TFHE
Homomorphic encryption and (a little bit) faster bootstrapping.

I. Chillotti\textsuperscript{1} N. Gama\textsuperscript{2,1} M. Georgieva\textsuperscript{3} M. Izabachène\textsuperscript{4}

Journées Codage et Cryptographie 2017
La Bresse - April 25, 2017
**IDEA:** perform computations on encrypted data, without decrypting it.

\( b_1, b_2 \in \{0, 1\} \)

\[
\begin{align*}
\boxed{b_1} & \quad \quad \boxed{b_1 \oplus_{hom} b_2} = \quad \boxed{b_1 \oplus b_2} \\
\boxed{b_2} & \quad \quad \boxed{b_1 \land_{hom} b_2} = \quad \boxed{b_1 \land b_2}
\end{align*}
\]
Homomorphic Encryption

More generally

\[
\begin{align*}
\begin{array}{c}
\vdots \\
\hline
b_1 \\
\hline
b_n
\end{array} & \quad \rightarrow \quad \varphi_{\text{hom}} \left( \begin{array}{c} b_1 \\
\vdots \\
\hline
b_n
\end{array} \right) & = & \varphi \left( b_1, \ldots, b_n \right)
\end{align*}
\]

where \( b_1, \ldots, b_n \in \{0, 1\} \) and \( \varphi \) is a boolean circuit.
More generally

$$\begin{array}{c}
\begin{bmatrix}
  b_1 \\
  \vdots \\
  b_n
\end{bmatrix} \\
\end{array} \rightarrow \phi_{hom}\left(\begin{bmatrix}
  b_1, \ldots, b_n
\end{bmatrix}\right) = \phi(b_1, \ldots, b_n)$$

where $b_1, \ldots, b_n \in \{0, 1\}$ and $\phi$ is a boolean circuit.

A scheme that can homomorphically evaluate all functions/circuits is said **Fully Homomorphic** (FHE).
Too much noise

anim
Bootstrapping

$c_1$

$c_2$

$\ldots$

$c_\ell$

-secret key

$k_1$

$k_2$

$\ldots$

$k_n$

-ciphertext

Decryption circuit (public)

message

encrypted

encrypted

Decryption circuit (public)

hom.
Bootstrapping

ciphertext
secret key

Decryption circuit (public)

message encrypted
Bootstrapping is the most expensive part of the entire homomorphic procedure

- Original idea by Gentry [Gen09]
- Last years: work to reduce the execution time and memory consuming

...but a lot has to be done!
Boostrap, or not boostrap, that is the question!

Leveled versions
Set the function: there exists parameters to homomorphically evaluate it!

- Very fast for small depth circuits
- Almost impractical for large depth circuits
- The depth has to be known in advance

Boostraped versions
Set the parameter: evaluate any function!

- No depth limitations
- Very slow
- Large bootstrap keys
Bootstrapping, or not bootstrapping, that is the question!

**Leveled versions**
Set the function: there exists parameters to homomorphically evaluate it!
- Very fast for small depth circuits

**Bootstrapped versions**
Set the parameter: evaluate any function!
Bootstrapping, or not bootstrapping, that is the question!

**Leveled versions**
Set the function: there exists parameters to homomorphically evaluate it!
- ✔ Very fast for small depth circuits
- ✗ Almost impractical for large depth circuits
- ✗ The depth has to be known in advance

**Bootstrapped versions**
Set the parameter: evaluate any function!
Boostrap, or not bootstrapping, that is the question!

Leveled versions
Set the function: there exists parameters to homomorphically evaluate it!

- Very fast for small depth circuits
- Almost impractical for large depth circuits
- The depth has to be known in advance

Bootstrapped versions
Set the parameter: evaluate any function!

- No depth limitations
Bootstrapping, or not bootstrapping, that is the question!

Leveled versions
Set the function: there exists parameters to homomorphically evaluate it!

- ✔ Very fast for small depth circuits
- ✗ Almost impractical for large depth circuits
- ✗ The depth has to be known in advance

Bootstrapped versions
Set the parameter: evaluate any function!

- ✔ No depth limitations
- ✗ Very slow
- ✗ Large bootstrapping keys
The construction

<table>
<thead>
<tr>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The construction

<table>
<thead>
<tr>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>$T$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example: $M = \{0, 1/3, 2/3\} \mod 1$

$\mu = 1/3 \mod 1 \in M$
The construction

<table>
<thead>
<tr>
<th></th>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example: \( M = \{0, 1/3, 2/3\} \mod 1 \)
\( \mu = 1/3 \mod 1 \in M \)

\( \varphi = \mu + \text{Gaussian Error} \)
The construction

<table>
<thead>
<tr>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>T</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example: \( M = \{0, 1/3, 2/3\} \mod 1 \)

\( \mu = 1/3 \mod 1 \in M \)

1. \( \varphi = \mu + \text{Gaussian Error} \)
2. Random mask \( a \in T^n \)
The construction

<table>
<thead>
<tr>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>$\mathbb{T}$</td>
<td>$\mathbb{T}^{n+1}$</td>
<td></td>
</tr>
</tbody>
</table>

**Example:** $M = \{0, 1/3, 2/3\} \mod 1$

$\mu = 1/3 \mod 1 \in M$

**Secret key:** $s \in \{0, 1\}^n$

$2/3 \quad 1/3$

$b = s \cdot a + \varphi$

$\varphi = \mu + \text{Gaussian Error}$

Random mask $a \in \mathbb{T}^n$
The construction

<table>
<thead>
<tr>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>$T$</td>
<td>$T^{n+1}$</td>
<td></td>
</tr>
</tbody>
</table>

secret key: $s \in \{0, 1\}^n$
The construction

<table>
<thead>
<tr>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>$\mathbb{T}$</td>
<td>$\mathbb{T}^{n+1}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

secret key: $s \in \{0, 1\}^n$

\[ \varphi = b - s \cdot a \]

Unlock the representation \((a, \varphi)\)
The construction

<table>
<thead>
<tr>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>ℤ</td>
<td>ℤ^{n+1}</td>
<td></td>
</tr>
</tbody>
</table>

secret key: \( s \in \{0, 1\}^n \)

1. Unlock the representation \((a, \varphi)\)
2. Round \(\varphi\) to the nearest message \(\mu \in M\)
The construction

<table>
<thead>
<tr>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>$\mathbb{T}$</td>
<td>$\mathbb{T}^{n+1}$</td>
<td>✓</td>
</tr>
</tbody>
</table>

$x \cdot \mathbb{a} + y \cdot \mathbb{a}' = \mathbb{a}''$

$\mathbb{a}'' = x \cdot \mathbb{a} + y \cdot \mathbb{a}'$

$\mathbb{b}'' = x \cdot \mathbb{b} + y \cdot \mathbb{b}'$
The construction

<table>
<thead>
<tr>
<th></th>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>$\mathbb{T}$</td>
<td>$\mathbb{T}^{n+1}$</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

\[\begin{align*}
  x \cdot a + y \cdot a' &= a'' & a'' &= x \cdot a + y \cdot a' \\
  x \cdot \varphi + y \cdot \varphi' &= \varphi'' & \varphi'' &= x \cdot \varphi + y \cdot \varphi'
\end{align*}\]

\[\begin{align*}
  \mu &= \mathbb{E}(\varphi) & \mu' & \mu'' & \mu'' &= x \cdot \mu + y \cdot \mu' \\
  \alpha &= \text{stdev}(\varphi) & \alpha' & \alpha'' & \alpha'' &= x^2 \alpha^2 + y^2 \alpha'^2
\end{align*}\]
## The construction

<table>
<thead>
<tr>
<th></th>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>$T$</td>
<td>$T^{n+1}$</td>
<td>$\checkmark$</td>
<td>$\times$</td>
</tr>
</tbody>
</table>
The construction

<table>
<thead>
<tr>
<th></th>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>( \mathbb{T} )</td>
<td>( \mathbb{T}^{n+1} )</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>TLWE</td>
<td>( \mathbb{T}_N[X] )(^1)</td>
<td>( \mathbb{T}_N[X]^{k+1} )</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

\(^1\)\( \mathbb{T}\_N[X] = \mathbb{T}[X]/(X^N + 1) \)
The construction

<table>
<thead>
<tr>
<th></th>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>$\mathbb{T}$</td>
<td>$\mathbb{T}^{n+1}$</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>TLWE</td>
<td>$\mathbb{T}_N[X]$</td>
<td>$\mathbb{T}_N[X]^{k+1}$</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>TGSW</td>
<td>$\mathbb{Z}[X]/(X^N + 1)$</td>
<td>$\ell'$-vector of TLWE</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

\(^1\mathbb{T}_N[X] = \mathbb{T}[X]/(X^N + 1)\)
The construction

<table>
<thead>
<tr>
<th></th>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>$T$</td>
<td>$T^{n+1}$</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>TLWE</td>
<td>$T_N[X]^1$</td>
<td>$T_N[X]^{k+1}$</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>TGSW</td>
<td>$\mathbb{Z}[X]/(X^N + 1)$</td>
<td>$\ell'$-vector of TLWE</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

 prohibitions

2 Internal Product (classical):

$\blacklozenge: \text{TGSW} \times \text{TGSW} \rightarrow \text{TGSW}$

\(^1T_N[X] = \mathbb{T}[X]/(X^N + 1)\)
The construction

<table>
<thead>
<tr>
<th></th>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>$T^n$</td>
<td>$T^{n+1}$</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>TLWE</td>
<td>$T_N[X]^{1}$</td>
<td>$T_N[X]^{k+1}$</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>TGSW</td>
<td>$\mathbb{Z}[X]/(X^N + 1)$</td>
<td>$\ell'$-vector of TLWE</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

1. **External product (found independently by [BP16]):**

   $\square: TGSW \times TLWE \rightarrow TLWE$

   $((A, b) \mapsto A \square b = Dec_{h, \beta, \epsilon}(b) \cdot A$

   $(\mu_A, \mu_b) \mapsto \mu_A \cdot \mu_b$

2. **Internal Product (classical):**

   $\boxdot: TGSW \times TGSW \rightarrow TGSW$

$^1T_N[X] = T[X]/(X^N + 1)$
The construction

<table>
<thead>
<tr>
<th></th>
<th>message</th>
<th>ciphertext</th>
<th>linear combin.</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWE</td>
<td>$T$</td>
<td>$T^{n+1}$</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>TLWE</td>
<td>$T_N[X]^1$</td>
<td>$T_N[X]^{k+1}$</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>TGSW</td>
<td>$\mathbb{Z}[X]/(X^N + 1)$</td>
<td>$\ell'$-vector of TLWE</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**External product (found independently by [BP16]):**

\[\Box: TGSW \times TLWE \rightarrow TLWE\]

\[
(A, b) \mapsto A \Box b = Dec_{h, \beta, \epsilon}(b) \cdot A
\]

\[
(\mu_A, \mu_b) \mapsto \mu_A \cdot \mu_b
\]

**Internal Product (classical):**

\[\star: TGSW \times TGSW \rightarrow TGSW\]

\[
(A, B) \mapsto A \star B = \begin{bmatrix}
A \Box b_1 \\
\vdots \\
A \Box b_{\ell'}
\end{bmatrix}
\]
We reformalized and improved the fast bootstrapping proposed by Ducas and Micciancio [DM15]

- GSW [GSW13] based construction
- Fast addition and non-linear operations done via bootstrapping
- Homomorphic NAND gate with fast bootstrapping in $\sim 0.69$ seconds

Improved the leveled construction
[CGGI16] - I. Chillotti, N. Gama, M. Georgieva, M. Izabachène, Faster Fully Homomorphic Encryption: Bootstrapping in less than 0.1 Seconds - Asiacrypt 2016

1. We reformalized and improved the fast bootstrapping proposed by Ducas and Micciancio [DM15]
   - GSW [GSW13] based construction
   - Fast addition and non-linear operations done via bootstrapping
   - Homomorphic NAND gate with fast bootstrapping in $\sim 0.69$ seconds

2. Improved the leveled constructon
Boostrapping
Bootstrapping
[Gentry09]-style bootstrap
[Gentry09]-style bootstrap
Bootstrapping

[DM15]-style bootstrap
false := LWE(−\(\frac{1}{8}\)), noise < \(\frac{1}{16}\)
true := LWE(+$\frac{1}{8}$), noise < $\frac{1}{16}$
Gate Bootstrapping

\[ c_1 + c_2 = \frac{1}{2} \]

\[ \frac{3}{4} - \frac{1}{8} = \frac{1}{4} \]
Gate Bootstrapping

\[ \text{NAND}(c_1, c_2) : \]

- Return false if \( c_1 = 0 \) and \( c_2 = 1 \)
- Return true if \( c_1 = 1 \) and \( c_2 = 0 \)
- Return true if \( c_1 = c_2 = 1 \)

Diagram: Connections and probabilities for gate bootstrapping.
Gate Bootstrapping

\[ \text{NAND}(c_1, c_2) : \]

return false

\[ \frac{1}{2} \]

return true

\[ \frac{3}{4} \]

\[ \frac{1}{8} \]

\[ \frac{1}{8} \]

\[ 0 \]

\[ 0 \]

\[ \frac{1}{2} \]

\[ \frac{1}{2} \]

\[ \text{return false} \]

\[ \text{return true} \]

\[ 0 \]

\[ \frac{1}{2} \]

\[ \frac{3}{4} \]

\[ \frac{1}{4} \]

\[ \frac{1}{8} \]

\[ \frac{1}{8} \]
Gate Bootstrapping

[DM15/BR15]-(revisited)

\[ v_0, v_1, v_2, \ldots, v_{2N-1} \]
Bootstrapping Algorithm (animation)

Bootstrapping algorithm of (a, b)

1. Start from (a trivial) $\text{TLWE}(v_0 + v_1 X + \cdots + v_{N-1} X^{N-1})^a$
2. Rotate it by $p = -\varphi_s(a, b)$ positions
3. Extract the constant term (which encrypts $v_p$)

$^aN$ coefs mod $X^N + 1$ can be viewed as $2N$ coefs mod $X^{2N} - 1$ s.t. $v_{N+i} = -v_i$
### Bootstrapping Algorithm (animation)

**Bootstrapping algorithm of \((a, b)\)**

1. Start from (a trivial) TLWE\((v_0 + v_1X + \cdots + v_{N-1}X^{N-1})^a\)
2. Rotate it by \(p = -\varphi_s(a, b)\) positions
3. Extract the constant term (which encrypts \(v_p\))

---

\(^a\text{N coefs mod } X^N + 1\) can be viewed as \(2N\) coefs mod \(X^{2N} - 1\) s.t. \(v_{N+i} = -v_i\)
### Bootstrapping Algorithm of \((a, b)\)

1. **Start from** (a trivial) \(\text{TLWE}(v_0 + v_1X + \cdots + v_{N-1}X^{N-1})^a\)
2. **Rotate it** by \(p = -\varphi_s(a, b)\) positions
3. **Extract the constant term** (which encrypts \(v_p\))

\[N\text{ coefs mod } X^N + 1\text{ can be viewed as }2N\text{ coefs mod } X^{2N} - 1\text{ s.t. }v_{N+i} = -v_i\]

### Rotate by \(p\) positions the coefficients \(c \in \text{TLWE}\)
Bootstrapping algorithm of \((a, b)\)

1. Start from (a trivial) TLWE\((v_0 + v_1X + \cdots + v_{N-1}X^{N-1})^a\)
2. Rotate it by \(p = -\varphi_s(a, b)\) positions
3. Extract the constant term (which encrypts \(v_p\))

\(a\)N coefs mod \(X^N + 1\) can be viewed as \(2N\) coefs mod \(X^{2N} - 1\) s.t. \(v_{N+i} = -v_i\)

Rotate by \(p\) positions the coefficients \(c \in \text{TLWE}\)

- \((X^p \cdot c)\) when \(p\) is known
Bootstrapping Algorithm (animation)

Bootstrapping algorithm of \((a, b)\)

1. Start from (a trivial) \(\text{TLWE}(v_0 + v_1X + \cdots + v_{N-1}X^{N-1})^a\)
2. Rotate it by \(p = -\varphi_s(a, b)\) positions
3. Extract the constant term (which encrypts \(v_p\))

\[ aN \text{ coefs mod } X^N + 1 \text{ can be viewed as } 2N \text{ coefs mod } X^{2N} - 1 \text{ s.t. } v_{N+i} = -v_i \]

Rotate by \(p\) positions the coefficients \(c \in \text{TLWE}\)

- \((X^p \cdot c)\) when \(p\) is known
- \((\text{TGSW}(X^p) \boxdot c)\) when \(p\) is unknown
# Bootstrapping Algorithm (animation)

## Bootstrapping algorithm of \((a, b)\)

1. Start from (a trivial) \(\text{TLWE}(v_0 + v_1X + \cdots + v_{N-1}X^{N-1})^a\)
2. Rotate it by \(p = -\varphi_s(a, b)\) positions
3. Extract the constant term (which encrypts \(v_p\))

\[^a\text{N coefs mod } X^N + 1\text{ can be viewed as } 2N\text{ coefs mod } X^{2N} - 1\text{ s.t. } v_{N+i} = -v_i\]

## Rotate by \(p\) positions the coefficients \(c \in \text{TLWE}\)

- \((X^p \cdot c)\) when \(p\) is known
- \((\text{TGSW}(X^p) \Box c)\) when \(p\) is unknown

## How to rotate by \(-\varphi_s(a, b) = -b + \sum_{i=1}^{n} a_i s_i\)?
## Bootstrapping Algorithm (animation)

### Bootstrapping algorithm of \((a, b)\)

1. Start from (a trivial) \(\text{TLWE}(v_0 + v_1X + \cdots + v_{N-1}X^{N-1})^a\)
2. Rotate it by \(p = -\varphi_s(a, b)\) positions
3. Extract the constant term (which encrypts \(v_p\))

\[^aN\text{ coefs mod }X^N + 1\text{ can be viewed as }2N\text{ coefs mod }X^{2N} - 1\text{ s.t. }v_{N+i} = -v_i\]

### Rotate by \(p\) positions the coefficients \(c \in \text{TLWE}\)

- \((X^p \cdot c)\) when \(p\) is known
- \((\text{TGSW}(X^p) \boxtimes c)\) when \(p\) is unknown

### How to rotate by \(-\varphi_s(a, b) = -b + \sum_{i=1}^n a_is_i?\)

1. Multiply by \(X^{-b}\)
Bootstrapping Algorithm (animation)

Bootstrapping algorithm of \((a, b)\)

1. Start from (a trivial) \(\text{TLWE}(v_0 + v_1X + \cdots + v_{N-1}X^{N-1})^a\)
2. Rotate it by \(p = -\varphi_s(a, b)\) positions
3. Extract the constant term (which encrypts \(v_p\))

\[ aN \text{ coefs mod } X^N + 1 \text{ can be viewed as } 2N \text{ coefs mod } X^{2N} - 1 \text{ s.t. } v_{N+i} = -v_i \]

Rotate by \(p\) positions the coefficients \(c \in \text{TLWE}\)

- \((X^p \cdot c)\) when \(p\) is known
- \((\text{TGSW}(X^p) \square c)\) when \(p\) is unknown

How to rotate by \(-\varphi_s(a, b) = -b + \sum_{i=1}^{n} a_is_i?\)

1. Multiply by \(X^{-b}\)
2. For \(i \in [1, n]\) multiply by \(\text{TGSW}(X^{-a_is_i})\)
# Bootstrapping Algorithm (animation)

## Bootstrapping algorithm of \((a, b)\)

1. Start from (a trivial) TLWE\((v_0 + v_1X + \cdots + v_{N-1}X^{N-1})^a\)
2. Rotate it by \(p = -\varphi_s(a, b)\) positions
3. Extract the constant term (which encrypts \(v_p\))

\[ aN \text{ coefs mod } X^{N} + 1 \text{ can be viewed as } 2N \text{ coefs mod } X^{2N} - 1 \text{ s.t. } v_{N+i} = -v_i \]

## Rotate by \(p\) positions the coefficients \(c \in \text{TLWE}\)

- \((X^p \cdot c)\) when \(p\) is known
- \((\text{TGSW}(X^p) \boxtimes c)\) when \(p\) is unknown

## How to rotate by \(−\varphi_s(a, b) = −b + \sum_{i=1}^{n} a_i s_i\)?

1. Multiply by \(X^{-b}\)
2. For \(i \in [1, n]\) multiply by \(\text{TGSW}(X^{-a_i s_i})\)
   - \(X^{a_i s_i} = 1 + (X^{a_i} - 1) \cdot s_i\), with \(s_i \in \{0, 1\}\)
Bootstrapping Algorithm (animation)

## Bootstrapping algorithm of \((a, b)\)

1. Start from (a trivial) TLWE\((v_0 + v_1 X + \cdots + v_{N-1} X^{N-1})^a\)
2. Rotate it by \(p = -\varphi_s(a, b)\) positions
3. Extract the constant term (which encrypts \(v_p\))

\[aN\text{ coefs mod } X^N + 1\text{ can be viewed as }2N\text{ coefs mod } X^{2N} - 1\text{ s.t. }v_{N+i} = -v_i\]

## Rotate by \(p\) positions the coefficients \(c \in TLWE\)

- \((X^p \cdot c)\) when \(p\) is known
- \((\text{TGSW}(X^p) \sqcap c)\) when \(p\) is unknown

## How to rotate by \(-\varphi_s(a, b) = -b + \sum_{i=1}^{n} a_i s_i?\)

1. Multiply by \(X^{-b}\)
2. For \(i \in [1, n]\) multiply by \(\text{TGSW}(X^{-a_i s_i})\)
   - \(X^{a_i s_i} = 1 + (X^{a_i} - 1) \cdot s_i\), with \(s_i \in \{0, 1\}\)
   - \(\text{TGSW}(X^{a_i s_i}) = h + (X^{a_i} - 1) \cdot \text{TGSW}(s_i)\), where BK = \(\text{TGSW}(s_i)\)
Faster bootstrapping

- [DM15] Bootstrapped NAND gate: $\sim 0.69$ seconds
- [CGGI16] Several bootstrapped gates: $\sim 0.052$ seconds$^2$

---

$^2$Implemented in C/C++, run the bootstrapping algorithm on a 64-bit single core (i7-4930MX) at 3.00GHz.
Faster bootstrapping

- [DM15] Bootstrapped NAND gate: $\sim 0.69$ seconds
- [CGGI16] Several bootstrapped gates: $\sim 0.052$ seconds$^2$
- **TFHE:** Several bootstrapped gates: $\sim 20\text{ms}$

TFHE $\rightarrow$ [https://github.com/tfhe/tfhe](https://github.com/tfhe/tfhe)

---

$^2$Implemented in C/C++, run the bootstrapping algorithm on a 64-bit single core (i7-4930MX) at 3.00GHz.
[CGGI16] - I. Chillotti, N. Gama, M. Georgieva, M. Izabachène, Faster Fully Homomorphic Encryption: Bootstrapping in less than 0.1 Seconds - Asiacrypt 2016

- We reformalized and improved the fast bootstrapping proposed by Ducas and Micciancio [DM15]
  - GSW [GSW13] based construction
  - Fast addition and non-linear operations done via bootstrapping
  - Homomorphic NAND gate with fast bootstrapping in $\sim 0.69$ seconds
- Improved the leveled construction
Leveled construction

- No bootstrapping
- Linear combinations and external products

\[
\begin{align*}
\mu_A &\quad \text{T-GSW} \\
\eta_A &\quad \mu_A \cdot \mu_b \\
\mu_b &\quad \text{T-LWE} \\
\eta_b &\quad \|\mu_A\|_1 \eta_b + O(\eta_A)
\end{align*}
\]

- Different types of wires: composability?
Conclusion

- Improved bootstrapping constructions
- Improved leveled constructions
- Leveled vs. Fully

Thank you! Questions?

m.s.n.
Conclusion

- Improved bootstrapping constructions
- Improved leveled constructions
- Leveled vs. Fully

Open problems

- Solve the composability problem
- Make the FHE-dream come "(N)TRU"\(^a\)

\(^a\)Merci Cédric :)

Thank you! Questions?
Conclusion

- Improved bootstrapping constructions
- Improved leveled constructions
- Leveled vs. Fully

Open problems

- Solve the composability problem
- Make the FHE-dream come "(N)TRU"\(^a\)

\(^a\)Merci Cédric :)

Thank you!
Questions?