What is the Next Goal?

“HE system can evaluate an arbitrary circuit in a polynomial time.”
What is the Next Goal?

“HE system can evaluate an arbitrary circuit in a polynomial time.”

Cryptography community has improved the Efficiency of HE system.
- Performance: Speed, Storage, Expansion rate, etc.
- Functionality: Key-switching, Rotation, Plaintext Space, etc.
What is the Next Goal?

“What HE system can evaluate an arbitrary circuit in a polynomial time.”

Cryptography community has improved the Efficiency of HE system.

- Performance: Speed, Storage, Expansion rate, etc.
- Functionality: Key-switching, Rotation, Plaintext Space, etc.

Reduction of “Gap” between Real and Encrypted computations.

Datatypes and operations

- Boolean Circuit (Bit Operation)
- Integer Operations
- Modular Arithmetic
- Approximate Arithmetic (Fixed/Floating-point Operation)
- Logical Operations (If & Else statement)
# Homomorphic Encryption Schemes

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Plaintext Slot</th>
<th>Good</th>
<th>Bad</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Word Encryption” Brakerski-Gentry-Vaikuntanathan’12 Gentry-Halevi-Smart’12a,b,c Brakerski’12, Fan-Vercauteren’12 Halevi-Shoup’13,14,15</td>
<td>GF($p^d$) ($Z_p$)</td>
<td>Polylog overhead (Amortized time &amp; Expansion rate)</td>
<td>Bootstrapping</td>
<td>HElib SEAL …</td>
</tr>
<tr>
<td>Gentry-Sahai-Waters’13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Bitwise Encryption” Ducas-Micciancio’15 Chillotti-Gama-Georgieva-Izabachene’16,17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Homomorphic Encryption Schemes

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Plaintext Slot</th>
<th>Good</th>
<th>Bad</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Word Encryption”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brakerski-Gentry-Vaikuntanathan’12</td>
<td>$\text{GF}(p^d) \ (Z_p)$</td>
<td>Polylog overhead (Amortized time &amp; Expansion rate)</td>
<td>Bootstrapping</td>
<td>HElib SEAL ...</td>
</tr>
<tr>
<td>Gentry-Halevi-Smart’12a,b,c</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brakerski’12, Fan-Vercauteren’12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Halevi-Shoup’13,14,15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gentry-Sahai-Waters’13</td>
<td>$Z, Z[X]\ {{{0,1}}}$</td>
<td>Beauty 😊</td>
<td>Inefficient</td>
<td></td>
</tr>
<tr>
<td>“Bitwise Encryption”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ducas-Micciancio’15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chillotti-Gama-Georgieva-Izabachene’16,17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Homomorphic Encryption Schemes

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Plaintext Slot</th>
<th>Good</th>
<th>Bad</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Word Encryption” Brakerski-Gentry-Vaikuntanathan’12 Gentry-Halevi-Smart’12a,b,c Brakerski’12, Fan-Vercauteren’12 Halevi-Shoup’13,14,15</td>
<td>( \text{GF}(p^d) (\mathbb{Z}_p) )</td>
<td>Polylog overhead (Amortized time &amp; Expansion rate)</td>
<td>Bootstrapping</td>
<td>HElib SEAL …</td>
</tr>
<tr>
<td>Gentry-Sahai-Waters’13</td>
<td>( \mathbb{Z}, \mathbb{Z}[X] {0,1} )</td>
<td>Beauty Toolkit for FHEW</td>
<td>Inefficient</td>
<td></td>
</tr>
<tr>
<td>“Bitwise Encryption” Ducas-Micciancio’15 Chillotti-Gama-Georgieva-Izabachene’16,17</td>
<td>( {0,1}, {(0,1)^*} )</td>
<td>Evaluation with Bootstrapping. Latency.</td>
<td>Amortized time &amp; Expansion rate</td>
<td>FHEW TFHE</td>
</tr>
</tbody>
</table>
Application Researches of HE (2017~2018)

“Homomorphic Encryption” in ePrint and IEEE Xplore
Application Researches of HE (2017~2018)

“Homomorphic Encryption” in ePrint and IEEE Xplore

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
<th>2018/2017</th>
<th>Journals/Conferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine Learning</td>
<td>11</td>
<td>233,202,139,074, 979,715</td>
<td>SSCI, IEEE Access, IEEE Journal, ICCV, SMARTCOMP</td>
</tr>
<tr>
<td>Neural Network</td>
<td>2</td>
<td>073, 1114</td>
<td></td>
</tr>
<tr>
<td>Genomic Data</td>
<td>7</td>
<td>955,770,294,228</td>
<td>EUSIPCO, SMARTCOMP, IEEE Journal</td>
</tr>
<tr>
<td>Health Data</td>
<td>2</td>
<td></td>
<td>IBM Journal, IEEE Journal</td>
</tr>
<tr>
<td>Biometric Data</td>
<td>2</td>
<td></td>
<td>IEEE Access, IEEE Conference</td>
</tr>
<tr>
<td>Energy Management</td>
<td>3</td>
<td>1212</td>
<td>IEEE Big Data, IET Journal</td>
</tr>
<tr>
<td>Big Data</td>
<td>1</td>
<td></td>
<td>ICBDA</td>
</tr>
<tr>
<td>Advertising</td>
<td>1</td>
<td></td>
<td>WIFS</td>
</tr>
<tr>
<td>Internet of Things</td>
<td>1</td>
<td></td>
<td>IWCMC</td>
</tr>
<tr>
<td>Election</td>
<td>1</td>
<td>166</td>
<td></td>
</tr>
</tbody>
</table>
Application Researches of HE (2017~2018)

“Homomorphic Encryption” in ePrint and IEEE Xplore

- **Neural Network**: 2 articles (2018/073, 2017/1114)
- **Genomic Data**: 7 articles (2017/955,770,294,228. EUSIPCO, SMARTCOMP, IEEE Journal)
- **Health Data**: 2 articles (IBM Journal, IEEE Journal)
- **Biometric Data**: 2 articles (IEEE Access, IEEE Conference)
- **Big Data**: 1 article (ICBDA)
- **Advertising**: 1 article (WIFS)
- **Internet of Things**: 1 article (IWCMC)
- **Election**: 1 article (2017/166)
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?
How to perform Approximate Arithmetic on HE?

\[1.234 \times 0.689 \times 2.194 \times 0.917 \times 3.323 \times 4.154 \times 0.489 \times 3.772 = ?\]

Word Encryption

- Represent a real number as an integer.
- No **Rounding** operation is very expensive.
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

Word Encryption

- Represent a real number as an integer.
- No Rounding operation is very expensive.
- Bit size of message grows exponentially.

1,234 * 689 * 2,194 * 917 * 3,323 * 4,154 * 489 * 3,772 = 4,355,296,408,921,213,975,719,328 > 2^{85}. 
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

**Word Encryption**

- Represent a real number as an integer.
- No *Rounding* operation is very expensive.
- Bit size of message grows *exponentially.*

1,234 * 689 * 2,194 * 917 * 3,323 * 4,154 * 489 * 3,772 = 4,355,296,408,921,213,975,719,328 > 2^{85}.

**Base Encoding [DGL+’15, CSCW’16, CLPX’17 (High-precision HE)]**

- Express a real number as a (small) polynomial. e.g. \((1.234) \rightarrow (1 + 2X^{-1} + 3X^{-2} + 4X^{-3})\)
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

Word Encryption

- Represent a real number as an integer.
- No Rounding operation is very expensive.
- Bit size of message grows exponentially.

1,234 * 689 * 2,194 * 917 * 3,323 * 4,154 * 489 * 3,772 = 4,355,296,408,921,213,975,719,328 > 2^{85}.

Base Encoding [DGL+’15, CSCW’16, CLPX’17 (High-precision HE)]

- Express a real number as a (small) polynomial. e.g. \( 1.234 \rightarrow (1 + 2X^{-1} + 3X^{-2} + 4X^{-3}) \)
- Exponential growth of Degree
- Trade-off between Precision & Number of slots
- No Bootstrapping
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

Bitwise Encryption

- 0.06 sec for (2-to-1) gate.
- 10 sec for (6-to-6) circuit.
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

Bitwise Encryption

- 0.06 sec for (2-to-1) gate.
- 10 sec for (6-to-6) circuit.

75 Gates for an operation on two four-bit strings.

How many gates for 16-bit / 32-bit precision multiplication?
## Homomorphic Encryption Schemes

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Plaintext Slot</th>
<th>Good</th>
<th>Bad</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Word Encryption” Brakerski-Gentry-Vaikuntanathan’12 Gentry-Halevi-Smart’12a,b,c Brakerski’12, Fan-Vercauteren’12 Halevi-Shoup’13,14,15</td>
<td>$\text{GF}(p^d) \ (\mathbb{Z}_p)$</td>
<td>Polylog overhead (Amortized time &amp; Expansion rate)</td>
<td>Bootstrapping</td>
<td>HElib SEAL ...</td>
</tr>
<tr>
<td>Gentry-Sahai-Waters’13</td>
<td>$\mathbb{Z}, \mathbb{Z}[X] \ {0,1}$</td>
<td>Beauty Toolkit for FHEW</td>
<td>Inefficient</td>
<td></td>
</tr>
<tr>
<td>“Bitwise Encryption” Ducas-Micciancio’15 Chillotti-Gama-Georgieva-Izabachene’16,17</td>
<td>${0,1}, \ {(0,1)^k}$</td>
<td>Evaluation with Bootstrapping. Latency.</td>
<td>Amortized time &amp; Expansion rate</td>
<td>FHEW TFHE</td>
</tr>
<tr>
<td>Scheme</td>
<td>Plaintext Slot</td>
<td>Good</td>
<td>Bad</td>
<td>Library</td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
<td>------</td>
<td>-----</td>
<td>---------</td>
</tr>
<tr>
<td>“Word Encryption” Brakerski-Gentry-Vaikuntanathan’12 Gentry-Halevi-Smart’12a,b,c Brakerski’12, Fan-Vercauteren’12 Halevi-Shoup’13,14,15</td>
<td>GF($p^d$) ($Z_p$)</td>
<td>Polylog overhead (Amortized time &amp; Expansion rate)</td>
<td>Bootstrapping</td>
<td>HElib SEAL</td>
</tr>
<tr>
<td>Gentry-Sahai-Waters’13</td>
<td>$Z, Z[X] {0,1}$</td>
<td>Beauty Toolkit for FHEW</td>
<td>Inefficient</td>
<td></td>
</tr>
<tr>
<td>“Bitwise Encryption” Ducas-Micciancio’15 Chillotti-Gama-Georgieva-Izabachene’16,17</td>
<td>${0,1}, {(0,1)^k}$</td>
<td>Evaluation with Bootstrapping. Latency.</td>
<td>Amortized time &amp; Expansion rate</td>
<td>FHEW TFHE</td>
</tr>
</tbody>
</table>
Approximate Homomorphic Encryption

Motivation: *Imitate* the approximate arithmetic on computer system.

\[ 1.234 \times 0.689 = (1,234 \times 10^{-3}) \times (689 \times 10^{-3}) \]
Approximate Homomorphic Encryption

Motivation: *Imitate* the approximate arithmetic on computer system.

\[1.234 \times 0.689 = (1,234 \times 10^{-3}) \times (689 \times 10^{-3}) = 850,226 \times 10^{-6} = 850 \times 10^{-3}\]
Approximate Homomorphic Encryption

Motivation: Imitate the approximate arithmetic on computer system.

\[ 1.234 \times 0.689 = (1,234 \times 10^{-3}) \times (689 \times 10^{-3}) = 850,226 \times 10^{-6} = 850 \times 10^{-3} \]

Idea 1. Every number contains an Approximation Error (between unknown true value).

Consider an RLWE error as part of it.
Approximate Homomorphic Encryption

Motivation: *Imitate* the approximate arithmetic on computer system.

\[ 1.234 \times 0.689 = (1,234 \times 10^{-3}) \times (689 \times 10^{-3}) = 850,226 \times 10^{-6} = 850 \times 10^{-3} \]

Idea 1. Every number contains an Approximation Error (between unknown true value).

Consider an RLWE error as part of it.

\[ ct = \text{Enc} (m) \quad \text{if} \quad [\langle ct, sk \rangle]_q = m + e \approx m. \]
Approximate Homomorphic Encryption

Motivation: Imitate the approximate arithmetic on computer system.

\[ 1.234 \times 0.689 = (1,234 \times 10^{-3}) \times (689 \times 10^{-3}) = 850,226 \times 10^{-6} = 850 \times 10^{-3} \]

Idea 1. Every number contains an Approximation Error (between unknown true value).
Consider an RLWE error as part of it.

\[ \text{ct} = \text{Enc} \ (m) \quad \text{if} \quad [<\text{ct},\text{sk}>]_q = m + e \approx m. \]

Approximate HE: \(1.234 \Rightarrow \) (scale by \( p=10^4 \)) \( \Rightarrow 12,340. \)
\( \Rightarrow \text{(Encrypt)} \Rightarrow [<\text{ct},\text{sk}>]_q = 12,342 \approx 1.234 \times 10^4. \)
Approximate Homomorphic Encryption

Motivation: Imitate the approximate arithmetic on computer system.

\[
1.234 \times 0.689 = (1.234 \times 10^{-3}) \times (689 \times 10^{-3}) = 850,226 \times 10^{-6} = 850 \times 10^{-3}
\]

Idea 2. Approximate Rounding is easy!

\[
<ct, sk> = m \pmod{q}
\]

\[
ct \mapsto ct' = \left\lfloor p^{-1} \times ct \right\rfloor
\]

\[
<ct', sk> \pmod{p^{-1}q} \approx p^{-1}m
\]
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

<table>
<thead>
<tr>
<th>1.234</th>
<th>0.689</th>
<th>2.194</th>
<th>0.917</th>
<th>3.323</th>
<th>4.154</th>
<th>0.489</th>
<th>3.772</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to perform Approximate Arithmetic on HE?

\[1.234 \times 0.689 \times 2.194 \times 0.917 \times 3.323 \times 4.154 \times 0.489 \times 3.772 = ?\]
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

12,340
6,890
21,940
9,170
33,230
41,540
4,890
37,720

Scaling

12,337
6,893
21,941
9,175
33,225
41,543
4,892
37,718

Encrypt

85,038,943
201,308,673
1,380,266,171
184,516,459

HomMult

8,499
20,125
138,021
18,451

HomRnd
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?

8,499
20,125
138,021
18,451
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = ?
How to perform Approximate Arithmetic on HE?

$$1.234 \times 0.689 \times 2.194 \times 0.917 \times 3.323 \times 4.154 \times 0.489 \times 3.772 = 43.555$$
How to perform Approximate Arithmetic on HE?

1.234 * 0.689 * 2.194 * 0.917 * 3.323 * 4.154 * 0.489 * 3.772 = 43.555

Linear Bit size of Ciphertext Modulus: \( \log q = O(\text{depth} \times \text{precision}) \)

\[
\begin{align*}
8,499 & \xrightarrow{\text{HomMult}} 171,042,375 \\
20,125 & \xrightarrow{\text{HomMult}} 17,103 \\
138,021 & \xrightarrow{\text{HomMult}} 4,355,518,392 \\
18,451 & \xrightarrow{\text{HomMult}} 435,552 \\
\end{align*}
\]
Functionality of Approximate HE

Packing Technique

- \( R = \mathbb{Z}[X] / (\Phi_m(X)) \).
- \( \Phi(X) = \prod_i (X - \zeta_i) \) where \( \zeta_i \)'s are \( m \)-th roots of unity.
- Encoding map: from \((M_i)_i\) to \( M(X) \) such that \( M(\zeta_i) = M_i \).
Functionality of Approximate HE

Packing Technique
- \( R = \mathbb{Z}[X] / (\Phi_m(X)) \).
- \( \Phi(X) = \prod_i (X - \zeta_i) \) where \( \zeta_i \)'s are m-th roots of unity.
- Encoding map: from \( (M_i)_i \) to \( M(X) \) such that \( M(\zeta_i) = M_i \).

Rotation, Conjugation
- Evaluation of \( \sigma(X) = X^k \) in \( \text{Gal}( K = \mathbb{Q}[X]/(X^N+1) / \mathbb{Q} ) \).
- Based on the key-switching technique.
Functionality of Approximate HE

Packing Technique
- \( R = \mathbb{Z}[X] / (\Phi_m(X)) \).
- \( \Phi(X) = \prod_i (X - \zeta_i) \) where \( \zeta_i \)'s are m-th roots of unity.
- Encoding map: from \((M_i)_i\) to \(M(X)\) such that \(M(\zeta_i) = M_i\)

Rotation, Conjugation
- Evaluation of \( \sigma(X) = X^k \) in \( \text{Gal}(K=\mathbb{Q}[X]/(X^n+1)/\mathbb{Q}) \).
- Based on the key-switching technique.

Evaluation of Analytic Functions
- \( \exp(z) \),
- \( z^{-1} \)
Bootstrapping for the Approximate HE (EC’18)

Decryption circuit

- $M = \langle ct, sk \rangle \pmod{q}$.
- Goal: Represent modular reduction as a circuit over the complex numbers.
Bootstrapping for the Approximate HE (EC’18)

Decryption circuit

- $M = \langle ct, sk \rangle \pmod{q}$.
- Goal: Represent modular reduction as a circuit over the complex numbers.
Bootstrapping for the Approximate HE (EC’18)

Decryption circuit

- $M = \langle ct, sk \rangle \pmod q$.  
  $M \approx (q/2\pi) \sin \theta$,  
  $\theta = (2\pi/q) \langle ct, sk \rangle$.

- Goal: Represent modular reduction as a circuit over the complex numbers.
Bootstrapping for the Approximate HE (EC’18)

Decryption circuit

- $M = \langle ct, sk \rangle \pmod{q}$.  
- $M \approx \left(\frac{q}{2\pi}\right) \sin \theta$,  
- $\theta = \left(\frac{2\pi}{q}\right) \langle ct, sk \rangle$.

- Goal: Represent modular reduction as a circuit over the complex numbers.

Evaluation of sine

- $\cos \theta = \cos^2(\theta/2) - \sin^2(\theta/2)$,  
- $\sin \theta = 2 \cos(\theta/2) \sin(\theta/2)$. 
Bootstrapping for the Approximate HE (EC’18)

Decryption circuit

- $M = \langle ct, sk \rangle \pmod{q}$. 
  
  $M \approx \left(\frac{q}{2\pi}\right) \sin \theta$, 
  
  $\theta = \left(\frac{2\pi}{q}\right) \langle ct, sk \rangle$. 

- Goal: Represent modular reduction as a circuit over the complex numbers.

Evaluation of sine

- $\cos \theta = \cos^2(\theta/2) - \sin^2(\theta/2)$, 
  
  $\sin \theta = 2 \cos(\theta/2) \sin(\theta/2)$. 

- From $[-2K\pi/2^r, 2K\pi/2^r]$ to $[-2K\pi, 2K\pi]$.

- **Linear** Complexity for Modulus Reduction Operation!

- $\langle ct', sk \rangle \pmod{Q} \approx M$. 

- $\langle ct', sk \rangle (\mod Q) \approx M$.
Following Work

Privacy-preserving Training of Logistic Regression Model

- Kim-Song-Wang-Xia-Jiang, JMIR Med Inform’18
  
  e.g. Six minutes to obtain a LR model from dataset of size 1579 * (18+1).

- (ongoing) ML based on the financial data with Bootstrapping.
Following Work

Privacy-preserving Training of Logistic Regression Model
- Kim-Song-Wang-Xia-Jiang, JMIR Med Inform’18
  e.g. Six minutes to obtain a LR model from dataset of size 1579 * (18+1).
- (ongoing) ML based on the financial data with Bootstrapping.

A Full-RNS Variant of Approx-HE
- Double-CRT (RNS+NTT) representation.
- Implementation without CRT composition or big-integer library.
- Based on the use of approximate basis & approximate modulus switching.
Following Work

Privacy-preserving Training of Logistic Regression Model
- Kim-Song-Wang-Xia-Jiang, JMIIR Med Inform’18
  e.g. Six minutes to obtain a LR model from dataset of size 1579 * (18+1).
- (ongoing) ML based on the financial data with Bootstrapping.

A Full-RNS Variant of Approx-HE
- Double-CRT (RNS+NTT) representation.
- Implementation without CRT composition or big-integer library.
- Based on the use of approximate basis & approximate modulus switching.

Open problems??
Homomorphic Encryption Framework
Homomorphic Encryption Framework

\[ m_1 \]

\[ m_2 \]

\[ 0 \]
Homomorphic Encryption Framework (Encryption)

\[ \text{Enc}(m_1) \xrightarrow{m_1} \]
\[ \text{Enc}(m_2) \xrightarrow{m_2} \]

\[ 0 \]
Homomorphic Encryption Framework (Addition)

$$m_3 = m_1 + m_2$$
Homomorphic Encryption Framework (Addition)

\[ m_3 = m_1 + m_2 \]

\[ \text{Enc}(m_1) + \text{Enc}(m_2) \]
Homomorphic Encryption Framework (Addition)

\[ m_3 = m_1 + m_2 \]

\[ \text{Enc}(m_1) + \text{Enc}(m_2) \]
Homomorphic Encryption Framework (Multiplication)

\[ \text{Enc}(m_1) \times \text{Enc}(m_2) = \text{Enc}(m_1 \cdot m_2) \]
Homomorphic Encryption Framework (Multiplication)

\[ \text{Enc}(m_1) \times \text{Enc}(m_2) = \text{Enc}(m_1 \times m_2) \]
Homomorphic Encryption Framework (Multiplication)

\[ m_3 = m_1 \times m_2 \]

\[ \text{Enc}(m_1) \times \text{Enc}(m_2) \]
Homomorphic Encryption Framework (Multiplication)

\[ m_3 = m_1 \cdot m_2 \]

\[ \text{Enc}(m_1) \cdot \text{Enc}(m_2) \]

\[ \text{Enc}(m_1) \]

\[ m_1 \]

\[ m_2 \]

\[ \text{Enc}(m_2) \]

\[ 0 \]
Homomorphic Encryption Framework (Multiplication)

\[ m_3 = m_1 \times m_2 \]
Approximate Homomorphic Encryption
Approximate Homomorphic Encryption
Approximate Homomorphic Encryption
Approximate Homomorphic Encryption (Encryption)

\[ m_1' = m_1 + e_1 \]

\[ m_2' = m_2 + e_2 \]
Approximate Homomorphic Encryption (Operations)
Approximate Homomorphic Encryption (Operations)
Approximate Homomorphic Encryption (Operations)

\[ m_2 + m_2' \approx m_1 + m_2 \]
Approximate Homomorphic Encryption (Operations)

\[ m_1 + m_2 = m_1' + m_2' \]
Approximate Homomorphic Encryption (Operations)
Approximate Homomorphic Encryption (Operations)

\[ m_1' \cdot m_2' \approx m_1 \cdot m_2 \]
Approximate Homomorphic Encryption (Operations)
Approximate Homomorphic Encryption (Operations)

\[ m_1' \times m_2' \]

\[ \frac{1}{4} m_1' \times m_2' \approx \frac{1}{4} m_1 \times m_2 \]

\[ m_1 + m_2 \]

\[ m_1' + m_2' \]