Fully Homomorphic Encryption

Francisco Vial-Prado

ASCrypto - LatinCrypt '19

IMFD Chile, Ecole Polytechnique, Université Paris-Saclay
Applied Cryptography @ ProtonMail
Overview

- Generic homomorphic encryption, *a priori* observations
- Gentry’s blueprint
- Second and third generation schemes
On Data Banks And Privacy Homomorphisms - 1978

- ... a system working with encrypted data can at most store or retrieve data for the user; any more complicated operations seem to require that the data be decrypted before being operated on.

- ... it appears likely that there exist [...] Privacy Homomorphisms.
Privacy Homomorphisms

Find an encryption scheme $S$ such that:

Let $y = S$.Enc$_k(x)$. For any PPT function $f$ mapping plaintexts to plaintexts, find $y'$ publicly such that $S$.Dec$_k(y') = f(x)$.

Example: If $S$.plainspace is a ring, provide functionalities Add, Mult such that

$$\text{Add} (\text{Enc}(x), \text{Enc}(y)) \text{ encrypts } x + y$$

$$\text{Mult} (\text{Enc}(x), \text{Enc}(y)) \text{ encrypts } x \times y.$$

Disclaimer

Along with reasonable security properties!
Privacy Homomorphisms

Find an encryption scheme $S$ such that:

Let $y = S . Enc_k(x)$. For any PPT function $f$ mapping plaintexts to plaintexts, find $y'$ publicly such that $S . Dec_k(y') = f(x)$.

Example: If $S . \text{plaintext space}$ is a ring, provide functionalities Add, Mult such that

$$\text{Add}(Enc(x), Enc(y)) \text{ encrypts } x + y$$

$$\text{Mult}(Enc(x), Enc(y)) \text{ encrypts } x \times y.$$
Privacy Homomorphisms

Find an encryption scheme $S$ such that:

Let $y = S.\text{Enc}_k(x)$. For any PPT function $f$ mapping plaintexts to plaintexts, find $y'$ publicly such that $S.\text{Dec}_k(y') = f(x)$.

Example: If $S.\text{plaintext}$ is a ring, provide functionalities Add, Mult such that

$$\text{Add} (\text{Enc}(x), \text{Enc}(y))$$ encrypts $x + y$

$$\text{Mult} (\text{Enc}(x), \text{Enc}(y))$$ encrypts $x \times y$.

Disclaimer
Along with reasonable security properties!
A priori observations
HE is non determinist

1. Homomorphic encryption must be non-determinist

The attacker could solve ring equations

\[ x = k \iff (x \neq 0) \land (x^2 = \underbrace{x + x + \cdots + x}_{k \text{ times}}) \]

1bis. Broccoli heuristics: If ciphertext spaces are distinguishable, they should be somewhat separable.
1. Homomorphic encryption must be non-determinist
The attacker could solve ring equations

\[ x = k \iff (x \neq 0) \land (x^2 = x + x + \cdots + x) \]

1bis. Broccoli heuristics: If ciphertext spaces are distinguishable, they should be somewhat separable.
2. Logical conditions translate to homomorphic comparison circuits.

Consider the equality circuit: Let $a, b \in \{0, 1\}^\kappa$.

$$\text{Eq}(a, b) = 1 \oplus \prod_{i=1}^{\kappa} (a_i \oplus b_i \oplus 1) = \begin{cases} 0 & \text{if } a = b, \\ 1 & \text{if } a \neq b. \end{cases}$$
3.– Decrypt **Verifiable Computations Only** If Possible (Homomorphic encryption schemes are known to be vulnerable to IND-CCA Key-Recovery attacks)
Connections with other cryptographic problems

- (implied by) Functional encryption
- (provides reduction of) Secure Multiparty Computation
- (compatible with) Identity/Attribute-Based Encryption
- (brick of?) Indistinguishability Obfuscation
- (first multi-hop?) Proxy Re-encryption
Connections with other cryptographic problems

- (implied by) Functional encryption
- (provides reduction of) Secure Multiparty Computation
- (compatible with) Identity/Attribute-Based Encryption
- (brick of?) Indistinguishability Obfuscation
- (first multi-hop?) Proxy Re-encryption
Connections with other cryptographic problems

- (implied by) Functional encryption
- (provides reduction of) Secure Multiparty Computation
- (compatible with) Identity/Attribute-Based Encryption
- (brick of?) Indistinguishability Obfuscation
- (first multi-hop?) Proxy Re-encryption
Connections with other cryptographic problems

- (implied by) Functional encryption
- (provides reduction of) Secure Multiparty Computation
- (compatible with) Identity/Attribute-Based Encryption
- (brick of?) Indistinguishability Obfuscation
- (first multi-hop?) Proxy Re-encryption
Connections with other cryptographic problems

- (implied by) Functional encryption
- (provides reduction of) Secure Multiparty Computation
- (compatible with) Identity/Attribute-Based Encryption
- (brick of?) Indistinguishability Obfuscation
- (first multi-hop?) Proxy Re-encryption
The Sophomore’s Dream

Let \( R \) be some ring and \( I \) be an ideal of \( R \). Let \( m \in R/I \). Let 
\[
\text{Enc}(m) := m + i \quad \text{where} \quad i \in I \quad \text{is sampled randomly.}
\]

\[
\text{Enc}(m_1) + \text{Enc}(m_2) = m_1 + m_2 + i',
\]

\[
\text{Enc}(m_1) \times \text{Enc}(m_2) = m_1 \times m_2 + i''.
\]

Good game; now look for

- Random efficient sampling from \( \alpha + I \) for every \( \alpha \in R/I \)
- Secret decryption power: ideal annihilation procedure
  \( \alpha + xI \mapsto \alpha \).
- Connection to hard problems.
The Sophomore’s Dream

Let $R$ be some ring and $I$ be an ideal of $R$. Let $m \in R/I$. Let $\text{Enc}(m) := m + i$ where $i \in I$ is sampled randomly.

$\text{Enc}(m_1) + \text{Enc}(m_2) = m_1 + m_2 + i'$,
$\text{Enc}(m_1) \times \text{Enc}(m_2) = m_1 \times m_2 + i''$.

Good game; now look for

- Random efficient sampling from $\alpha + I$ for every $\alpha \in R/I$
- Secret decryption power: ideal annihilation procedure $\alpha + xI \mapsto \alpha$.
- Connection to hard problems.
The Sophomore’s Dream

Let $R$ be some ring and $I$ be an ideal of $R$. Let $m \in R/I$. Let $\text{Enc}(m) := m + i$ where $i \in I$ is sampled randomly.

$\text{Enc}(m_1) + \text{Enc}(m_2) = m_1 + m_2 + i'$,
$\text{Enc}(m_1) \times \text{Enc}(m_2) = m_1 \times m_2 + i''$.

Good game; now look for

- Random efficient sampling from $\alpha + I$ for every $\alpha \in R/I$
- Secret decryption power: ideal annihilation procedure $\alpha + xI \mapsto \alpha$.
- Connection to hard problems.
The Sophomore’s Dream

Let $R$ be some ring and $I$ be an ideal of $R$. Let $m \in R/I$. Let $\text{Enc}(m) := m + i$ where $i \in I$ is sampled randomly.

$\text{Enc}(m_1) + \text{Enc}(m_2) = m_1 + m_2 + i'$,

$\text{Enc}(m_1) \times \text{Enc}(m_2) = m_1 \times m_2 + i''$.

Good game; now look for

- Random efficient sampling from $\alpha + I$ for every $\alpha \in R/I$
- Secret decryption power: ideal annihilation procedure
  $\alpha + xI \mapsto \alpha$.
- Connection to hard problems.
Gentry’s solution

The Sophomore’s Dream

Let $R$ be some ring and $I$ be an ideal of $R$. Let $m \in R/I$. Let $\text{Enc}(m) := m + i$ where $i \in I$ is sampled randomly.

$\text{Enc}(m_1) + \text{Enc}(m_2) = m_1 + m_2 + i'$,
$\text{Enc}(m_1) \times \text{Enc}(m_2) = m_1 \times m_2 + i''$.

Good game; now look for

- Random efficient sampling from $\alpha + I$ for every $\alpha \in R/I$
- Secret decryption power: ideal annihilation procedure
  $\alpha + xI \mapsto \alpha$.
- Connection to hard problems.
Ideals $+$ Lattices $=$ Ideal Lattices

Gentry’s first FHE scheme

Specialized the latter construction using polynomial rings and two sets of ideal lattices.

Secret and public keys are parallelepipeds in $\mathbb{R}^n$, with large $n$, and plaintexts/ciphertexts are polynomials in $\mathbb{Z}[X]/(X^n - 1)$. 
Generic homomorphic encryption
Gentry’s blueprint
Second generation

Ideals + Lattices = Ideal Lattices

Gentry’s first FHE scheme

Specialized the latter construction using polynomial rings and two sets of ideal lattices.

Secret and public keys are parallelepipeds in $\mathbb{R}^n$, with large $n$, and plaintexts/ciphertexts are polynomials in $\mathbb{Z}[X]/(X^n - 1)$. 
Disclaimer

What follows is an Unfair and Informal and Incomplete Description of Gentry’s scheme
More on lattices on yesterdays’ talk:

*Engineering lattice-based crypto – Peter Schwabe*

\[ \mathcal{L} = \mathbb{Z} \cdot b_1 + \mathbb{Z} \cdot b_2 \]

\( B = \{b_1, b_2\} \) is called a basis of \( \mathcal{L} \).
More on lattices on yesterdays’ talk:
*Engineering lattice-based crypto – Peter Schwabe*

\[ \mathcal{L} = \mathbb{Z} \cdot b_1 + \mathbb{Z} \cdot b_2 \]

\( B = \{b_1, b_2\} \) is called a basis of \( \mathcal{L} \).
More on lattices on yesterdays’ talk:
*Engineering lattice-based crypto – Peter Schwabe*

\[ \mathcal{L} = \mathbb{Z} \cdot b_1 + \mathbb{Z} \cdot b_2 \]

\( \mathbf{B} = \{b_1, b_2\} \) is called a basis of \( \mathcal{L} \).
\( \mathbf{B} = \mathbf{U} \cdot \mathbf{B}' \) for \( \mathbf{U} \in \text{GL}_n(\mathbb{Z}) \).

In particular, for any base, \( \det(\mathcal{L}) := \sqrt{\det(\mathbf{B} \cdot \mathbf{B}^t)} \).
Lattices

\[ \mathbf{B} = U \cdot \mathbf{B}' \text{ for } U \in \text{GL}_n(\mathbb{Z}). \]

In particular, for any base,

\[ \det(\mathcal{L}) := \sqrt{\det(\mathbf{B} \cdot \mathbf{B}^t)}. \]
Lattices

\[ \mathbf{B} = U \cdot \mathbf{B}' \text{ for } U \in \text{GL}_n(\mathbb{Z}). \]

In particular, for any base,

\[ \det(\mathcal{L}) := \sqrt{\det(\mathbf{B} \cdot \mathbf{B}^t)}. \]
\[ P(B) := \left[ -\frac{1}{2}, \frac{1}{2} \right] \cdot b_1 + \left[ -\frac{1}{2}, \frac{1}{2} \right] \cdot b_2 \]

\[ \text{Vol}(P) = \det(L) \]
\( P(B) := \left[ -\frac{1}{2}, \frac{1}{2} \right] \cdot b_1 + \left[ -\frac{1}{2}, \frac{1}{2} \right] \cdot b_2 \)

\[ \text{Vol}(P) = \det(L) \]
\[ \mathcal{P}(\mathcal{B}) := \left[ -\frac{1}{2}, \frac{1}{2} \right] \cdot b_1 + \left[ -\frac{1}{2}, \frac{1}{2} \right] \cdot b_2 \]

\[ \text{Vol}(\mathcal{P}) = \det(\mathcal{L}) \]
\[ \forall x \in \mathbb{R}^n \quad x \mod B := x - B[B^{-1} \cdot x] \]
A message $m = (1, 0, 0, 0, 1, 1)$ is encrypted by

$$c = m \mod B_{pk}.$$ 

Then,

$$c = (1, 3, 0, -2, 0, -521159786514568)$$

is decrypted by

$$m = c \mod B_{sk}.$$
A message $m = (1, 0, 0, 0, 1, 1)$ is encrypted by

$$c = m \mod B_{pk}.$$  

Then,

$$c = (1, 3, 0, -2, 0, -521159786514568)$$

is decrypted by

$$m = c \mod B_{sk}.$$
Concretely:

Let $p \in \mathbb{Z}[X]/(X^n - 1)$. Then

$$B_{sk} = \{p(x), xp(x), x^2 p(x), \ldots, x^{n-1} p(x)\}$$

In order to decrypt a ciphertext $c = (c_0, \ldots, c_{n-1})$,

$$c \mod B_{sk} = c - B_{sk} \cdot \lfloor B_{sk}^{-1} \cdot c \rfloor \quad \text{(in } \mathbb{Z}^n)$$

$$= c(x) - p(x) \cdot \lfloor p(x)^{-1} \cdot c(x) \rfloor \quad \text{(in } \mathbb{Z}[X]/X^n - 1)$$
Gentry’s scheme

Concretely:

Let \( p \in \mathbb{Z}[X]/(X^n - 1) \). Then

\[
B_{sk} = \{ p(x), xp(x), x^2 p(x), \ldots, x^{n-1} p(x) \}
\]

In order to decrypt a ciphertext \( c = (c_0, \ldots, c_{n-1}) \),

\[
c \mod B_{sk} = c - B_{sk} \cdot \lfloor B_{sk}^{-1} \cdot c \rfloor \quad \text{(in } \mathbb{Z}^n)\]

\[
= c(x) - p(x) \cdot \lfloor p(x)^{-1} \cdot c(x) \rfloor \quad \text{(in } \mathbb{Z}[X]/X^n - 1)\).
\]
Gentry’s scheme

Concretely:

Let \( p \in \mathbb{Z}[X]/(X^n - 1) \). Then

\[
B_{sk} = \{ p(x), xp(x), x^2 p(x), \ldots, x^{n-1} p(x) \}
\]

In order to decrypt a ciphertext \( c = (c_0, \ldots, c_{n-1}) \),

\[
c \mod B_{sk} = c - B_{sk} \cdot \lfloor B_{sk}^{-1} \cdot c \rfloor \quad \text{(in } \mathbb{Z}^n \text{)}
\]

\[
= c(x) - p(x) \cdot \lfloor p(x)^{-1} \cdot c(x) \rfloor \quad \text{(in } \mathbb{Z}[X]/X^{n-1} \text{)}.
\]
Gentry’s scheme

Homomorphic operations? Ring structure transport from $R = \mathbb{Z}[X]/(P(X))$, to $\mathbb{Z}^n$ via the coefficients homomorphism.
The noise problem and Gentry’s Glovebox

Encryption $m + xl$ is subject to the 'size' of $x$. After a threshold, decryption breaks.

Bootstrapping operation: *Homomorphically decrypt*
The noise problem and Gentrys’ Glovebox

Encryption $m + xl$ is subject to the 'size' of $x$. After a threshold, decryption breaks.

Bootstrapping operation: *Homomorphically decrypt*
Second and third gen schemes
Second and third generation schemes

Same blueprint

- Provide Add, Mult operations, bootstrap to reduce noise, repeat

Improved efficiency and security

- RLWE, NTRU-based, Approximate Eigenvectors
- Better noise growth, key sizes, ciphertext compression, ciphertext packing, SIMD style
- Efficient bootstrapping

New flavors, properties, and already practical for applications.
Second and third generation schemes

Same blueprint
- Provide Add, Mult operations, bootstrap to reduce noise, repeat

Improved efficiency and security
- RLWE, NTRU-based, Approximate Eigenvectors
- Better noise growth, key sizes, ciphertext compression, ciphertext packing, SIMD style
- Efficient bootstrapping

New flavors, properties, and already practical for applications.
Second and third generation schemes

Same blueprint

- Provide Add, Mult operations, bootstrap to reduce noise, repeat

Improved efficiency and security

- RLWE, NTRU-based, Approximate Eigenvectors
  - Better noise growth, key sizes, ciphertext compression, ciphertext packing, SIMD style
  - Efficient bootstrapping

New flavors, properties, and already practical for applications.
Second and third generation schemes

Same blueprint

- Provide Add, Mult operations, bootstrap to reduce noise, repeat

Improved efficiency and security

- RLWE, NTRU-based, Approximate Eigenvectors
- Better noise growth, key sizes, ciphertext compression, ciphertext packing, SIMD style
- Efficient bootstrapping

New flavors, properties, and already practical for applications.
Second and third generation schemes

Same blueprint
- Provide Add, Mult operations, bootstrap to reduce noise, repeat

Improved efficiency and security
- RLWE, NTRU-based, Approximate Eigenvectors
- Better noise growth, key sizes, ciphertext compression, ciphertext packing, SIMD style
- Efficient bootstrapping

New flavors, properties, and already practical for applications.
Second and third generation schemes

Same blueprint

- Provide Add, Mult operations, bootstrap to reduce noise, repeat

Improved efficiency and security

- RLWE, NTRU-based, Approximate Eigenvectors
- Better noise growth, key sizes, ciphertext compression, ciphertext packing, SIMD style
- Efficient bootstrapping

New flavors, properties, and already practical for applications.
Regev’s folklore example: Recover an integer vector \( s = (s_1, s_2, s_3, s_4) \in \mathbb{Z}^4_{17} \) satisfying

\[
\begin{align*}
14s_1 + 15s_2 + 5s_3 + 2s_4 & \approx 8 \pmod{17}, \\
13s_1 + 14s_2 + 14s_3 + 6s_4 & \approx 16 \pmod{17}, \\
6s_1 + 10s_2 + 13s_3 + 1s_4 & \approx 3 \pmod{17}, \\
10s_1 + 4s_2 + 12s_3 + 16s_4 & \approx 12 \pmod{17}, \\
9s_1 + 5s_2 + 9s_3 + 6s_4 & \approx 9 \pmod{17}, \\
3s_1 + 6s_2 + 4s_3 + 5s_4 & \approx 16 \pmod{17},
\end{align*}
\]

where “\( \approx \)” means that the equation is correct up to an error of \( \pm 1 \).

BGV (2011) FHE scheme
Let $\chi$ be an error distribution over $R = \mathbb{F}_q[X]/(P_n(X))$. Let $s_i(x) \leftarrow \chi$ and for $i = 0, 1, 2, \ldots$, $a_i(x) \leftarrow R$, $s_i \leftarrow \chi$. Finally, let $b_i := a_i \cdot s + e_i$.

### Search-RLWE

Guess $s$ given a list of pairs $(a_i, b_i) = (a_i, a_i \cdot s + e_i)$.

### Decision-RLWE

Given a list of pairs $(a_i(x), b_i(x))$, decide whether the $b_i$’s were sampled randomly, or constructed as above.

### BFV (2012) FHE scheme - with new techniques

→ See LatinCrypt’19 - Compact and simple RLWE based key encapsulation mechanism
Let $\chi$ be an error distribution over $R = \mathbb{F}_q[X]/(P_n(X))$. Let $s_i(x) \leftarrow \chi$ and for $i = 0, 1, 2, \ldots$, $a_i(x) \leftarrow R$, $s_i \leftarrow \chi$. Finally, let $b_i := a_i \cdot s + e_i$.

**Search-RLWE**

Guess $s$ given a list of pairs $(a_i, b_i) = (a_i, a_i \cdot s + e_i)$.

**Decision-RLWE**

Given a list of pairs $(a_i(x), b_i(x))$, decide whether the $b_i$’s were sampled randomly, or constructed as above.

**BFV (2012) FHE scheme - with new techniques**

→ See LatinCrypt’19 - Compact and simple RLWE based key encapsulation mechanism
Let $\chi$ be an error distribution over $R = \mathbb{F}_q[X]/(P_n(X))$. Let $s_i(x) \leftarrow \chi$ and for $i = 0, 1, 2, \ldots$, $a_i(x) \leftarrow R$, $s_i \leftarrow \chi$. Finally, let $b_i := a_i \cdot s + e_i$.

**Search-RLWE**

Guess $s$ given a list of pairs $(a_i, b_i) = (a_i, a_i \cdot s + e_i)$.

**Decision-RLWE**

Given a list of pairs $(a_i(x), b_i(x))$, decide whether the $b_i$’s were sampled randomly, or constructed as above.

BFV (2012) FHE scheme - with new techniques

→ See LatinCrypt’19 - Compact and simple RLWE based key encapsulation mechanism
Let $\chi$ be an error distribution over $R = \mathbb{F}_q[X]/(P_n(X))$. Let $s_i(x) \leftarrow \chi$ and for $i = 0, 1, 2, \ldots$, $a_i(x) \leftarrow R$, $s_i \leftarrow \chi$. Finally, let $b_i := a_i \cdot s + e_i$.

**Search-RLWE**

Guess $s$ given a list of pairs $(a_i, b_i) = (a_i, a_i \cdot s + e_i)$.

**Decision-RLWE**

Given a list of pairs $(a_i(x), b_i(x))$, decide whether the $b_i$’s were sampled randomly, or constructed as above.

**BFV (2012) FHE scheme - with new techniques**

→ See LatinCrypt’19 - Compact and simple RLWE based key encapsulation mechanism
Ring Learning With Errors

Let $\chi$ be an error distribution over $R = \mathbb{F}_q[X]/(P_n(X))$. Let $s_i(x) \leftarrow \chi$ and for $i = 0, 1, 2, \ldots$, $a_i(x) \leftarrow R$, $s_i \leftarrow \chi$. Finally, let $b_i := a_i \cdot s + e_i$.

**Search-RLWE**

Guess $s$ given a list of pairs $(a_i, b_i) = (a_i, a_i \cdot s + e_i)$.

**Decision-RLWE**

Given a list of pairs $(a_i(x), b_i(x))$, decide whether the $b_i$’s were sampled randomly, or constructed as above.

**BFV (2012) FHE scheme - with new techniques**

→ See LatinCrypt’19 - Compact and simple RLWE based key encapsulation mechanism
NTRU-based

$N$-th truncated: Security problems related to Gaussian distributions and inversions in polynomial rings. Exposed strong connections with MPC (LTV12 scheme)

Subfield lattice attacks on overstretched NTRU assumptions - ABD 2016.

→ Same ideas behind the new Mersenne cryptosystem (AJPS17), see LatinCrypt’19, Quantum LLL with an Application to Mersenne Number Cryptosystems
N-th truncated: Security problems related to Gaussian distributions and inversions in polynomial rings. Exposed strong connections with MPC (LTV12 scheme)

Subfield lattice attacks on overstretched NTRU assumptions - ABD 2016.

→ Same ideas behind the new Mersenne cryptosystem (AJPS17), see LatinCrypt’19, Quantum LLL with an Application to Mersenne Number Cryptosystems
$N$-th truncated: Security problems related to Gaussian distributions and inversions in polynomial rings. Exposed strong connections with MPC (LTV12 scheme)

Subfield lattice attacks on overstretched NTRU assumptions - ABD 2016.

→ *Same ideas behind the new Mersenne cryptosystem (AJPS17), see LatinCrypt’19, Quantum LLL with an Application to Mersenne Number Cryptosystems*
Third Generation

GSW and Approximate Eigenvectors

\[ C \cdot v = m \cdot v + e \mod q \]

- Asymmetric nose growth
- Bootstrapping after each gate - the homomorphic brick
- Ring variant and inspired optimizations: TorusFHE (https://tfhe.github.io/tfhe/)
Third Generation

GSW and Approximate Eigenvectors

\[ C \cdot v = m \cdot v + e \mod q \]

- Asymmetric nose growth
- Bootstrapping after each gate - the homomorphic brick
- Ring variant and inspired optimizations: TorusFHE ([https://tfhe.github.io/tfhe/](https://tfhe.github.io/tfhe/))
Third Generation

GSW and Approximate Eigenvectors

\[ C \cdot v = m.v + e \mod q \]

- Asymmetric nose growth
- Bootstrapping after each gate - the homomorphic brick
- Ring variant and inspired optimizations: TorusFHE (https://tfhe.github.io/tfhe/)
Conclusion

Thank you!
Conclusion

Thank you!