Practical Packing Method in Somewhat Homomorphic Encryption

Masaya Yasuda, Takeshi Shimoyama, Jun Kogure (Fujitsu Laboratories Ltd.), Kazuhiro Yokoyama (Rikkyo University), and Takeshi Koshiba (Saitama University)
Homomorphic Encryption (HE)

- **Public-key encryption supporting “some operations” on encrypted data**
  - It enables us to compute the total score so that the cloud cannot know any score.

- **Additively HE**
  - Practical in performance, but it can only support addition (e.g., Paillier scheme).
  - Limited applications (e.g., e-voting).

- **FHE (fully HE)**
  - It supports any operations, and is expected to be applied to cloud computing.
  - **But, difficulty on performance and size**

---

Summary of Our Work

- **SHE (Somewhat HE)**
  - It supports “a limited number” of additions and multiplications
    - Well known as a building block for the FHE construction
  - Much faster and shorter than FHE
    - (Performance and Size) Additively HE < SHE << FHE

- **Motivation**: practical use of SHE in wider applications

- **Contributions**:
  - New packing method in SHE for the practical use
    - Use of the ring-LWE based LWE scheme
    - Our method can pack a vector into a single ciphertexts
    - It gives several efficient computations over packed ciphertexts
  - Application to privacy-preserving biometrics
    - Our method gives faster performance and shorter size for secret authentication in the HE approach
Ring-LWE Based SHE Scheme

**Key Generation**
- Secret key \( sk = s \), Public key \( pk = (a_{\downarrow 0}, a_{\downarrow 1}) \)
  - \( a_{\downarrow 0} = -(a_{\downarrow 1} s + te) \) in \( R \downarrow q \), \( s, a_{\downarrow 1}, e \): small noises in \( R \downarrow q \)
  - \( R = \mathbb{Z}[x]/(x^n + 1) \): base ring, \( t \): plaintext modulus, \( q \): ciphertext modulus

**Encryption**
- For a plaintext \( m \in R \downarrow t \), \( Enc(m, pk) = (c_{\downarrow 0}, c_{\downarrow 1}) \in (R \downarrow q)^2 \)
  - \( c_{\downarrow 0} = a_{\downarrow 0} u + tg + m \), \( c_{\downarrow 1} = a_{\downarrow 1} u + tf \)
  - \( u, g, f \): small noises in \( R \downarrow q \)

**Homomorphic Operations**
- [Add] \( Enc(m, pk) + Enc(m', pk) := (c_{\downarrow 0} + c'_{\downarrow 0}, c_{\downarrow 1} + c'_{\downarrow 1}) \)
- [Mul] \( Enc(m, pk) \ast Enc(m', pk) := (c_{\downarrow 0} \cdot c'_{\downarrow 0}, c_{\downarrow 0} \cdot c'_{\downarrow 1} + c'_{\downarrow 0} \cdot c_{\downarrow 1}, c_{\downarrow 1} \cdot c'_{\downarrow 1}) \)

**Decryption**
- For a ciphertext \( ct = (c_{\downarrow 0}, c_{\downarrow 1}, \ldots, c_{\downarrow k}) \), \( Dec(ct, sk) = \left[ \sum c_{\uparrow i} s_{\uparrow i} \right] q \mod t \)
  - \( [a]q \): \( a \) modulo \( q \) in \([−q/2, q/2)\)
Introduction of Our Packing Method

**Strategy**
1. Transform a vector of length $n$ to a certain polynomial in $R$
2. Pack its polynomial into a single ciphertext

**Our Trick: two types of polynomials in $R = \mathbb{Z}[x]/(x^n + 1)$**
- **[Type1]** $A = (A \downarrow 0, \ldots, A \downarrow n-1) \rightarrow F\downarrow 1 (A) := \sum A_i x^i$ (ascending order)
- **[Type2]** $B = (B \downarrow 0, \ldots, B \downarrow n-1) \rightarrow F\downarrow 2 (B) := -\sum B_i x^{n-i}$ (descending order)

**Packed ciphertexts**
- **[Type1]** $v\text{Enc}\downarrow 1 (A) := \text{Enc}(F\downarrow 1 (A), \text{pk})$
- **[Type2]** $v\text{Enc}\downarrow 2 (B) := \text{Enc}(F\downarrow 2 (B), \text{pk})$
  - It does not change the security level of the SHE scheme
Efficient Secure Inner Product

**Only one** homomorphic multiplication
\(v\text{Enc}_1(A) \times v\text{Enc}_2(B)\)
over packed ciphertexts gives the inner product between \(A\) and \(B\) on encrypted data.

Sketch of Proof:

\[ F \downarrow_1(A) \times F \downarrow_2(B) \]
over plaintexts

\[ F \downarrow_1(A) = A \downarrow_0 + A \downarrow_1 x + A \downarrow_2 x \uparrow_2 + \cdots + A \downarrow_{n-1} x \uparrow_{n-1} \]

\[ F \downarrow_2(B) = -B \downarrow_{n-1} x - \cdots - B \downarrow_1 x \uparrow_{n-1} - B \downarrow_0 x \uparrow_n \]

\[ F \downarrow_1(A) \times F \downarrow_2(B) = -(A \downarrow_0 B \downarrow_0 + A \downarrow_1 B \downarrow_1 + \cdots A \downarrow_{n-1} B \downarrow_{n-1}) x \uparrow_n + \cdots = -1 \text{ in } R \]

The constant term gives our desired inner product.
Computations over Packed Ciphertexts

Combinations of our packed ciphertexts also give us the following computations:

- **Private Statistic:**
  - Sum, and Mean
  - Variance, and Standard deviation

- **Statistical Analysis:**
  - Covariance
  - Correlation

- **Distances**
  - Hamming distance
  - Euclid distance

**etc.,....**

**Remark:** It is difficult to perform computations such as
- the median,
- the 1-norm distance

since we generally can not compare two values without decryption in the use of homomorphic encryption (irrespective of packing method).
Privacy-Preserving Biometrics

- Biometric authentication with protecting the privacy of biometric data

Homomorphic Encryption Approach

- **Secure Hamming distance**: metric to measure the similarity of feature vectors $A, B$ on encrypted data
  - $d_{\downarrow H}(A, B) = \sum_{i=0}^{2047} (A_{\downarrow i} - B_{\downarrow i})^2 = \sum_{i=0}^{2047} (A_{\downarrow i} + B_{\downarrow i} - 2A_{\downarrow i} \cdot B_{\downarrow i})$
  - The authentication result is “OK” if $d_{\downarrow H}(A, B) < \sigma$

Assume 2048-bit for various biometrics (e.g., vein codes)
Since all computations are performed on encrypted data, we hope that we would use “the cloud” as the computation server.

* Enrollment phase
- Generate a feature vector $A$
- Packed Encryption of type 1
- Secure Hamming distance over packed ciphertexts

* Authentication phase
- Generate a feature vector $B$
- Packed Encryption of type 2
- Decrypt and return the result

OK if $d_{\Delta H} < \sigma$
Comparison with Related Work

Table: A comparison on the performance and the encrypted data size

<table>
<thead>
<tr>
<th>Protocols (feature vector size)</th>
<th>Performance of Secure Hamming</th>
<th>Size increase rate by encryption (cipher. size)</th>
<th>Homomorphic encryption scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCiFI [25] (900-bit)</td>
<td>310 ms&lt;sup&gt;(a)&lt;/sup&gt;</td>
<td>2048 times (230 KByte)</td>
<td>Paillier-1024 (additive scheme)</td>
</tr>
<tr>
<td>Protocol of [2] (2048-bit)</td>
<td>150 ms&lt;sup&gt;(b)&lt;/sup&gt;</td>
<td>1024 times (262 KByte)</td>
<td>DGK-1024 (additive scheme)</td>
</tr>
<tr>
<td>Previous work&lt;sup&gt;†&lt;/sup&gt; [31]</td>
<td>18.10 ms&lt;sup&gt;(c)&lt;/sup&gt;</td>
<td>about 80 times (19 KByte)</td>
<td>ideal lattices-4096 (SHE)</td>
</tr>
<tr>
<td>This work (2048-bit)</td>
<td>5.31 ms&lt;sup&gt;(c)&lt;/sup&gt;</td>
<td>about 120 times (31 KByte)</td>
<td>ring-LWE-2048 (SHE)</td>
</tr>
</tbody>
</table>

<sup>†</sup> denotes the ratio of (encrypted feature vector size)/(plain feature vector size)
<sup>‡</sup> uses a similar packing method as in this work
(a) on an 8 core machine of 2.6 GHz AMD Opteron processors with 1 GByte memory
(b) on an Intel Core 2 Duo 2.13 GHz with 3 GByte memory
(c) on an Intel Xeon X3480 at 3.07 GHz with 16 GByte memory

Only one feature vector is protected
Two feature vectors are protected, which condition is tighter

Conclusions and Future Work

Conclusions
- We proposed a packing method in the ring-LWE based SHE scheme
  - Main trick: two types of packed ciphertexts
  - It gives us an efficient computation of secure inner product
- Our method gave a practical solution in privacy-preserving biometrics
  - It takes 5.31 ms for secure Hamming distance of 2048-bit vectors
  - Faster than additively schemes even under tighter condition
  - Lattice-based SHE with our packing method would be practical

Future Work
- Application of our packing method in wider applications
  - e.g., secure pattern matching for secret search
shaping tomorrow with you