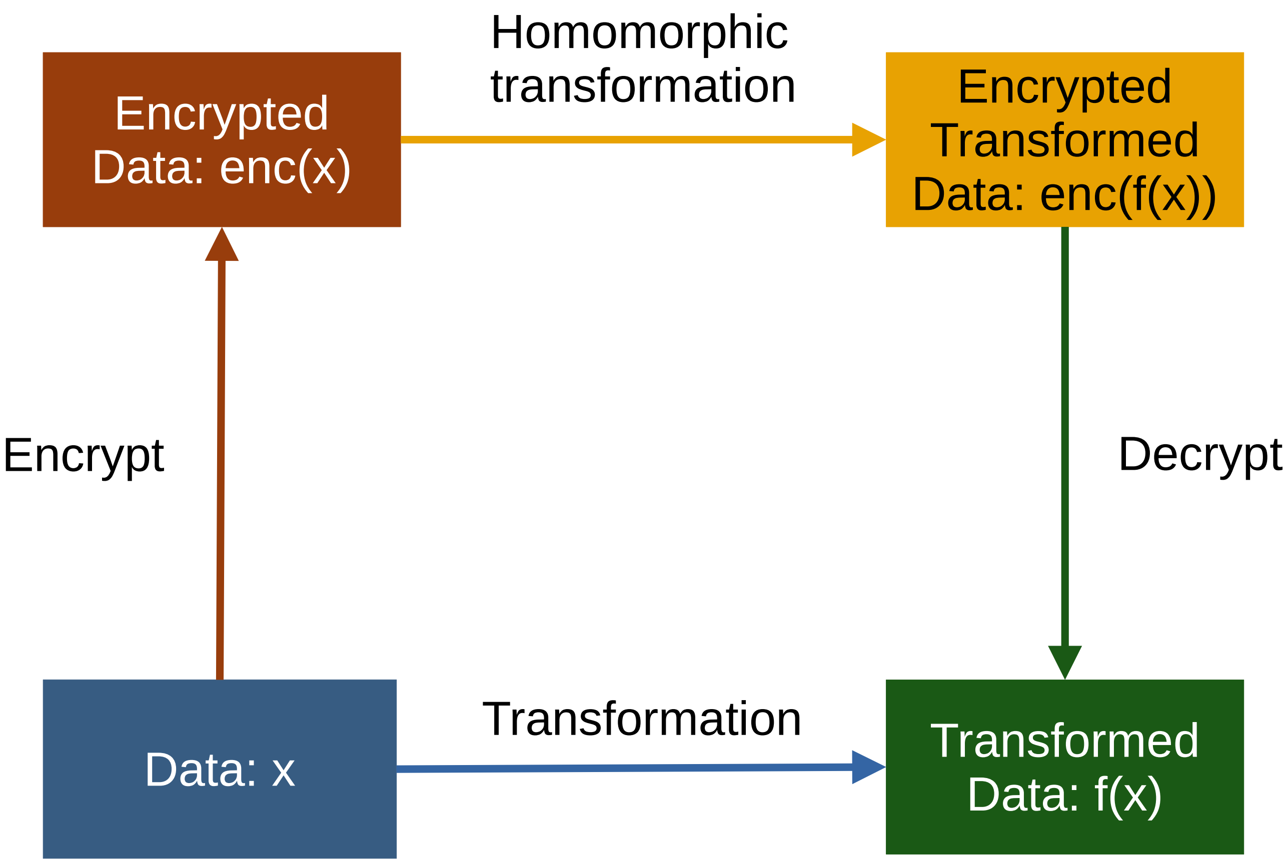
Homomorphic Post-Quantum Cryptography

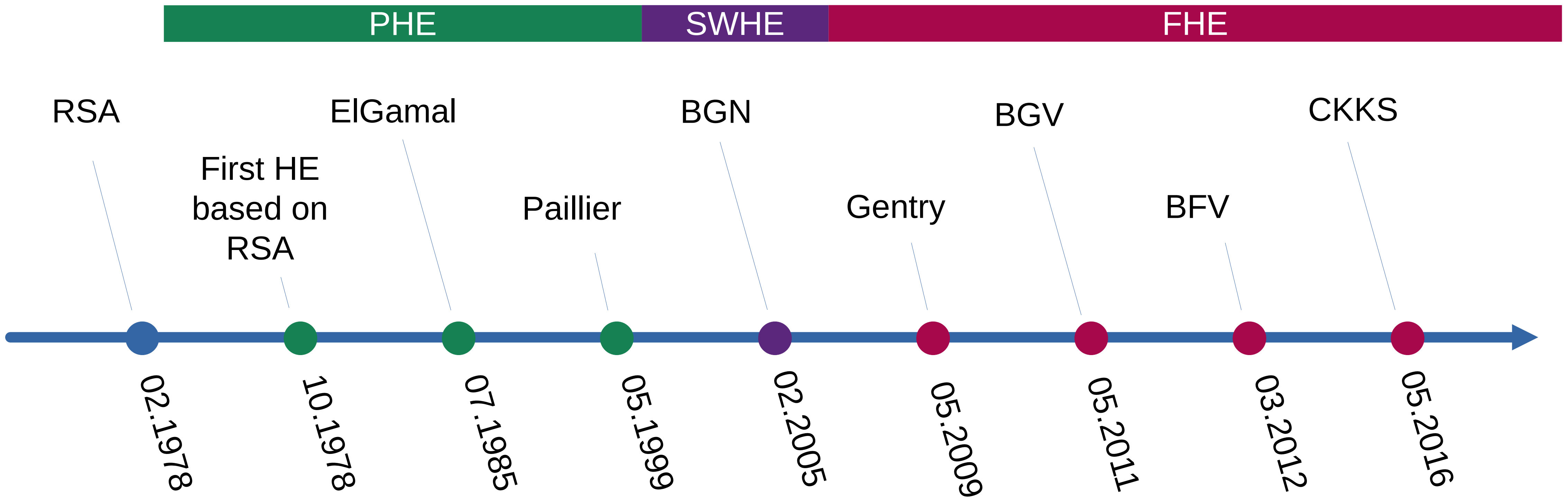
Evaluation of Module Learning with Error in Homomorphic Cryptography

HOCHSCHULE WISMAR
UNIVERSITY OF APPLIED SCIENCES TECHNOLOGY, BUSINESS AND DESIGN

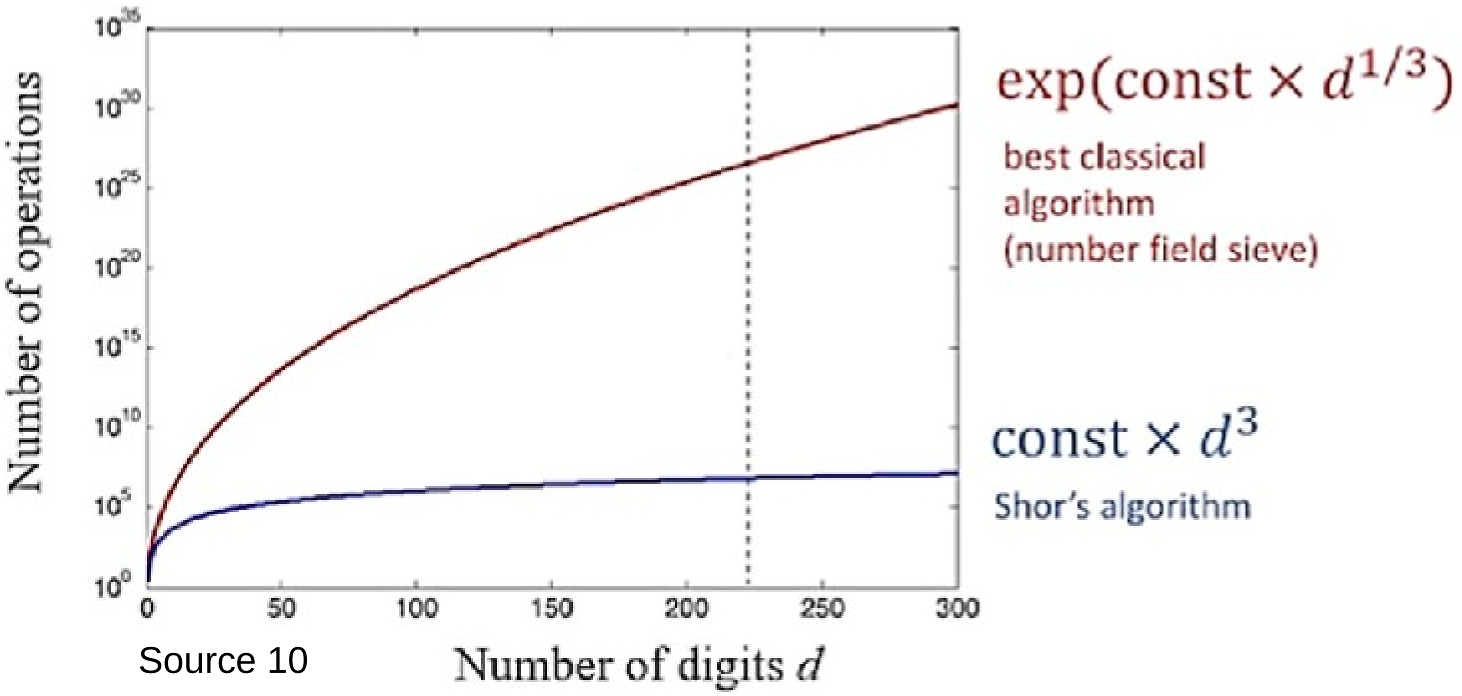
What is homomorphic encryption?



History of Homomorphic Encryption

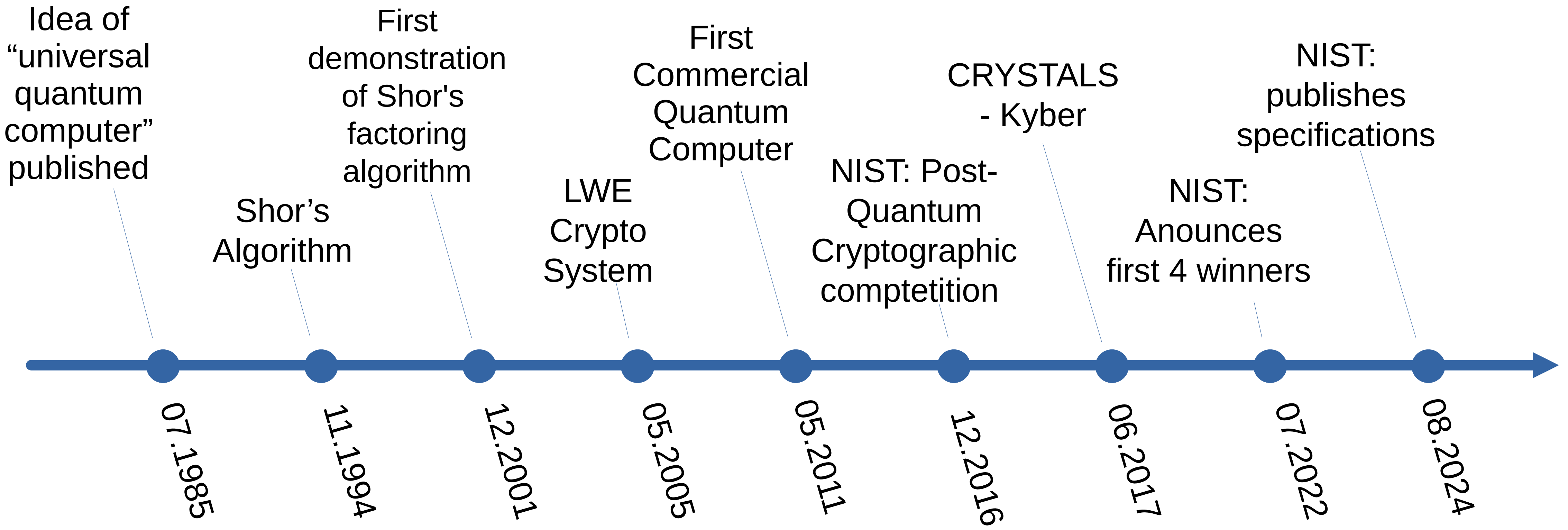


What is post-quantum cryptography?



Source 10

History of Post-Quantum Cryptography



What is LWE, R-LWE and M-LWE

LWE	→ Learning With Errors → Mathematical Framework for encryption schemes → Based on Multidimensional Lattices → Security depends on dimension n
R-LWE	→ uses Polynomials instead of Matrices/vectors → Security depends in polynomial rank ***d***
M-LWE	→ Uses Multidimensional Polynomials → Security depends on dimension ***n*** AND rank ***d***

$$
\begin{aligned}
\mathbf{A}_{11}\mathbf{s}_1 + \mathbf{A}_{12}\mathbf{s}_2 + \cdots + \mathbf{A}_{1m}\mathbf{s}_m + \mathbf{e}_1 &= \mathbf{b}_1 \\
\mathbf{A}_{21}\mathbf{s}_1 + \mathbf{A}_{22}\mathbf{s}_2 + \cdots + \mathbf{A}_{2m}\mathbf{s}_m + \mathbf{e}_2 &= \mathbf{b}_1 \\
\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots &= \vdots \\
\mathbf{A}_{n1}\mathbf{s}_1 + \mathbf{A}_{n2}\mathbf{s}_2 + \cdots + \mathbf{A}_{nm}s_m + \mathbf{e}_n &= \mathbf{b}_n
\end{aligned}
$$

$$\mathbf{A} \cdot \mathbf{s} + \mathbf{e} = \mathbf{b}$$

$$\mathbf{A} \in R_q^{n \times m}, \mathbf{s} \in R_q^m, \mathbf{e}, \mathbf{b} \in R_q^n$$

What is LWE, R-LWE and M-LWE

LWE

$$\mathbf{s} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \mathbf{A} = \begin{bmatrix} 56 & 77 \\ 29 & 59 \end{bmatrix} \mathbf{e} = \begin{bmatrix} 99 \\ 1 \end{bmatrix}$$

$$\begin{aligned} \mathbf{b} &= \mathbf{As} + \mathbf{e} \\ &= \begin{bmatrix} 56 & 77 \\ 29 & 59 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 99 \\ 1 \end{bmatrix} \\ &= 1 \cdot \begin{bmatrix} 56 \\ 29 \end{bmatrix} + 2 \cdot \begin{bmatrix} 77 \\ 59 \end{bmatrix} + \begin{bmatrix} 99 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 309 \\ 148 \end{bmatrix}_{100} \\ &= \begin{bmatrix} 9 \\ 48 \end{bmatrix} \end{aligned}$$

R-LWE

$$\begin{aligned} s &= 1 + 0x + 1x^2 \\ A &= 28 + 56x + 1x^2 \\ e &= 1 + -1x + 2x^2 = 1 + 99x + 2x^2 \\ b &= As + e \\ &= (28 + 56x + 1x^2) \cdot (1 + 0x + 1x^2) + (1 + 99x + 2x^2) \\ &= (28 + 28x^2) + (56x + 56x^3) + (1x^2 + 1x^4) + (1 + 99x + 2x^2) \\ &= 29 + 155x + 31x^2 + 56x^3 + 1x^4 \mod x^3 + 1 \\ &= 29 + 155x + 31x^2 - 56 - 1x \\ &= -27 + 154x + 31x^2 \mod 100 \\ &= 73 + 54x + 31x^2 \end{aligned}$$

M-LWE

$$\mathbf{s} = \begin{bmatrix} 2 + 1x + 0x^2 \\ 3 + 1x + 1x^2 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 27 + 2x + 43x^2 & 30 + 10x + 35x^2 \\ 91 + 34x + 50x^2 & 82 + 21x + 94x^2 \end{bmatrix}$$

$$\mathbf{e} = \begin{bmatrix} 1 + 1x + 2x^2 \\ -3 + 3x + 3x^2 = 97 + 3x + 3x^2 \end{bmatrix}$$

$$\begin{aligned} \mathbf{b} &= \mathbf{A}\mathbf{s} + \mathbf{e} \\ &= \begin{bmatrix} 27 + 2x + 43x^2 & 30 + 10x + 35x^2 \\ 91 + 34x + 50x^2 & 82 + 21x + 94x^2 \end{bmatrix} \cdot \begin{bmatrix} 2 + 1x + 0x^2 \\ 3 + 1x + 1x^2 \end{bmatrix} + \begin{bmatrix} 1 + 1x + 2x^2 \\ 97 + 3x + 3x^2 \end{bmatrix} \\ &= \begin{bmatrix} 56 + 56x + 233x^2 \\ 263 + 210x + 519x^2 \end{bmatrix} + \begin{bmatrix} 1 + 1x + 2x^2 \\ 97 + 3x + 3x^2 \end{bmatrix} \\ &= \begin{bmatrix} 57 + 57x + 235x^2 \\ 360 + 213x + 522x^2 \end{bmatrix}_{100} \\ &= \begin{bmatrix} 57 + 57x + 35x^2 \\ 60 + 13x + 22x^2 \end{bmatrix} \end{aligned}$$

Where are we today?

LWE and R-LWE based FHE

New encryption Standard based on M-LWE

Active research on hardware acceleration of M-LWE

Alot of cheap computing power (AI Hype)

An ongoing shift into cloud environments

→ Why is there no M-LWE based homomorphic encryption?

R-LWE vs M-LWE?

	R-LWE	M-LWE
Dimension	One big Polynomial	Multiple smaller Polynomials
Size Cost	Better, except Ciphertext	Worse, except Ciphertext
Time Cost	Faster, except decrypt	Slightly slower, except decrypt
Calculation Depth	Equal	

Size Cost in KB

	Source	n	d	q	q_b	sk	pk	rlk	ct
R-LWE	[32]		512	25601	15	0.96	1.92	7.68	1.92
M-LWE	[4]	3	256	7681	13	1.25	4.992	59.904	1.66

Time Cost

	Source	Addition	Decrypt	Encryption	KeyGen	Multiplication
R-LWE	[32]	0.000163	0.061521	0.125041	0.182526	0.473052
M-LWE	[4]	0.000176	0.041224	0.174293	0.696122	1.039662

Calculation Depth

	Addition						Multiplication					
q_b	13	15	20	32	64	128	13	15	20	32	64	128
R-LWE [4]	-	56	200	200	200	200	-	0	0	1	3	7
M-LWE [32]	7	-	200	200	200	200	0	-	0	1	3	7

Word Size

Word Size	→ natural unit of data, e.g 32 or 64 bit → reflects CPU structure, e.g. registers, memory lanes
HE & Word Size	→ one ciphertext represent one data point → polynomial rank ***d*** equals word size: R-LWE 512 bit M-LWE 256 bit → Each Data Point needs padding: Plaintext padded to Ciphertext space

Improving Ciphertext Size

Ciphertext growth

→ R-LWE: 64 TO 1920 bit: 30x
→ M-LWE: 64 TO 1660 bit: ~26x

	Source	n	d	q	q_b	sk	pk	rlk	ct
R-LWE	[32]		512	25601	15	0.96	1.92	7.68	1.92
M-LWE	[4]	3	256	7681	13	1.25	4.992	59.904	1.66

HE & Ciphertext

→ every data point needs to be transformed into ciphertext
→ by decreasing the growth rate, a lot of storage can be saved
→ the other Keys size is less important

M-LWE Improvement

→ Space depended on polynomial rank d
→ M-LWE can decrease d by increasing n
→ Ciphertext size can be decreased

	R-LWE	M-LWE
ct	$\mathbb{Z}_q^d \times \mathbb{Z}_q^d$	$\mathbb{Z}_q^{n \times d} \times \mathbb{Z}_q^d$

Improving Performance

Existing Optimization	→ hardware near languages → improved Algorithms → Vectorization & Parallel Computing (GPUs)
Active Research	→ for improving CRYSTALS-Kyber → for improving Matrix calculation (AI) → Ongoing FHE improvements
M-LWE	→ Already near R-LWE → A lot of options for improving performance

Future of M-LWE based HE

Just begun	→ very few concepts of M-LWE in FHE → not older than a year
A lot to research	→ Many open questions → Improvements need to be tested with more research
Big Impact	→ FHE is next big think since a decade → If practical feasible, big impact on cloud

Sources

1. R. L. Rivest, A. Shamir, and L. Adleman. “A method for obtaining digital signatures and public-key cryptosystems.”
2. R L Rivest, L Adleman, and M L Dertouzos. “On Data Banks and Privacy Homomorphisms.”
3. T. Elgamal. “A public key cryptosystem and a signature scheme based on discrete logarithms.”
4. Pascal Paillier. “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes.”
5. Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. “Evaluating 2-DNF Formulas on Ciphertexts.”
6. Craig Gentry. “A fully homomorphic encryption scheme.”
7. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. “Fully Homomorphic Encryption without Bootstrapping.”
8. Junfeng Fan and Frederik Vercauteren. “Somewhat Practical Fully Homomorphic Encryption”
9. Jung Hee Cheon and Andrey Kim and Miran Kim and Yongsoo Song: “Homomorphic Encryption for Arithmetic of Approximate Numbers”
10. Kumar, Manish. “Post-quantum cryptography Algorithm's standardization and performance analysis.”
11. https://thequantuminsider.com/2020/05/26/history-of-quantum-computing/
12. P.W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring.”
13. Oded Regev. “On lattices, learning with errors, random linear codes, and cryptography.”
14. https://csrc.nist.gov/news/2016/public-key-post-quantum-cryptographic-algorithms
15. Joppe Bos et al. “CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM”
16. https://csrc.nist.gov/news/2023/three-draft-fips-for-post-quantum-cryptography

Thanks for listening :)

HOCHSCHULE WISMAR
UNIVERSITY OF APPLIED SCIENCES TECHNOLOGY, BUSINESS AND DESIGN