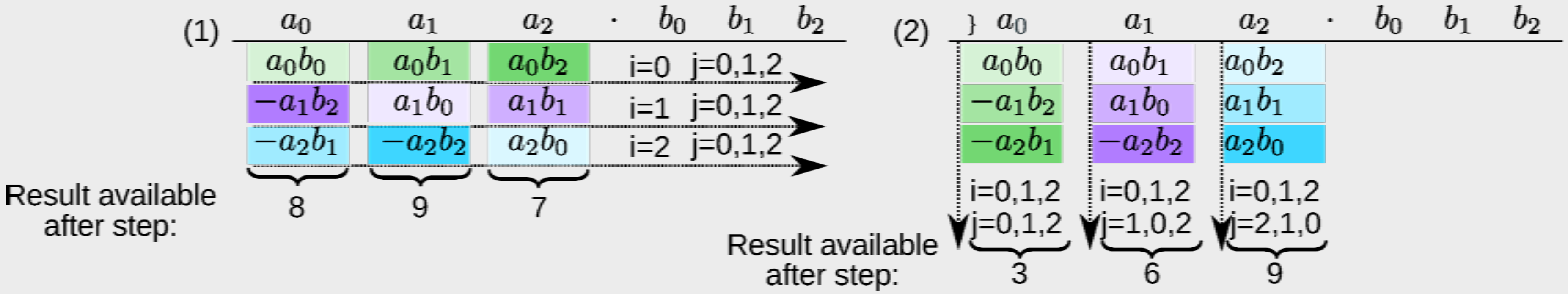
Next in Focus: Compression



Implementation: Compression

- Sparse Multiplication in $z_{1,2}=s_{1,2}c+y_{1,2}$
  - s_1 and s_2 have coefficients in the range [-1,1]
  - c has only 32 coefficients that are either -1 or 1
  - Comba-multiplication for early abort- test in place if $k \in R_{k-32}$
    - Product scanning vs. operand scanning: Reject at the first occurrence of an out of bound coefficient

$$c = \sum_{i=0}^{2} \sum_{j=0}^{2} a_i b_j x^{i+j} \mod (x^3+1).$$



Agenda

- Introduction
- Proposed Scheme
- FPGA Implementation
- **Results**
- Future Work

Results: Performance

- Target hardware: Spartan 6/Virtex 6

	Aspect	Spartan 6 LX16	Spartan 6 LX100	Virtex 6 LX130
Signing	Engines/Multiplier	1/7	4/9	9/8
	Total Multipliers	7	36	72
	Max. freq. domain (1)	270 MHz	250 MHz	416 MHz
	Max. freq. domain (2)	162 MHz	154 MHz	204 MHz
	Throughput σ/s	931	4284	12627
Verification	Independent engines	2	14	20
	Max. frequency domain (1)	272 MHz	273 MHz	402 MHz
	Max. frequency domain (2)	158 MHz	103 MHz	156 MHz
	Throughput σ/s	998	7015	14580

Results: Resource Consumption

Operation	Algorithm	Device	Resources	Ops/s
Our work	-	XC6SLX16	7465 LUTs/ 28 DSPs/ 29.5 BRAMs	931
Our work	-	XC6SLX100	30854 LUTs/ 144 DSPs/ 138 BRAMs	4284
Our work	-	XC6VLX130	67027 LUTs/ 216 DSPs/ 234 BRAMs	12627
RSA Signature [39]	RSA-1024; private key	XC4VFX12-10	3937 LS/ 17 DSPs	548
ECDSA [15]	NIST-P224; point mult.	XC4VFX12-12	1580 LS/ 26 DSPs	2,739
ECDSA [1]	NIST-B163; point mult.	XC2V2000	8300 LUTs/ 7 BRAMs	24,390
UOV-Signature [5]	UOV(60,20)	XC5VLX50-3	13437 LUTs	170,940

Outline

- Introduction
- Proposed Scheme
- Implementation
- Results
- **Future Work**

Future Work and Conclusion

Conclusion

- Practical, fast, scalable and area efficient implementation of lattice-based signature scheme on FPGAs
- Follow up work: *Towards Efficient Arithmetic for Lattice-Based Cryptography on Reconfigurable Hardware*, Thomas Pöppelmann and Tim Güneysu, Latincrypt 2012, to appear

Future Work

- Lightweight/low-cost resource sharing implementation
- Consideration of different architectures (uC, PC, ARM)
- Side-channel evaluation

RUHR-UNIVERSITÄT BOCHUM

RUB

Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems

CHES 2012, Leuven, Belgium

Tim Güneysu[1], Vadim Lyubashevsky[2] and Thomas Pöppelmann[1]

[1] Horst Görtz Institute for IT-Security, Ruhr-University Bochum, Germany

[2] INRIA / ENS, Paris

12.09.2012

Thank You for Your Attention!
Any Questions?