On the Design of Hardware Building Blocks for Modern Lattice-Based Encryption Schemes

Norman Göttert, **Thomas Feller**, Michael Schneider,
Johannes Buchmann, and Sorin A. Huss

Motivation

- Alternative security background required
- Current systems may be broken by quantum computers
- No sub-exponential attacks on commodity computers

→ Learning With Errors (LWE)
[LP11] Lindner, Peikert CT-RSA 2011

Contributions

CASED

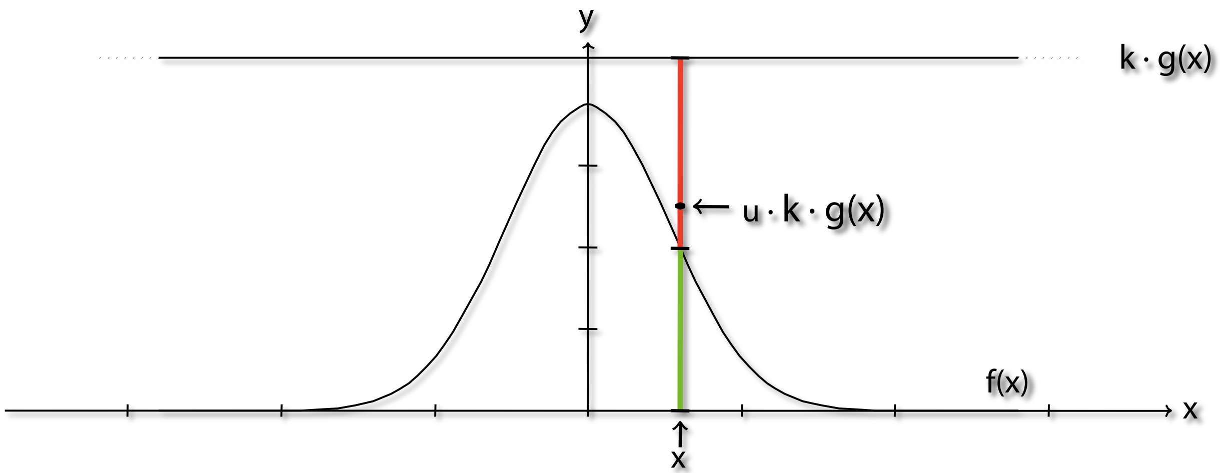
- Implementation of three variants
  - LWE-Matrix and LWE-Polynomial in software
  - Polynomial variant in Hardware
- Fundamental Building Blocks for Lattice-Based Cryptography
- Performance evaluation

LWE Basics

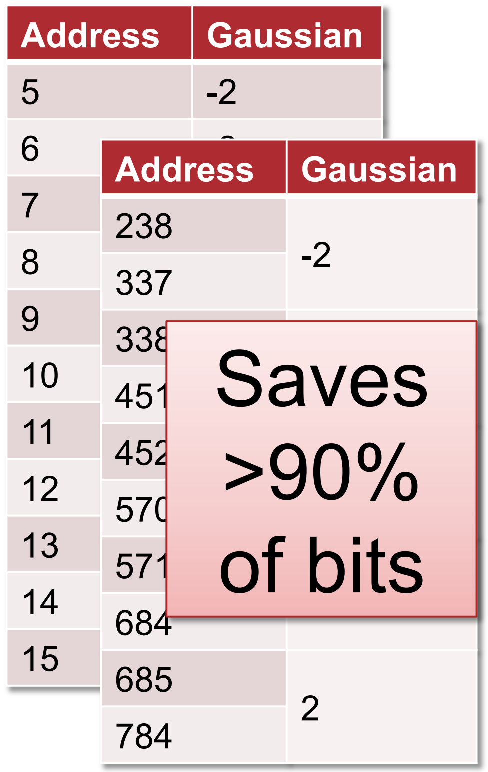
- Strong security proofs for lattice-based crypto exist
- Encryption introduces Gaussian distributed errors
- Decoding works if errors are below threshold
- Error tolerant message encoding

Gaussian Error Sampling

- Software: Rejection Sampling
- Hardware: Look-Up-Table



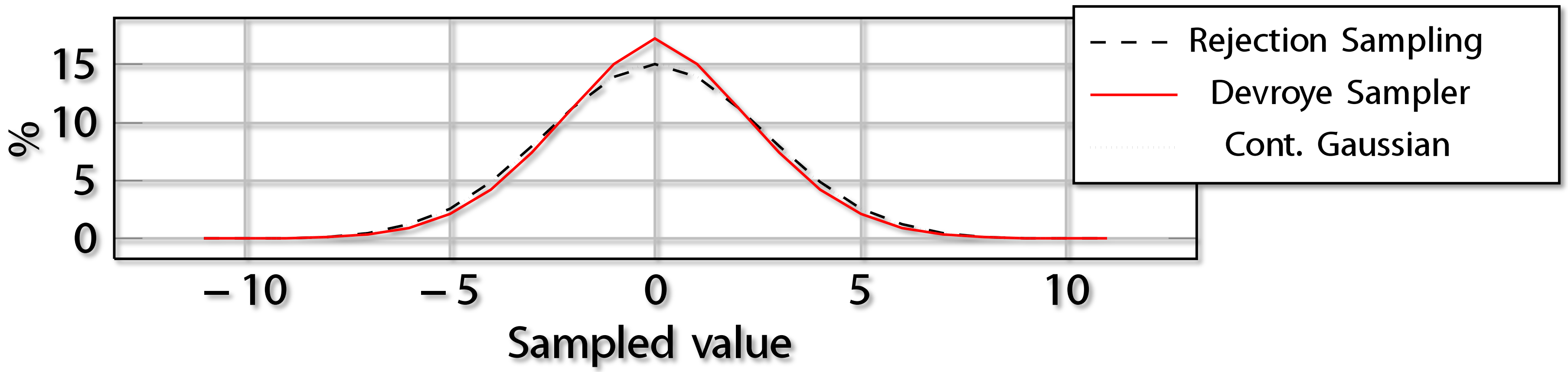
Rejection Sampling.



Address-based

Gaussian Error Sampling

- Rejection Sampling performance
- Devroye Sampler



Histogram of samples.

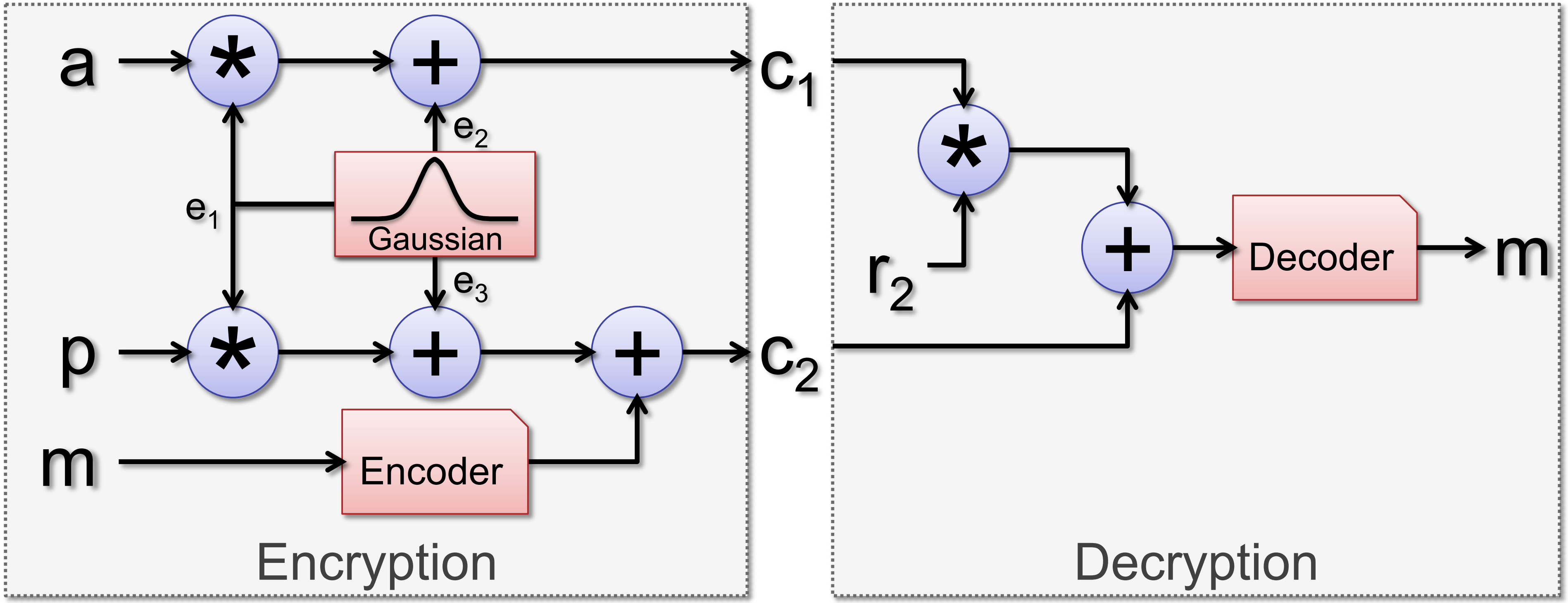
Preliminaries

Public Key $(\mathbf{a},\mathbf{p})$ | Private Key $\mathbf{r}_2$

- $\mathbf{a} \in \mathbb{Z}_q[X]/\langle \mathbf{f}(\mathbf{x}) \rangle$ chosen uniformly at random
- Choose $r_1, r_2 \leftarrow \chi$ a Gaussian distribution
- $\mathbf{p} = \mathbf{r}_1 - \mathbf{a} \cdot \mathbf{r}_2$
- Decoding works if errors are below threshold

  $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \leftarrow \chi$ $\delta = \left| \mathbf{e}_1 \cdot \mathbf{r}_1 + \mathbf{e}_2 \cdot \mathbf{r}_2 + \mathbf{e}_3 \right| \leq \mathbf{t}$

LWE-based Cryptosystem



(a,p) – public key
m – message

r_2 – private key
(c_1,c_2) – ciphertext

Fast Fourier Transform-Based Polynomial Multiplication I

Algorithm:

```
A = FFT(a,ω_m)
B = FFT(b,ω_m)
for i=0 to 2n-1 do
C[i] = A[i] * B[i]
end
c = FFT^-1(C,ω_m)
return c
```

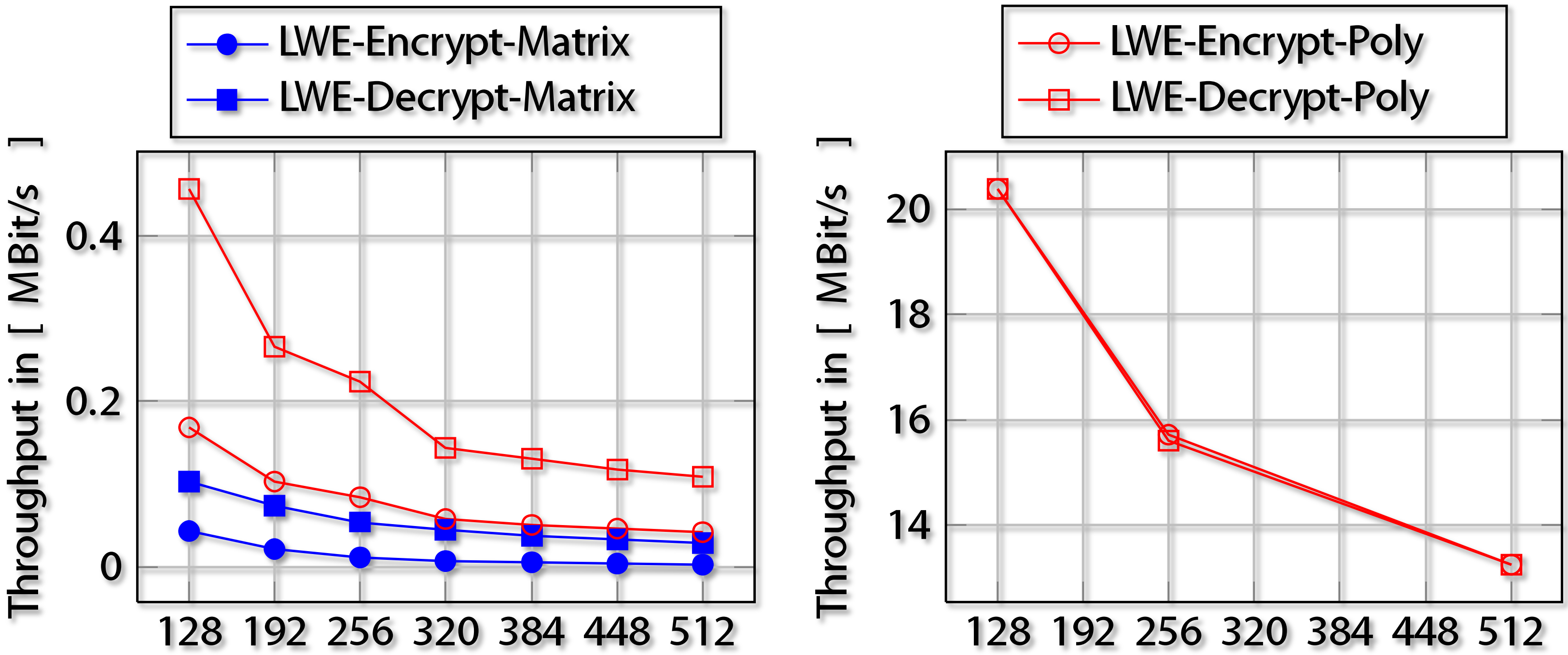
Fast Fourier Transform-Based Polynomial Multiplication II

- Polynomial multiplication using coefficients $O(n^2)$
- Point-value representation
  - Serial $O(n \log(n))$
  - Parallel $O(\log(n))$
- FFT-based approach can be utilized for other lattice-based encryption schemes

Message Expansion Factors

	LWE-Matrix		LWE-Polynomial		LWE-Hardware		[LP11]
n	cipher	cipher/plain	cipher	cipher/plain	cipher	cipher/plain	cipher/plain
128	512	32	512	32	384	24	22
192	640	40	768	48	---	---	30
256	768	48	1024	64	832	52	36
320	896	56	1280	80	---	---	42
384	1024	64	1536	96	---	---	---
448	1152	72	1792	112	---	---	---
512	1280	80	2048	128	1792	112	---

Evaluation Results I

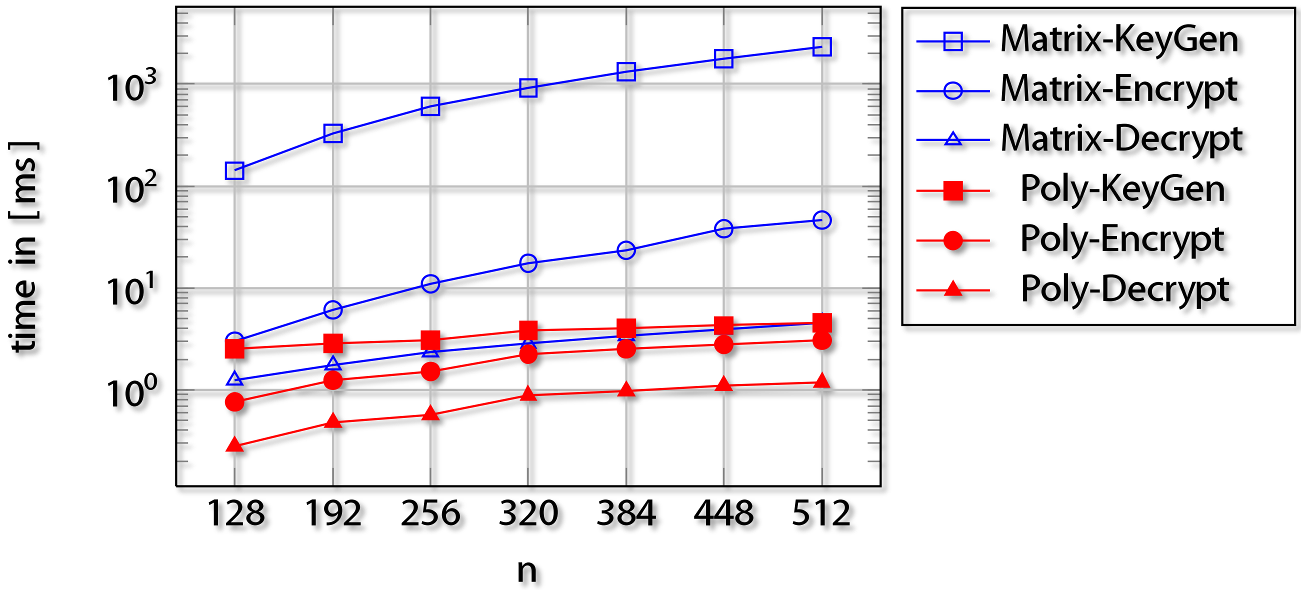


(a) Software (b) Hardware

Throughput

Evaluation Results II

CASED



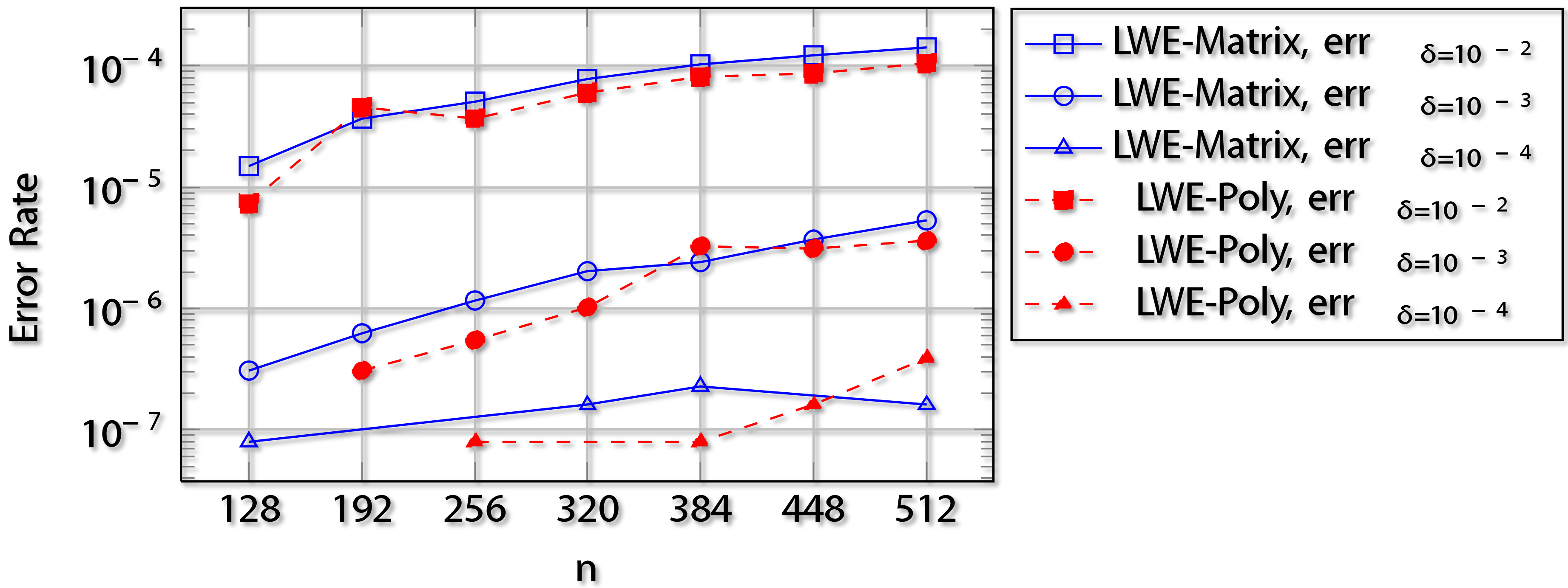
	KeyGen			Encrypt			Decrypt		
n	t_{matrix}	t_{poly}	t_{matrix}/t_{poly}	t_{matrix}	t_{poly}	t_{matrix}/t_{poly}	t_{matrix}	t_{poly}	t_{matrix}/t_{poly}
256	604.9	3.10	195.3	11.01	1.52	7.23	2.37	0.57	4.15
512	2338.5	4.53	516.5	46.05	3.06	15.04	4.52	1.18	3.84

LWE-Matrix vs. LWE-Polynomial

Execution times in *ms*

Evaluation Results III

CASED



Decryption Error Rate

Evaluation Results IV

	Virtex-6 LX240T		Virtex-7 2000T	
	n=128	%	n=512	%
# Register	37918	12	174757	7
# LUTs	64804	42	348204	28

KeyGen

	Virtex-6 LX240T		Virtex-7 2000T	
	n=128	%	n=512	%
# Register	64680	21	296207	12
# LUTs	131254	87	634893	51

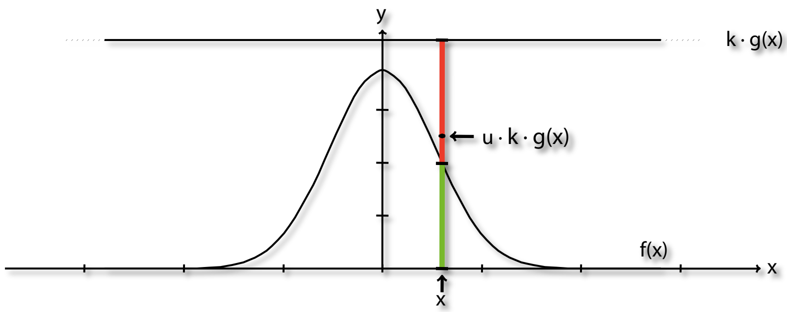
Encrypt

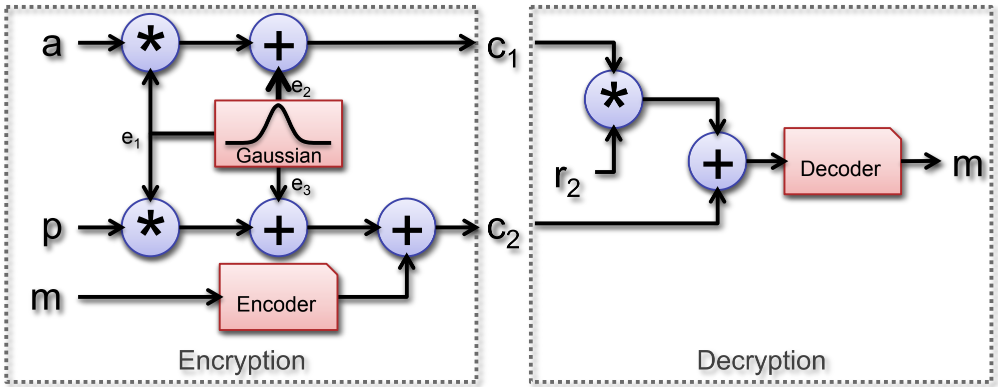
	Virtex-6 LX240T		Virtex-7 2000T	
	n=128	%	n=512	%
# Register	31884	10	134036	5
# LUTs	56311	37	260772	21

Decrypt

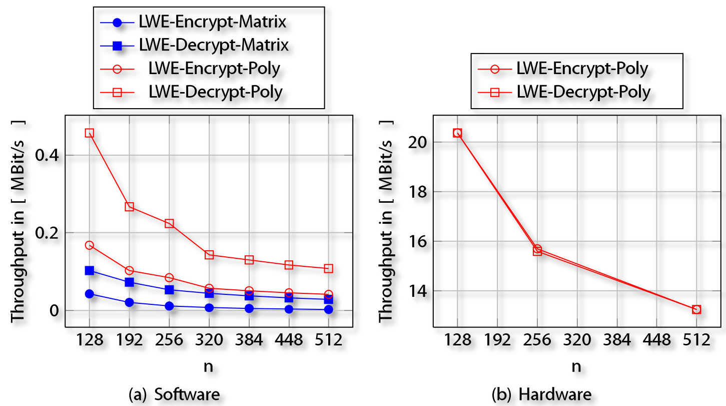
Hardware Resource Utilization

Conclusion





- Sampling of Gaussian distributed numbers
- HW/SW Implementation of LWE-Scheme
- Extensive evaluation



FFT-based polynomial multiplication as building block for Lattice-based Schemes

TOP SECRET

Message Encoding

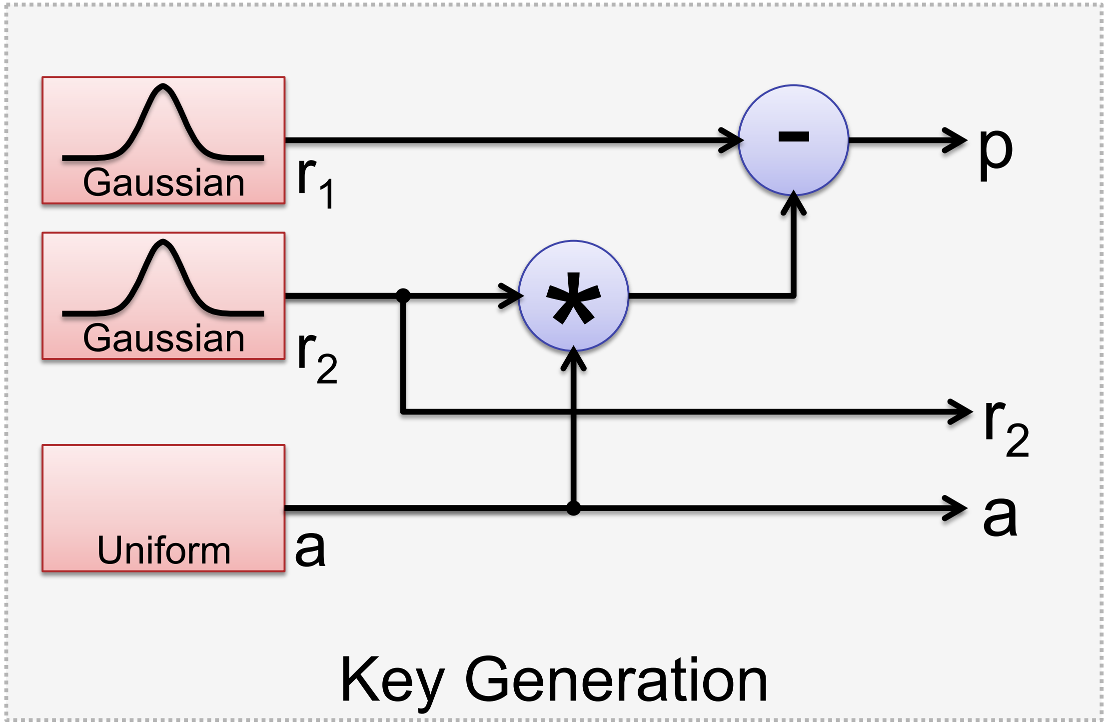
- Message $\mathbf{m} \in \Sigma$, alphabet $\Sigma = \{0,1\}$
- $\mathbf{m} \rightarrow \overline{\mathbf{m}} \in \mathbf{Z}_q^l$
- Encoder function:

$$\mathrm{encode}(\mathbf{m}_i) = \mathbf{m}_i \cdot \left\lfloor \frac{q}{2} \right\rfloor = \overline{\mathbf{m}_i}, \text{for } 0 \leq i < l$$

- Decoder function:

$$\mathrm{decode}(\overline{\mathbf{m}_i}) = \begin{cases} 0, \text{if } \overline{\mathbf{m}_i} \in \left[-\left\lfloor \frac{q}{2} \right\rfloor, \left\lfloor \frac{q}{2} \right\rfloor \right) \\ 1, \text{otherwise} \end{cases} = \mathbf{m}_i, \text{for } 0 \leq i < l$$

LWE Key Generation



(a,p) – public key

r_2 – private key