QAP-based HE

蘇正耀、蔡明忠

NVIDIA Joint-Lab：NCHC Tech Sharing Workshop，July 6, 2021

Homomorphic Encryption

Homomorphic Encryption (HE) is commonly known as the “Holy Grail of Encryption”, which is a method that permits users to perform computations on encrypted data without first decrypting it. The result of computation remains in encrypted form, and can be revealed by the owner of the secret key.

$$A \ast B = C$$

Encoding $\quad \bar{A} \;\bar{\ast}\; \bar{B} \neq \bar{C}$ $\quad$ Fully HE via QAPHE

$$\bar{A} \;\bar{\ast}\; \bar{B} \rightarrow \bar{C}$$

Approaching, approximated, solutions with expensive costs in current schemes

Current Schemes

HE in Latticed-based Cryptosystems, adopted by DARPA, IBM, Microsoft, … …

Craig Gentry, *Fully Homomorphic Encryption Using Ideal Lattices*, in the 41st ACM Symposium on Theory of Computing (STOC), 2009

QAP-based HE

Given a quantum code $[n,\ k,\ \mathrm{C}]$, $n > k$, and a pair of keys

$$\begin{cases} \text{PublicKey} = (Q_{en},\ \mathrm{E}) \\ \text{PrivateKey} = \mathrm{A}^{\dagger} P^{\dagger} \end{cases}$$

Q_{en} : An n-qubit encoding operator

E : A correctable error set

M : The k-qubit arithmetic operation

$P,\ \mathrm{A}$: n-qubit operators of mix-up

$$|\boldsymbol{x}\rangle \xrightarrow[(Q_{en},\ \mathrm{E})]{\text{Encryption}} |\boldsymbol{c}\rangle = EQ_{en}|\mathbf{0}\rangle \otimes |\boldsymbol{x}\rangle \xrightarrow[\text{Evaluation of } M|\boldsymbol{x}\rangle]{\text{Computation}} U_{en}|\boldsymbol{c}\rangle \xrightarrow[\mathrm{A}^{\dagger}P^{\dagger}]{\text{Decryption}} \mathrm{A}^{\dagger}P^{\dagger}U_{en}|\boldsymbol{c}\rangle = |\mathbf{0}\rangle \otimes M|\boldsymbol{x}\rangle$$

k-qubit plaintext

n-qubit ciphertext

$E \in \mathrm{E}$

Code-based Cryptography (a.k.a. Post-Quantum)

Given a linear code $[n,\ k]$, $n > k$, and a pair of keys

$$\begin{cases} \text{PublicKey} = (G = SGP,\ t) \\ \text{PrivateKey} = (S,\ G,\ P) \end{cases}$$

G : The $k \times n$ generator matrix of $[n,\ k]$

S : A $k \times k$ invertible matrix

P : An $n \times n$ permutation matrix

$$\boldsymbol{x} \in Z_2^k \xrightarrow[(G,\ t)]{\text{Encryption}} \boldsymbol{c} = \boldsymbol{x}\ G + \boldsymbol{e} \in Z_2^n \xrightarrow[(S,\ G,\ P)]{\text{Decryption}} \boldsymbol{c}\ P^{-1} = \boldsymbol{x}\ SG + \boldsymbol{e}\ P^{-1}$$

k-bit plaintext

n-bit ciphertext

$\boldsymbol{e}$: An error of weight t

$$\downarrow$$

$$\boldsymbol{y} = \boldsymbol{x}\ S \qquad \text{correction}$$

$$\downarrow$$

$$\boldsymbol{x} = \boldsymbol{y}\ S^{-1}$$

QAP - based HE

Given a quantum code $[n,\ k,\ \mathrm{C}]$, $n > k$, and a pair of keys

$$\begin{cases} \text{PublicKey} = (Q_{en},\ \mathrm{E}) \\ \text{PrivateKey} = \mathrm{A}^{\dagger} P^{\dagger} \end{cases}$$

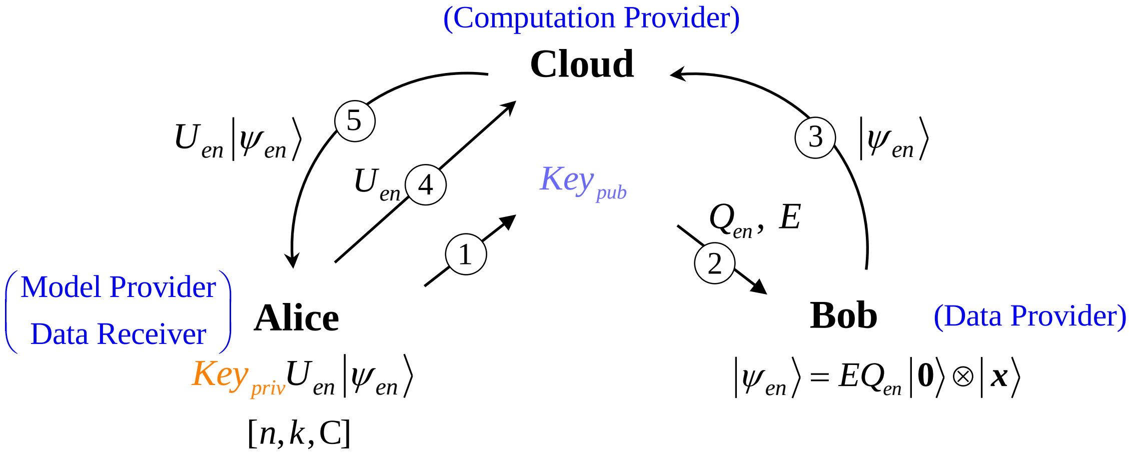
Q_{en} : An n-qubit encoding operator

E : A correctable error set

M : The k-qubit arithmetic operation

$P,\ \mathrm{A}$: n-qubit operators of mix-up

$$\underset{k\text{-qubit plaintext}}{|\boldsymbol{x}\rangle} \xrightarrow[(Q_{en},\ \mathrm{E})]{\text{Encryption}} \underset{\substack{n\text{-qubit ciphertext} \\ E \in \mathrm{E}}}{|\boldsymbol{c}\rangle = EQ_{en}|\boldsymbol{0}\rangle \otimes |\boldsymbol{x}\rangle} \xrightarrow[\text{Evaluation of } M|\boldsymbol{x}\rangle]{\text{Computation}} U_{en}|\boldsymbol{c}\rangle \xrightarrow[\mathrm{A}^{\dagger}P^{\dagger}]{\text{Decryption}} \mathrm{A}^{\dagger}P^{\dagger}U_{en}|\boldsymbol{c}\rangle = |\boldsymbol{0}\rangle \otimes M|\boldsymbol{x}\rangle$$



1. Alice generates the public-key $Key_{pub} = (Q_{en},\ \mathrm{E})$ and the private key $Key_{priv} = \mathrm{A}^{\dagger}P^{\dagger}$
2. Bob obtains $|\psi_{en}\rangle = EQ_{en}|0\rangle \otimes |\boldsymbol{x}\rangle$ via Key_{pub}
3. Bob sends $|\psi_{en}\rangle$ to the cloud
4. Alice sends the computation instruction of $U_{ed} = PAM\ Q^{\dagger}$ (mix-up)
5. Alice acquires the evaluation $U_{en}|\psi_{en}\rangle$ from the cloud, and the recovers $\mathrm{A}^{\dagger}P^{\dagger}U_{en}|\psi_{en}\rangle$ via decryption

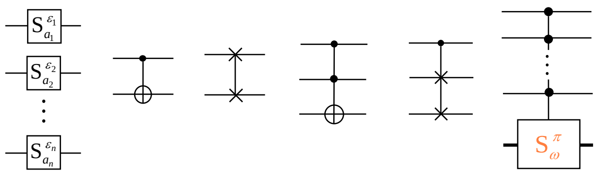
QAP-based HE

Given a quantum code $[n,\ k,\ \mathrm{C}],\ n > k,$ and a pair of keys

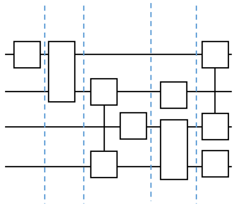
$$\begin{cases} \textbf{PublicKey} = (Q_{en},\ \mathrm{E}) \\ \textbf{PrivateKey} = \mathrm{A}^{\dagger} P^{\dagger} \end{cases}$$

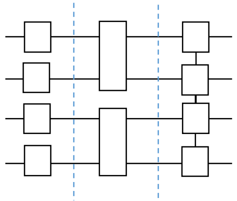
$$|\boldsymbol{x}\rangle \xrightarrow[(Q_{en},\ \mathrm{E})]{\textbf{Encryption}} |\boldsymbol{c}\rangle = E Q_{en} |\boldsymbol{0}\rangle \otimes |\boldsymbol{x}\rangle \xrightarrow[\textbf{Evaluation of } M|\boldsymbol{x}\rangle]{\textbf{Computation}} U_{en}|\boldsymbol{c}\rangle \xrightarrow[\mathrm{A}^{\dagger}P^{\dagger}]{\textbf{Decryption}} \mathrm{A}^{\dagger}P^{\dagger}U_{en}|\boldsymbol{c}\rangle = |\boldsymbol{0}\rangle \otimes M|\boldsymbol{x}\rangle$$

$$U_{en} = P\mathrm{A}\ \mathrm{M}\ B\ Q_{en}^{\dagger} \qquad \mathrm{M} = I \otimes M$$

$=$ 

$$= (PV^{\dagger}P^{\dagger})(PV\ \mathrm{A}\ \mathrm{M}\ V^{\dagger}P^{\dagger})(PV\ B\ V^{\dagger}P^{\dagger})(PV\ Q_{en}^{\dagger}\ V^{\dagger}P^{\dagger})(PV)$$

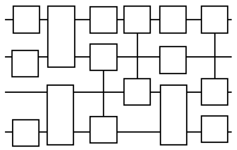


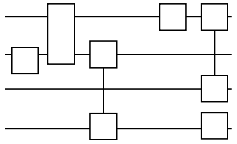
$=$ 

Exact computation

Blind computation

Problem-Dependent Optimizations of Circuits



$=$ 

Hamming Code

[5, 2]

C_{000}	00000	01011	10110	11101
C_{001}	00001	01010	10111	11100
C_{010}	00010	01001	10100	11111
C_{011}	01000	00011	11110	10101
C_{100}	00100	01111	10010	11001
C_{101}	00101	01110	10011	11000
C_{110}	10000	11011	00110	01101
C_{111}	10001	11010	00111	01100

The generator matrix

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

The parity-check matrix

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

$$GH^T = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The dual code [5, 3]

$$\mathrm{C}^{\perp} = \{ \ 00000,\ 10100,\ 11010,\ 01001,\ 01110,\ 11101,\ 10011,\ 00111 \ \}.$$

The relation between C and $\mathrm{C}^{\perp}$:

$$\forall u \in \mathrm{C},\ w \in \mathrm{C}^{\perp},\ \langle u | w \rangle = 0,$$

$$H|u\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad G|w\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Stabilizer Code [[5, 1]] with Stabilizer $C \equiv W(B_{0000},\ S_0^0)$

Hamming Code [5, 1]

Quantization

$$W(B_{0000},\ S_0^0)\quad S^{00000}_{00000}, S^{10111}_{00011}, S^{11000}_{00101}, S^{01111}_{00110}, S^{00110}_{01001}, S^{10001}_{01010}, S^{11110}_{01101}, S^{01001}_{01111}, S^{11011}_{10001}, S^{01100}_{10010}, S^{00011}_{10100}, S^{10100}_{10111}, S^{11101}_{11001}, S^{01010}_{11011}, S^{00101}_{11100}, S^{10100}_{10110}$$

$$W(B_{0000},\ S_0^1)\quad S^{11111}_{00000}, S^{01000}_{00011}, S^{00111}_{00101}, S^{10000}_{00110}, S^{11001}_{01001}, S^{01110}_{01010}, S^{00001}_{01101}, S^{10110}_{01111}, S^{00100}_{10001}, S^{10011}_{10010}, S^{11100}_{10100}, S^{01011}_{10111}, S^{00010}_{11001}, S^{10101}_{11011}, S^{11010}_{11100}, S^{01011}_{10110}$$

$$W(B_{0000},\ S_1^0)\quad S^{00000}_{11111}, S^{10111}_{11100}, S^{11000}_{11010}, S^{01111}_{11001}, \cdots \ \ldots\ldots$$

$$W(B_{0000},\ S_1^1)\quad S^{11111}_{11111}, S^{01000}_{11100}, S^{00111}_{11010}, S^{10000}_{11001}, \ \ldots\ldots$$

$$W(B_{0001},\ S_0^0)\quad S^{00000}_{00000}, S^{10111}_{00011}, S^{11000}_{00101}, S^{01111}_{00110}, S^{00110}_{01001}, S^{10001}_{01010}, S^{11110}_{01101}, S^{01001}_{01111}, S^{11011}_{10001}, S^{01100}_{10010}, S^{00011}_{10100}, S^{10100}_{10111}, S^{11101}_{11001}, S^{01010}_{11011}, S^{00101}_{11100}, S^{10100}_{10110}$$

$$W(B_{0001},\ S_0^1)\quad S^{11111}_{00000}, S^{01000}_{00011}, S^{00111}_{00101}, S^{10000}_{00110}, S^{11001}_{01001}, S^{01110}_{01010}, S^{00001}_{01101}, S^{10110}_{01111}, S^{00100}_{10001}, S^{10011}_{10010}, S^{11100}_{10100}, S^{01011}_{10111}, S^{00010}_{11001}, S^{10101}_{11011}, S^{11010}_{11100}, S^{01011}_{10110}$$

$$W(B_{0001},\ S_1^0)\quad S^{00000}_{11111}, S^{10111}_{11100}, S^{11000}_{11010}, S^{01111}_{11001}, \cdots \ \ldots\ldots$$

$$W(B_{0001},\ S_1^1)\quad S^{11111}_{11111}, S^{01000}_{11100}, S^{00111}_{11010}, S^{10000}_{11001}, \ \ldots\ldots$$

$$\vdots$$

$$W(B_{1111},\ S_0^0)\quad S^{00000}_{00000}, S^{10111}_{00011}, S^{11000}_{00101}, S^{01111}_{00110}, \cdots \ \ldots\ldots$$

$$W(B_{1111},\ S_0^1)\quad S^{11111}_{00000}, S^{01000}_{00011}, S^{00111}_{00101}, S^{10000}_{00110}, \cdots \ \ldots$$

$$W(B_{1111},\ S_1^0)\quad S^{00000}_{11111}, S^{10111}_{11100}, S^{11000}_{11010}, S^{01111}_{11001}, \cdots \ \ldots\ldots$$

$$W(B_{1111},\ S_1^1)\quad S^{11111}_{11111}, S^{01000}_{11100}, S^{00111}_{11010}, S^{10000}_{11001}, \ \ldots\ldots$$

All Versions of Encodings.

All k-th Maximal Bi-Subalgebras of All Cartan subalgebras.

$$\vdots$$

$$Q\ C\ Q^{\dagger} = C.$$

Find All k-th Maximal Subgroups of $\mathbb{Z}_2^n$.

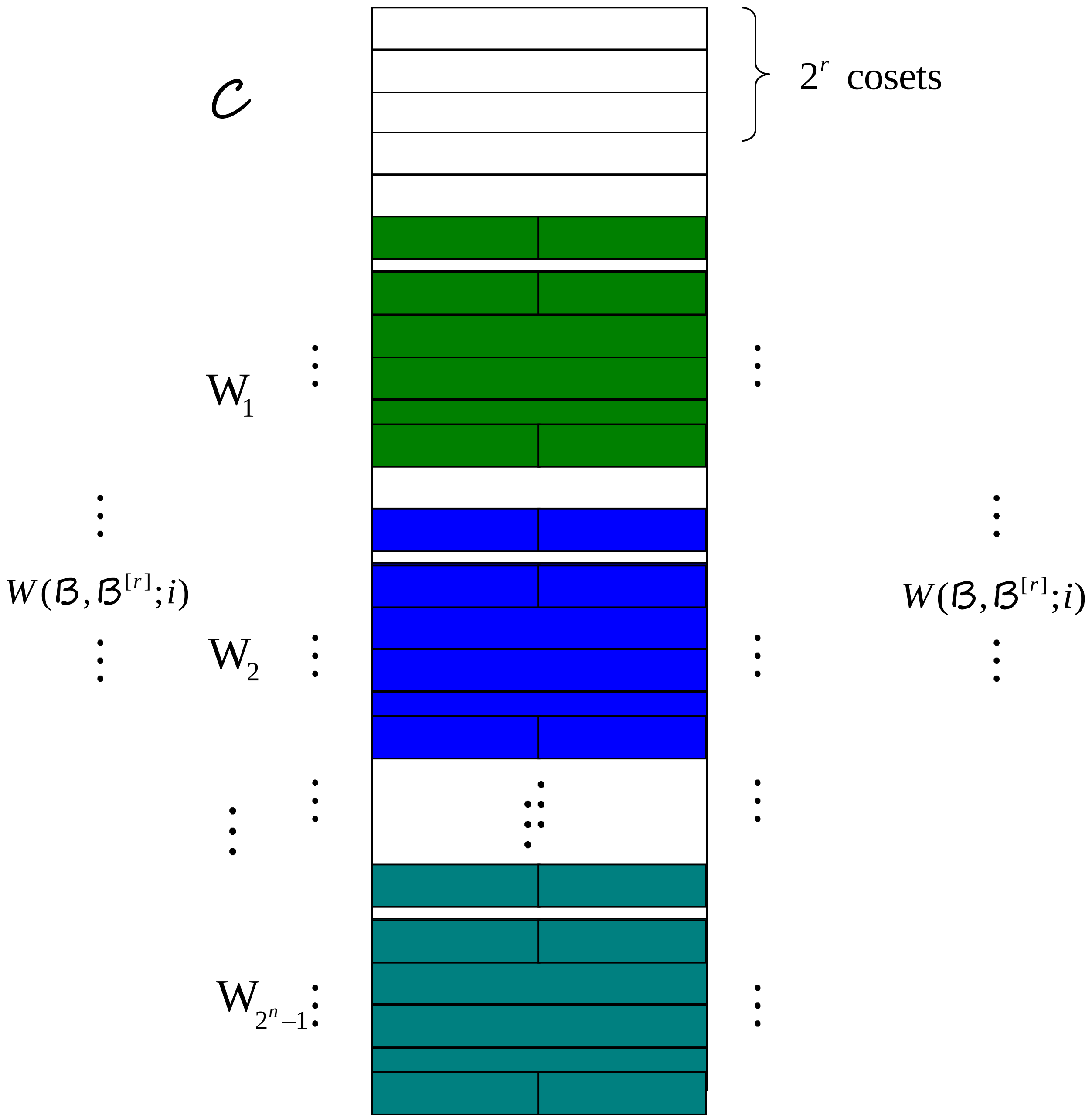
$$U_{ed}\ E\ U_{ed}^{\dagger} = E'.$$

C_{0000}	00000	11111
C_{0001}	00001	11110
C_{0010}	00010	11101
C_{0011}	00011	11100
C_{0100}	00100	11011
C_{0101}	00101	11010
C_{0110}	00110	11001
C_{0111}	00111	11000
C_{1000}	01000	10111
C_{1001}	01001	10110
C_{1010}	01010	10101
C_{1011}	01011	10100
C_{1100}	01100	10011
C_{1101}	01101	10010
C_{1110}	01110	10001
C_{1111}	01111	10000

Quotient Algebra Partition (**QAP**)

is an algebraic structure over $su(2^n)$ consisting of abelian subspaces closed under the commutation relation or the tri-addition.

Every quantum code $[n,\ k,\ C]$ is a QAP.



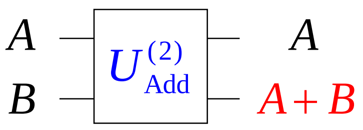
Quantum Adder

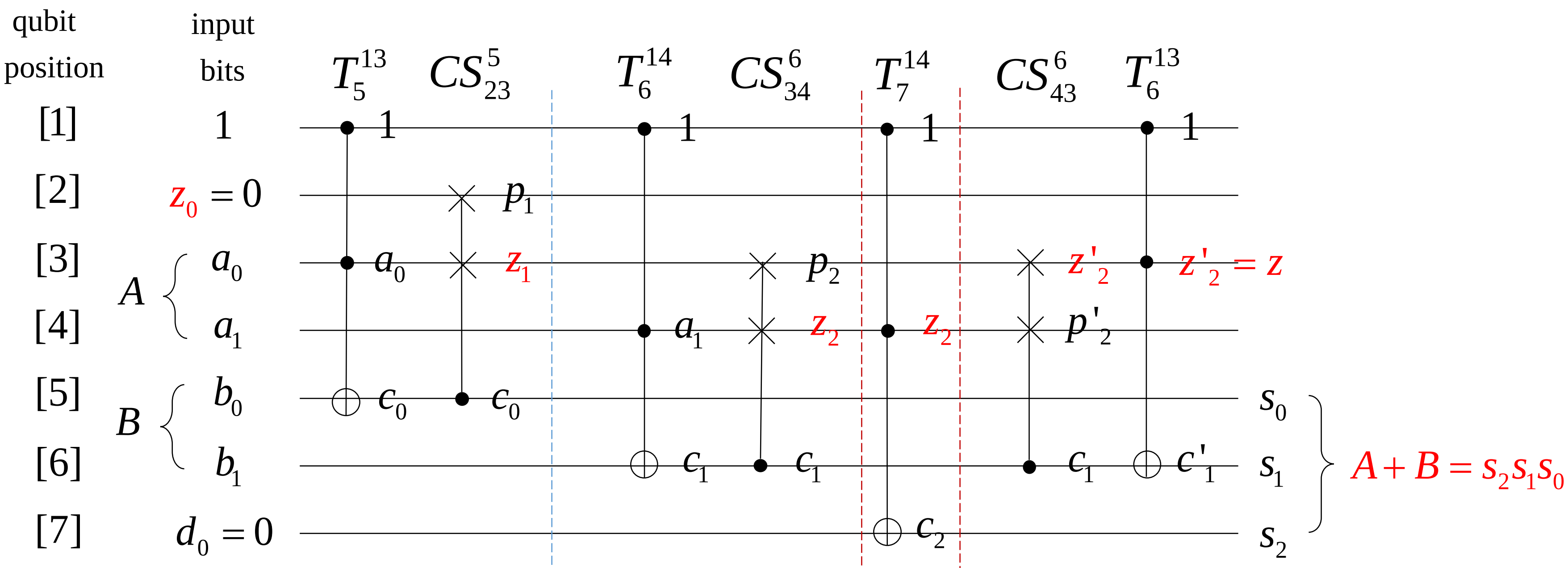
2-bit adder: given $A = a_1 a_0$ and $B = b_1 b_0 \in Z_2^2 = \{00, 01, 10, 11\}$,

1. Prepare the 7-qubit basis state $|\,1\ \ z_0\ \rangle|a_0 a_1\rangle|b_0 b_1\rangle|d_0\rangle$, $z_0 = 0 = d_0$

2. Apply the circuit $U^{(2)}_{\text{Add}}|\,1\ \ z_0\ \rangle|a_0 a_1\rangle|b_0 b_1\rangle|d_0\rangle = |\,1\ \ z\ \rangle|a_0 a_1\rangle|s_0 s_1 s_2\rangle$

$A + B = s_2 s_1 s_0$





$c_0 = b_0 \oplus (1 \wedge a_0)$ $\quad c_1 = b_1 \oplus (1 \wedge a_1)$ $\quad c_2 = 0 \oplus (1 \wedge z_2)$

$p_1 = (\bar{c}_0 \wedge z_0) \oplus (c_0 \wedge a_0)$ $\quad p_2 = (\bar{c}_1 \wedge z_1) \oplus (c_1 \wedge a_1)$ $\quad p'_2 = (\bar{c}_2 \wedge z_2) \oplus (c_2 \wedge p_2)$

$z_1 = (\bar{c}_0 \wedge a_0) \oplus (c_0 \wedge z_0)$ $\quad z_2 = (\bar{c}_1 \wedge a_1) \oplus (c_1 \wedge z_1)$ $\quad z'_2 = (\bar{c}_2 \wedge p_2) \oplus (c_2 \wedge z_2)$

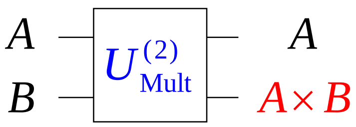
Quantum Multiplier

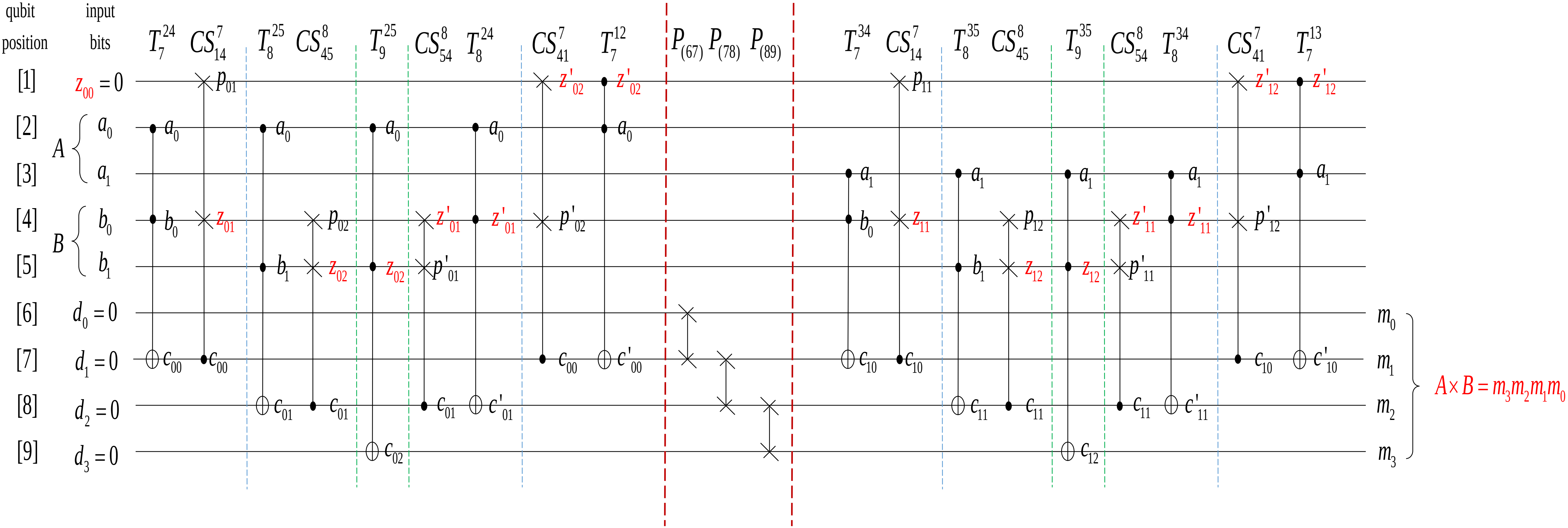
2-bit Multiplier: given $A = a_1 a_0$ and $B = b_1 b_0 \in Z_2^2 = \{00, 01, 10, 11\}$,

1. Prepare the 9-qubit basis state $|\psi\rangle = |0\rangle |a_0 a_1\rangle |b_0 b_1\rangle |0000\rangle$

2. Compute $U_{\text{Mult}}^{(2)} |\psi\rangle = |z_0\rangle |a_1 a_0\rangle |b_1 b_0\rangle |m_0 m_1 m_2 m_3\rangle$

$A \times B = m_3 m_2 m_1 m_0$





Runtimes of Basic Gates on codewords of length n

	$n=10^4$	$n=10^5$	$n=10^6$	$n=10^7$	$n=10^8$
S_α^ζ	2×10^{-4} s	2×10^{-3} s	3.6×10^{-2} s	2×10^{-1} s	2.1 s
C_j^i	10^{-8} s	10^{-8} s	10^{-8} s	10^{-8} s	10^{-8} s
$P_{(i\ j)}$	10^{-8} s	10^{-8} s	10^{-8} s	10^{-8} s	10^{-8} s
$T_l^{i\ j}$	10^{-8} s	10^{-8} s	10^{-8} s	10^{-8} s	10^{-8} s
$CS_l^{i\ j}$	10^{-8} s	10^{-8} s	10^{-8} s	10^{-8} s	10^{-8} s
$\Lambda_{n-1}^1(S_\omega^\pi)$	10^{-4} s	8×10^{-4} s	8.5×10^{-3} s	8×10^{-2} s	0.8 s

Hardware: Intel core i7-8565U CPU @ 1.80 GHz

Runtimes of the Adder $U^{(l)}_{\text{Add}}$ and the Multiplier $U^{(l)}_{\text{Mult}}$ given two l-bit integers

	$l=8$	$l=16$	$l=32$	$l=64$	$l=128$
$\text{Helib}_{\text{Add}}$		2.3×10^{-3}		$4.2\times10^{-2}s$	
$U^{(l)}_{\text{Add}}$	$1.08\times10^{-3}s$	$1.8\times10^{-3}s$	$2.2\times10^{-3}s$	$4\times10^{-3}s$	$5.8\times10^{-3}s$
$\text{Helib}_{\text{Multi}}$	4.17×10^{-3}	5.82×10^{-2}			
$U^{(l)}_{\text{Mult}}$	10^{-3}	2.8×10^{-3}	2.9×10^{-3}	3×10^{-3}	6×10^{-3}

Hardware: Intel core i7-8565U CPU @ 1.80 GHz

Hardware: Intel Core i7 4790 @ 3.60 GHz

J.-W. Chen, et. al., *Faster Binary Arithmetic Operations on Encrypted Integers*, WCSE'17

security level $= 128$

size of plaintext $\boldsymbol{x} = 64$ bits $\approx 10^{19}$ digits

HE computation	$\boldsymbol{x}^2$	$\boldsymbol{x}^4$	$\boldsymbol{x}^8$
HElib	$0.09\ s$ 30	$0.4\ s$ 100	$0.9\ s$ 112
SEAL	$0.4\ s$ 133	$1.5\ s$ 375	$1.5\ s$ 187
FV-NFLib	$0.08\ s$ 27	$0.09\ s$ 23	$0.3\ s$ 37
QAPHE	$0.003\ s$	$0.004\ s$	$0.008\ s$

Hardware: Intel core i7-8565U CPU @ 1.80 GHz

Hardware: Intel (R) Xeon (R) CPU E5-2695 v3 @ 2.30 GHz

C. A. Melchor, et.al., *A Comparison of the Homomorphic Encryption Libraries HElib, SEAL and FV-NFLlib*, SecITC 2018, 425–442, 2018.

security level = 128

size of plaintext $\boldsymbol{x}$ = 256 bits $\approx$ 10^{76} digits

HE computation	$\boldsymbol{x}^2$	$\boldsymbol{x}^4$	$\boldsymbol{x}^8$
HElib	1.5 s 187	2 s 68	7 s 58
SEAL	2 s 250	9 s 310	70 s 583
FV-NFLib	0.6 s 75	0.9 s 31	4 s 33
QAPHE	0.008 s	0.029 s	0.12 s

Hardware: Intel core i7-8565U CPU @ 1.80 GHz

Hardware: Intel (R) Xeon (R) CPU E5-2695 v3 @ 2.30 GHz

C. A. Melchor, et.al., *A Comparison of the Homomorphic Encryption Libraries HElib, SEAL and FV-NFLlib*, SecITC 2018, 425–442, 2018.

security level $= 128$

size of plaintext $\boldsymbol{x} = 2048$ bits $\approx 10^{146}$ digits

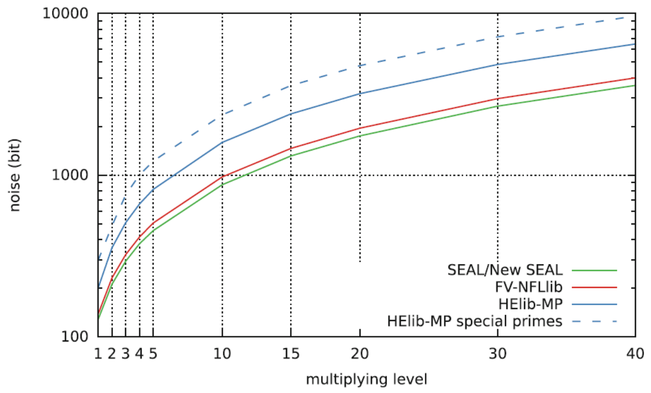
HE computation	$\boldsymbol{x}^2$	$\boldsymbol{x}^4$	$\boldsymbol{x}^8$
HElib	$50\ s$ 125	$200\ s$ 91	$500\ s$ 64
SEAL	N/A	N/A	N/A
FV-NFLib	$80\ s$ 200	$550\ s$ 250	$800\ s$ 102
QAPHE	$0.4\ s$	$2.2\ s$	$7.8\ s$

Hardware: Intel core i7-8565U CPU @ 1.80 GHz

Hardware: Intel (R) Xeon (R) CPU E5-2695 v3 @ 2.30 GHz

C. A. Melchor, et.al., *A Comparison of the Homomorphic Encryption Libraries HElib, SEAL and FV-NFLlib*, SecITC 2018, 425–442, 2018.

In current schemes, the computation time increases rapidly due to noise reduction.



C. A. Melchor, et.al., SecITC 2018, 425–442, 2018

In QAPHE, the computation time grows linearly with the number of basic gates applied.

size of plaintext $\boldsymbol{x}$ = 2048 bits	$\boldsymbol{x}^2$	$\boldsymbol{x}^4$	$\boldsymbol{x}^8$
Toffli gates	8.3×10^6	2.5×10^7	4.2×10^7
Ctrl. SWAPs	8.3×10^6	2.5×10^7	4.2×10^7
CNOTs \ SWAPs	4×10^3	6.1×10^3	1.4×10^4

QAPHE shows strong advantages with problems of

High Complexity and **Large Size**.

Current and Future Work

Parallelism of Gates and Circuits

One-Way Functions of Gates and Circuits

Problem-Dependent Optimization of Circuits

Dynamic and Miniature Modularization in Chips

Special-Purpose Chips, Architectures, Machines for QAPHE