Fully Homomorphic Encryption

based on Craig Gentry's PhD thesis

Martin Hlaváč

Department of Algebra, Charles University in Prague

March 24, 2010, Spring School of Algebra

RSA homomorphic encryption

- plaintexts π_i
- ciphertexts $\psi_i = (\pi_i)^e \mod N$
- RSA is multiplicatively homomorphic

$$\Pi_i \psi_i = (\Pi_i (\pi_i))^e \mod N$$

RSA homomorphic encryption

- plaintexts π_i
- ciphertexts $\psi_i = (\pi_i)^e \mod N$
- RSA is multiplicatively homomorphic

$$\Pi_i \psi_i = (\Pi_i (\pi_i))^e \mod N$$

- we can compute arbitrary circuits consisting of MUL gates on ciphertexts

Fully Homomorphic Encryption

Our goal

- compute arbitrary circuits on encrypted data, i.e. circuits of ADD, MUL, XOR, etc. gates, or universal NAND or NOR gates.

Fully Homomorphic Encryption

Our goal

- compute arbitrary circuits on encrypted data, i.e. circuits of ADD, MUL, XOR, etc. gates, or universal NAND or NOR gates.
- Indeed NAND is a universal gate

Fully Homomorphic Encryption

We are looking for encryption scheme $\mathcal{E}$ that consists of

- $\text{KeyGen}_{\mathcal{E}}$ key-generation algorithm
- $\text{Encrypt}_{\mathcal{E}}$ encryption algorithm
- $\text{Decrypt}_{\mathcal{E}}$ decryption algorithm

Fully Homomorphic Encryption

We are looking for encryption scheme $\mathcal{E}$ that consists of

- $\text{KeyGen}_{\mathcal{E}}$ key-generation algorithm
- $\text{Encrypt}_{\mathcal{E}}$ encryption algorithm
- $\text{Decrypt}_{\mathcal{E}}$ decryption algorithm
- $\text{Evaluate}_{\mathcal{E}}$ evaluation algorithm

Fully Homomorphic Encryption

Algorithm Evaluate$_{\mathcal{E}}$ is efficient and

- for any public key pk, and
- for any circuit C, and
- for any ciphertexts $\psi_i = \text{Encrypt}_{\mathcal{E}}(pk, \pi_i)$

outputs

$$\psi \leftarrow \text{Evaluate}_{\mathcal{E}}(pk, C, \psi_1, \ldots, \psi_t)$$

a valid ciphertext of $C(\pi_1, \ldots, \pi_t)$
(i.e. $C(\pi_1, \ldots, \pi_t) = \text{Decrypt}_{\mathcal{E}}(sk, \psi)$)

Of course, we want to avoid trivial "do-it-yourself" solutions such as

$$\text{Evaluate}_{\mathcal{E}}\left(pk, C, \psi_1, \ldots, \psi_t\right) = \left(C, \psi_1, \ldots, \psi_t\right)$$

thus, we require the output of $\text{Evaluate}_{\mathcal{E}}$ to be bounded

Of course, we want to avoid trivial "do-it-yourself" solutions such as

$$\text{Evaluate}_{\mathcal{E}}\left(pk, C, \psi_1, \ldots, \psi_t\right) = \left(C, \psi_1, \ldots, \psi_t\right)$$

thus, we require the output of $\text{Evaluate}_{\mathcal{E}}$ to be bounded or even to look like a "normal" ciphertext

Definition of $\mathcal{C}_{\mathcal{E}}$

Definition (Homomorphic encryption)

Scheme $\mathcal{E}$ is homomorphic for circuits in $\mathcal{C}_{\mathcal{E}}$ if $\mathcal{E}$ is correct for $\mathcal{C}_{\mathcal{E}}$ and $\text{Decrypt}_{\mathcal{E}}$ can be expressed as a circuit $D_{\mathcal{E}}$ of size $\text{poly}(\lambda)$ where λ is the security parameter.

Definition (Fully Homomorphic encryption)

Scheme $\mathcal{E}$ is fully homomorphic if it is homomorphic for all circuits.

Definition (Leveled Fully Homomorphic encryption)

A family of schemes $\mathcal{E}^{(d)}$ is leveled fully homomorphic if they use same decryption circuit and $\mathcal{E}^{(d)}$ is fully homomorphic for all circuits up to depth d.

3-step Construction

- "Bootstraping" encryption scheme

3-step Construction

- "Bootstraping" encryption scheme
- Ideal Lattices

3-step Construction

- "Bootstraping" encryption scheme
- Ideal Lattices
- Shortening decryption circuit

Bootstrapable Scheme is the one that can evaluate its own decryption circuit with a "little" add-on.
Evaluating the decryption circuit means finding different ciphertexts for the same plaintext without knowing the plaintext and hoping the new plaintext is "better".

If $\mathcal{E}$ is bootstrapable with plaintext space $\mathcal{P} = \{0, 1\}$ and

- $\psi_1 = \text{Encrypt}_{\mathcal{E}}(pk_1, \pi)$ is old noisy ciphertext
- $\overline{sk_{1j}} = \text{Encrypt}_{\mathcal{E}}(pk_2, sk_1)$
  - sk_1 is the secrect key corresponding to public key pk_1
  - $\overline{sk_{1j}}$ is sk_1 encrypted under public key pk_2

If $\mathcal{E}$ is bootstrapable with plaintext space $\mathcal{P} = \{0, 1\}$ and

- $\psi_1 = \mathsf{Encrypt}_{\mathcal{E}}(pk_1, \pi)$ is old noisy ciphertext
- $\overline{sk_{1j}} = \mathsf{Encrypt}_{\mathcal{E}}(pk_2, sk_1)$
  - sk_1 is the secrect key corresponding to public key pk_1
  - $\overline{sk_{1j}}$ is sk_1 encrypted under public key pk_2

then we can consider algorithm

$\mathsf{Recrypt}_{\mathcal{E}}(pk, D_{\mathcal{E}}, \langle \overline{sk_{1j}} \rangle, \psi_1)$

1. Set $\overline{\psi_{1j}} \xleftarrow{R} \mathsf{Encrypt}_{\mathcal{E}}(pk_2, \psi_{1j})$
2. Output $\psi_2 \leftarrow \mathsf{Evaluate}_{\mathcal{E}}(pk_2, D_{\mathcal{E}}, \langle \langle \overline{sk_{1j}} \rangle, \langle \overline{\psi_{1j}} \rangle \rangle)$

If $\mathcal{E}$ is bootstrapable with plaintext space $\mathcal{P} = \{0, 1\}$ and

- $\psi_1 = \text{Encrypt}_{\mathcal{E}}(pk_1, \pi)$ is old noisy ciphertext
- $\overline{sk_{1j}} = \text{Encrypt}_{\mathcal{E}}(pk_2, sk_1)$
  - sk_1 is the secrect key corresponding to public key pk_1
  - $\overline{sk_{1j}}$ is sk_1 encrypted under public key pk_2

then we can consider algorithm

$\text{Recrypt}_{\mathcal{E}}(pk, D_{\mathcal{E}}, \langle \overline{sk_{1j}} \rangle, \psi_1)$

1. Set $\overline{\psi_{1j}} \xleftarrow{R} \text{Encrypt}_{\mathcal{E}}(pk_2, \psi_{1j})$
2. Output $\psi_2 \leftarrow \text{Evaluate}_{\mathcal{E}}(pk_2, D_{\mathcal{E}}, \langle \langle \overline{sk_{1j}} \rangle, \langle \overline{\psi_{1j}} \rangle \rangle)$

We "refreshed" ciphertext old ψ_1 to (hopefully) fresh $\psi_2 = \text{Encrypt}_{\mathcal{E}}(sk_1, \pi)$

If $\mathcal{E}$ is bootstrapable, we can evaluate circuits of arbitrary length since we can

- do a little work - NAND gate
- when necessary, refresh ciphertext, i.e. lower error

Definition

Augmented Decryption Circuit Let Γ be a set of gates. For a gate $g \in \Gamma$, the g-augmented decryption circuit consist of g-gate connecting multiple copies of $D_{\mathcal{E}}$ on input. We denote the set of g-augmented decryption circuits with $g \in \Gamma$ by $D_{\mathcal{E}}(\Gamma)$.

Definition

Bootstrapable Encryption Let $\mathcal{C}_{\mathcal{E}}$ be a set of circuits with respect to which $\mathcal{E}$ is homomorphic. We say that $\mathcal{E}$ is bootstrapable with respect to Γ if

$$D_{\mathcal{E}}(\Gamma) \subseteq \mathcal{C}_{\mathcal{E}}$$

Circuicts of arbitrary length

- Suppose scheme $\mathcal{E}$ is bootstrapable, i.e. evaluates its own decryption circuit augmented by gates in Γ
- Then there exists $\mathcal{E}_\delta$ that evaluates arbitrary circuits with gates in Γ with depth at most δ
- Ciphertexts same size in $\mathcal{E}_\delta$ as in $\mathcal{E}$
- Public key:
  - $\delta + 1$ $\mathcal{E}$ public keys $pk_0, \ldots, pk_\delta$
  - δ encrypted secret keys $\text{Encrypt}(pk_i, sk_{i-1})$
  - length linear in δ
  - if $\mathcal{E}$ KDM[1]-secure, constant length public key is $pk, \text{Encrypt}(pk, sk)$

[1] Key Dependent Message, specifically circular-secure, we can encrypt private key with its public key

So what's the lesson learned from bootstrapability?

If we want to create a fully homomorphic encryption, we have to find one that

- can evaluate its (augmented) decryption circuit
- thus, it has "shallow" (short) decryption circuit

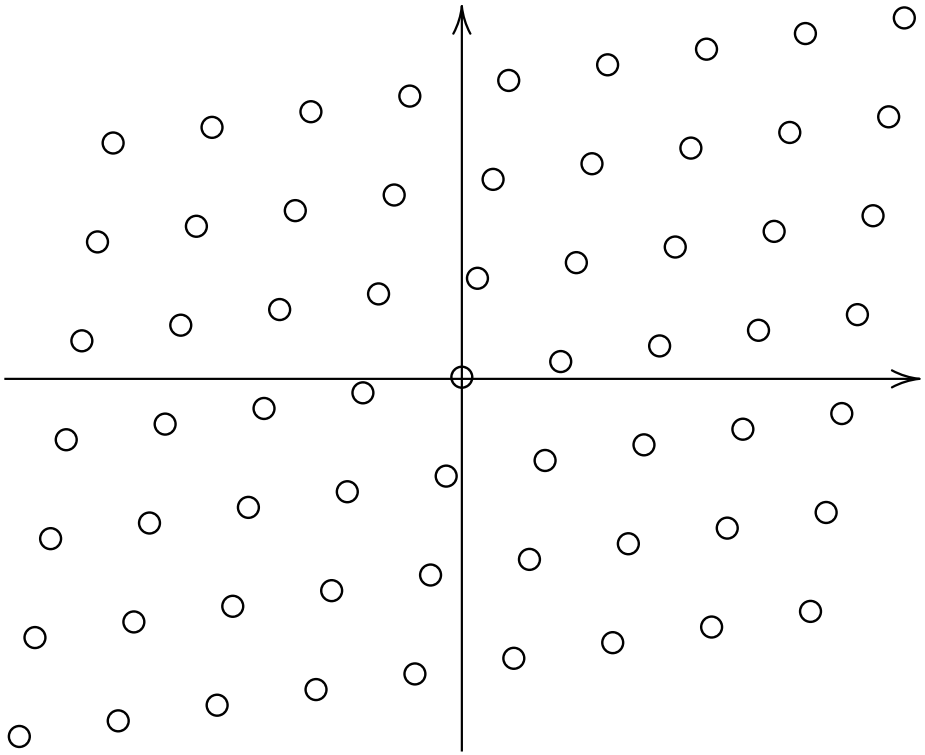
So what's the lesson learned from bootstrapability?

If we want to create a fully homomorphic encryption, we have to find one that

- can evaluate its (augmented) decryption circuit
- thus, it has "shallow" (short) decryption circuit
- thus RSA is out of the question (exponentiation is expensive and long, "unparallelizable")
- lattice based cryptosystems' decryption is often a simple inner product (shallow circuit)

What is a lattice?

2-dimensional lattice $\mathcal{L}$ is a set of $\{c_1\mathbf{u_1} + c_2\mathbf{u_2}\}$ where $c_1, c_2 \in \mathbb{Z}$ and $\mathbf{u_1}, \mathbf{u_1}$ are linearly independent



(Watch the blackboard for the example of a simple lattice encryption scheme :-))

What is an ideal lattice?

- Lattice: $\mathcal{L} = \left\{ \sum_{1 \leq i \leq n} \alpha_i \mathbf{u}_i, \alpha_i \in \mathbb{Z} \right\}$, $\mathbf{u}_i$ linearly independent
- Ideal I in ring R satisfies $I \subseteq R$ and $ri \in I$ for any $i \in I, r \in R$

What is an ideal lattice?

- Lattice: $\mathcal{L} = \left\{ \sum_{1 \leq i \leq n} \alpha_i \mathbf{u}_i, \alpha_i \in \mathbb{Z} \right\}$, $\mathbf{u}_i$ linearly independent
- Ideal I in ring R satisfies $I \subseteq R$ and $ri \in I$ for any $i \in I, r \in R$
- for $f(x)$ monic and $R = \mathbb{Z}[x]/f(x)$ ideal lattice is two objects at once
  - ideal $(a(x))$ in R, $\deg a = n$, (i.e. multiples of $a(x)$ modulo $f(x)$), multiplication structure
  - lattice in $\mathbb{Z}$ generated by coefficient vectors of $(x^i a(x) \mod f(x))$, for $1 \leq i \leq n-1$, addition structure

Initial construction

- $\mathrm{KeyGen}_{\mathcal{E}}(R, B_I)$
  - $(B_J^{sk}, B_J^{pk}) \leftarrow \mathrm{IdealGen}(R, B_I)$
  - public key pk includes R, B_I, B_J^{pk}, sampling algorithm Samp from $R \mod I$ to R
  - secrect key sk includes B_J^{sk}

Initial construction

- $\text{KeyGen}_{\mathcal{E}}(R, B_I)$
  - $(B_J^{sk}, B_J^{pk}) \leftarrow \text{IdealGen}(R, B_I)$
  - public key pk includes R, B_I, B_J^{pk}, sampling algorithm Samp from $R \mod I$ to R
  - secrect key sk includes B_J^{sk}
- $\text{Encrypt}_{\mathcal{E}}(pk, \pi)$
  - $\psi' \xleftarrow{R} \pi + I$ (using $\text{Samp}(\pi, B_I, R, B_J^{pk})$ algorithm)
  - $\psi \leftarrow \psi' \mod B_J^{pk}$

Initial construction

- $\text{KeyGen}_{\mathcal{E}}(R, B_I)$
  - $(B_J^{sk}, B_J^{pk}) \leftarrow \text{IdealGen}(R, B_I)$
  - public key pk includes R, B_I, B_J^{pk}, sampling algorithm Samp from $R \mod I$ to R
  - secrect key sk includes B_J^{sk}
- $\text{Encrypt}_{\mathcal{E}}(pk, \pi)$
  - $\psi' \xleftarrow{R} \pi + I$ (using $\text{Samp}(\pi, B_I, R, B_J^{pk})$ algorithm)
  - $\psi \leftarrow \psi' \mod B_J^{pk}$
- $\text{Decrypt}_{\mathcal{E}}(sk, \psi)$
  - $\pi \leftarrow (\psi \mod B_J^{sk}) \mod B_I$

Initial construction

- $\text{KeyGen}_{\mathcal{E}}(R, B_I)$
  - $(B_J^{sk}, B_J^{pk}) \leftarrow \text{IdealGen}(R, B_I)$
  - public key pk includes R, B_I, B_J^{pk}, sampling algorithm Samp from $R \mod I$ to R
  - secrect key sk includes B_J^{sk}
- $\text{Encrypt}_{\mathcal{E}}(pk, \pi)$
  - $\psi' \xleftarrow{R} \pi + I$ (using $\text{Samp}(\pi, B_I, R, B_J^{pk})$ algorithm)
  - $\psi \leftarrow \psi' \mod B_J^{pk}$
- $\text{Decrypt}_{\mathcal{E}}(sk, \psi)$
  - $\pi \leftarrow (\psi \mod B_J^{sk}) \mod B_I$
- $\text{Evaluate}_{\mathcal{E}}(pk, C, \Psi)$
  - circuit C on input is "$\mod B_I$" circuit on plaintexts
  - to create generalized circuit $g(C)$ replace Add_{B_I} and Mul_{B_I} by standard ring operations, i.e. drop "$\mod B_I$"
  - output the result of $g(C)$ with Ψ on input

Necessary notation

- X_{ENC} - image of Samp
- X_{DEC} - representatives of mod B_J^{sk} classes, i.e. R mod B_J^{sk}
- permitted circuit - any circuit satisfying $g(C)(X_{ENC}) \subseteq X_{DEC}$
- $C_{\mathcal{E}}$ - set of all permitted circuits
- valid ciphertext - output of Evaluate$_{\mathcal{E}}$ with any valid input

Theorem (Correctness of $\mathcal{E}$)

For any permitted circuit $\mathcal{E}$ is correct, i.e. Decrypt *correctly decrypts all valid ciphertexts.*

Theorem (Correctness of $\mathcal{E}$)

For any permitted circuit $\mathcal{E}$ is correct, i.e. Decrypt *correctly decrypts all valid ciphertexts.*

Proof.

$\Psi = \{\psi_1, \ldots, \psi_t\}$, *where* $\psi_k = \pi_k + i_k + j_k$ *and* $\pi_k + i_k \in X_{ENC}$

$$\text{Evaluate}_{\mathcal{E}}(pk, C, \Psi) = g(C)(\Psi) \mod B_J^{pk} \in g(C)(\pi_1 + i_1, \ldots, \pi_t + i_t) + J$$

Since C is permitted circuit $g(C)(\Psi) \in X_{DEC}$.

$$\text{Decrypt}(sk, \text{Evaluate}_{\mathcal{E}}(pk, C, \Psi)) = g(C)(\pi_1 + i_1, \ldots, \pi_t + i_t) \mod B_I$$
$$= g(C)(\pi_1, \ldots, \pi_t) \mod B_I = C(\pi_1, \ldots, \pi_t)$$ □

Addition and Multiplication

Geometrically, we see circuit is permitted if it "doesn't expand too much". Specifically

- addition is cheap: $\|u_1 + u_2\| \leq \|u_1\| + \|u_2\|$
- multiplication is worse. We can show for each ring $R = \mathbb{Z}[x]/f(x)$ there exists $\gamma_{\mathrm{MULT}}(R)$ such that for any two vectors u_1, u_2

$$\|u_1 \times_R u_2\| \leq \gamma_{\mathrm{MULT}}(R)\|u_1\|\|u_2\|$$

in lattice associated with R

Addition and Multiplication

Geometrically, we see circuit is permitted if it "doesn't expand too much". Specifically

- addition is cheap: $\|u_1 + u_2\| \leq \|u_1\| + \|u_2\|$
- multiplication is worse. We can show for each ring $R = \mathbb{Z}[x]/f(x)$ there exists $\gamma_{\text{MULT}}(R)$ such that for any two vectors u_1, u_2

$$\|u_1 \times_R u_2\| \leq \gamma_{\text{MULT}}(R)\|u_1\|\|u_2\|$$

  in lattice associated with R
- Moreover, if $f(x) = x^n - g(x)$ with $\deg g \leq n - \frac{n-1}{k}$, $k \geq 2$ it holds $\gamma_{\text{MULT}}(R) \leq \sqrt{2n}(1 + 2n + (\sqrt{(k-1)n}\|f\|)^k)$.
- Note: There's "some" evidence $f(x) = x^n - 1$ is not be best choice so we'll avoid it.

Scheme not bootstrapable, yet (cannot evaluate own decryption circuit). Some tweaks are needed

Scheme not bootstrapable, yet (cannot evaluate own decryption circuit). Some tweaks are needed

- Instead of X_{DEC} resp. $\mathcal{B}(r_{DEC})$ use $\mathcal{B}(r_{DEC}/2)$. Motivation: simplify $\psi \mod B_J^{sk} = \psi - B_J^{sk} \lfloor \left(B_J^{sk}\right)^{-1} \psi \rceil$. After tweak accented expression is at most $1/4$ off an integer.

Scheme not bootstrapable, yet (cannot evaluate own decryption circuit). Some tweaks are needed

- Instead of X_{DEC} resp. $\mathcal{B}(r_{DEC})$ use $\mathcal{B}(r_{DEC}/2)$. Motivation: simplify $\psi \mod B_J^{sk} = \psi - B_J^{sk} \lfloor \left(B_J^{sk}\right)^{-1} \psi \rceil$. After tweak accented expression is at most $1/4$ off an integer.
- Compute $\mathbf{v}_J^{sk} \in J^{-1} \subseteq \mathbb{Q}[x]/f(x)$ such that decryption simplifies to $\psi - \lfloor \mathbf{v}_J^{sk} \times \psi \rceil \mod B_I$ (after radius r_{DEC} adjustment)

Scheme not bootstrapable, yet (cannot evaluate own decryption circuit). Some tweaks are needed

- Instead of X_{DEC} resp. $\mathcal{B}(r_{DEC})$ use $\mathcal{B}(r_{DEC}/2)$. Motivation: simplify $\psi \mod B_J^{sk} = \psi - B_J^{sk} \lfloor \left(B_J^{sk}\right)^{-1} \psi \rceil$. After tweak accented expression is at most $1/4$ off an integer.
- Compute $\mathbf{v}_J^{sk} \in J^{-1} \subseteq \mathbb{Q}[x]/f(x)$ such that decryption simplifies to $\psi - \lfloor \mathbf{v}_J^{sk} \times \psi \rceil \mod B_I$ (after radius r_{DEC} adjustment)
- Decreasing the complexity of decryption circuit ... next time.

Finally, the scheme is bootstrapable, i.e. fully homomorphic.

Thank you for your attention!
Martin Hlaváč (based on Craig Gentry's PhD thesis)