Fully Homomorphic Encryption using Ideal Lattices

by Craig Gentry

Ahmer Raza
School of Mathematical and Statistical Sciences
Clemson University

April 10, 2025

Overview

1. Homomorphic Encryption
2. Somewhat Homomorphic Encryption
3. Bootstrapping
4. Fully Homomorphic Encryption

Homomorphic Encryption

Recall a regular PKE scheme: KeyGen(λ), Encrypt(pk, π), and Decrypt(sk, ψ).

Homomorphic Encryption

Recall a regular PKE scheme: $\mathsf{KeyGen}(\lambda)$, $\mathsf{Encrypt}(\mathsf{pk}, \pi)$, and $\mathsf{Decrypt}(\mathsf{sk}, \psi)$.

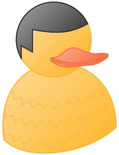
Goal: Allow anyone to evaluate a function of t plaintexts *without* decrypting them and *still* preserving encryption.

Homomorphic Encryption

Recall a regular PKE scheme: KeyGen(λ), Encrypt(pk, π), and Decrypt(sk, ψ).

Goal: Allow anyone to evaluate a function of t plaintexts *without* decrypting them and *still* preserving encryption.

Alice (Sender)
Plaintexts: $\boldsymbol{\pi}$
Keys: $(\mathsf{sk}, \mathsf{pk})$



Bob (Evaluator)
Arithmetic Circuit: C
Key: pk

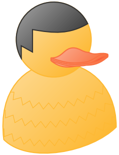
$$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_t)$$
$$\boldsymbol{\psi} = (\psi_1, \ldots, \psi_t)$$

Homomorphic Encryption

Recall a regular PKE scheme: $\mathsf{KeyGen}(\lambda)$, $\mathsf{Encrypt}(\mathsf{pk}, \pi)$, and $\mathsf{Decrypt}(\mathsf{sk}, \psi)$.

Goal: Allow anyone to evaluate a function of t plaintexts *without* decrypting them and *still* preserving encryption.

$$\boldsymbol{\psi} = \mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{\pi}) \longrightarrow$$



Alice (Sender)
Plaintexts: $\boldsymbol{\pi}$
Keys: $(\mathsf{sk}, \mathsf{pk})$

Bob (Evaluator)
Arithmetic Circuit: C
Key: pk

$$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_t)$$
$$\boldsymbol{\psi} = (\psi_1, \ldots, \psi_t)$$

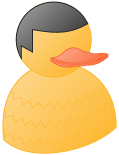
Homomorphic Encryption

Recall a regular PKE scheme: $\mathsf{KeyGen}(\lambda)$, $\mathsf{Encrypt}(\mathsf{pk}, \pi)$, and $\mathsf{Decrypt}(\mathsf{sk}, \psi)$.

Goal: Allow anyone to evaluate a function of t plaintexts *without* decrypting them and *still* preserving encryption.

$$\boldsymbol{\psi} = \mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{\pi}) \longrightarrow$$

$$\longleftarrow \psi = \mathsf{Evaluate}(\mathsf{pk}, C, \boldsymbol{\psi})$$



Alice (Sender)
Plaintexts: $\boldsymbol{\pi}$
Keys: $(\mathsf{sk}, \mathsf{pk})$

Bob (Evaluator)
Arithmetic Circuit: C
Key: pk

$$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_t)$$
$$\boldsymbol{\psi} = (\psi_1, \ldots, \psi_t)$$

Homomorphic Encryption

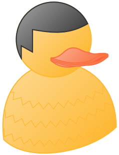
Recall a regular PKE scheme: $\mathsf{KeyGen}(\lambda)$, $\mathsf{Encrypt}(\mathsf{pk}, \pi)$, and $\mathsf{Decrypt}(\mathsf{sk}, \psi)$.

Goal: Allow anyone to evaluate a function of t plaintexts *without* decrypting them and *still* preserving encryption.

$$\boxed{C(\boldsymbol{\pi}) = \mathsf{Decrypt}(\mathsf{pk}, \psi)}$$

$$\boldsymbol{\psi} = \mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{\pi}) \longrightarrow$$

$$\longleftarrow \psi = \mathsf{Evaluate}(\mathsf{pk}, C, \boldsymbol{\psi})$$



Alice (Sender)
Plaintexts: $\boldsymbol{\pi}$
Keys: $(\mathsf{sk}, \mathsf{pk})$

Bob (Evaluator)
Arithmetic Circuit: C
Key: pk

$$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_t)$$
$$\boldsymbol{\psi} = (\psi_1, \ldots, \psi_t)$$

Homomorphic Encryption

Type	Add/Multiply	Operations	Examples
Partially HE (PHE)	One	Unlimited	RSA, ElGamal, Paillier
Somewhat HE (SHE)	Both	Limited	[Gentry, 2009], Boneh-Goh-Nissim, Melchor-Gaborit-Herranz
Fully HE (FHE)	Both	Unlimited	[Gentry, 2009]

[Gentry, 2009] begins with an unstable lattice-based SHE scheme, and uses **bootstrapping** to derive a FHE scheme.

Somewhat Homomorphic Encryption

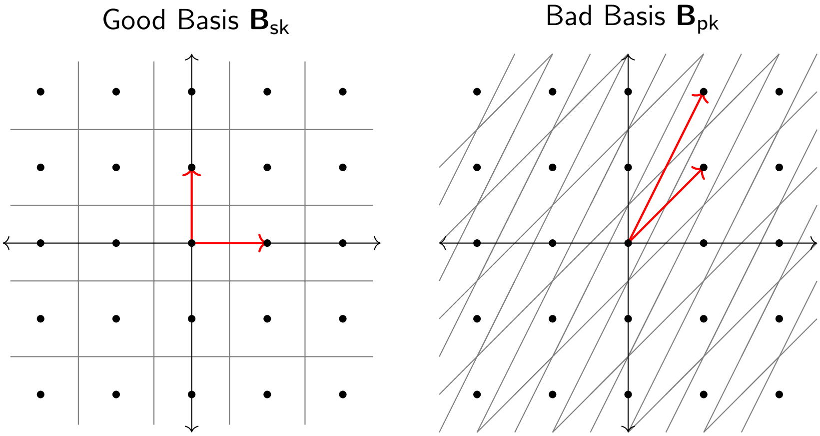
Given a **lattice** $\{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^n\}$, its **half-open parallelepiped** is $\{\mathbf{Bx} : \mathbf{x} \in [-0.5, 0.5)^n\}$.

$\mathbf{v} \bmod \mathbf{B} = \mathbf{v} - \mathbf{B}\lceil \mathbf{B}^{-1}\mathbf{v} \rfloor$ reduces any point $\mathbf{v}$ onto the half-open parallelepiped.

Somewhat Homomorphic Encryption

Given a **lattice** $\{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^n\}$, its **half-open parallelepiped** is $\{\mathbf{Bx} : \mathbf{x} \in [-0.5, 0.5)^n\}$.

$\mathbf{v} \bmod \mathbf{B} = \mathbf{v} - \mathbf{B}\lceil \mathbf{B}^{-1}\mathbf{v} \rfloor$ reduces any point $\mathbf{v}$ onto the half-open parallelepiped.

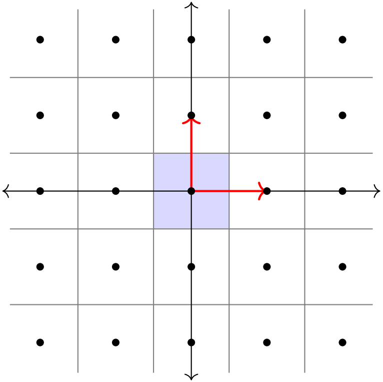


Somewhat Homomorphic Encryption

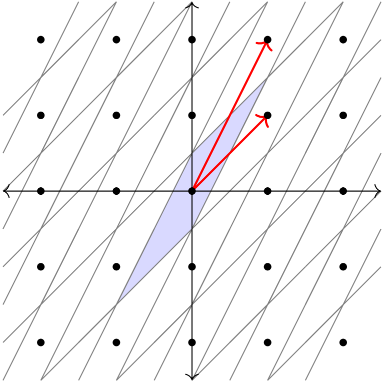
Given a **lattice** $\{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^n\}$, its **half-open parallelepiped** is $\{\mathbf{Bx} : \mathbf{x} \in [-0.5, 0.5)^n\}$.

$\mathbf{v} \bmod \mathbf{B} = \mathbf{v} - \mathbf{B}\lceil \mathbf{B}^{-1}\mathbf{v} \rfloor$ reduces any point $\mathbf{v}$ onto the half-open parallelepiped.

Good Basis $\mathbf{B}_{\mathsf{sk}}$



Bad Basis $\mathbf{B}_{\mathsf{pk}}$

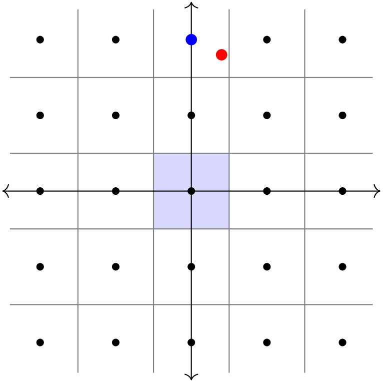


Somewhat Homomorphic Encryption

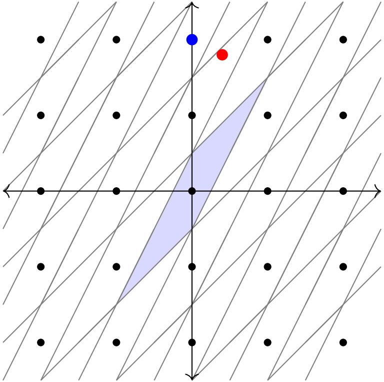
Given a **lattice** $\{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^n\}$, its **half-open parallelepiped** is $\{\mathbf{Bx} : \mathbf{x} \in [-0.5, 0.5)^n\}$.

$\mathbf{v} \bmod \mathbf{B} = \mathbf{v} - \mathbf{B}\lceil \mathbf{B}^{-1}\mathbf{v} \rfloor$ reduces any point $\mathbf{v}$ onto the half-open parallelepiped.

Good Basis $\mathbf{B}_{\mathsf{sk}}$



Bad Basis $\mathbf{B}_{\mathsf{pk}}$

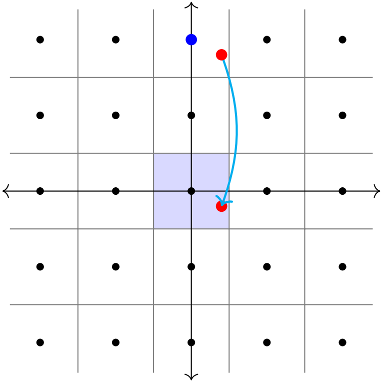


Somewhat Homomorphic Encryption

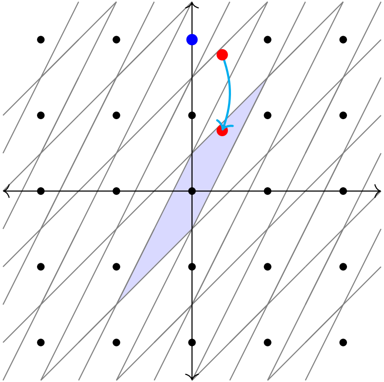
Given a **lattice** $\{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^n\}$, its **half-open parallelepiped** is $\{\mathbf{Bx} : \mathbf{x} \in [-0.5, 0.5)^n\}$.

$\mathbf{v} \bmod \mathbf{B} = \mathbf{v} - \mathbf{B}\lceil \mathbf{B}^{-1}\mathbf{v} \rfloor$ reduces any point $\mathbf{v}$ onto the half-open parallelepiped.

Good Basis $\mathbf{B}_{\mathsf{sk}}$



Bad Basis $\mathbf{B}_{\mathsf{pk}}$



Somewhat Homomorphic Encryption

Let $R = \mathbb{Z}[x]/f(x)$ be a **quotient ring**, where $f(x)$ is a monic polynomial of degree n.

Note that $R \cong \mathbb{Z}^n$, so R and any ideal $I \subset R$ can be viewed as a lattice.

$$a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} \in R \quad \longleftrightarrow \quad (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{Z}^n$$

Somewhat Homomorphic Encryption

Let $R = \mathbb{Z}[x]/f(x)$ be a **quotient ring**, where $f(x)$ is a monic polynomial of degree n.

Note that $R \cong \mathbb{Z}^n$, so R and any ideal $I \subset R$ can be viewed as a lattice.

$$a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} \in R \quad \longleftrightarrow \quad (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{Z}^n$$

An **ideal lattice** is any subset $I \subset R$ that is closed under addition, and closed under multiplication with R.

Example: $I = 2\mathbb{Z} = \{2k : k \in \mathbb{Z}\}$ is an ideal lattice of the ring $\mathbb{Z}$.

Somewhat Homomorphic Encryption

Parameters: Ring $R = \mathbb{Z}[x]/f(x)$ and basis $\mathbf{B}_I$ of a "small" ideal lattice I. The plaintext space is R/I (cosets of I in R), meaning π is a coset ($\pi = r + I$ for some $r \in R$).

Somewhat Homomorphic Encryption

Parameters: Ring $R = \mathbb{Z}[x]/f(x)$ and basis $\mathbf{B}_I$ of a “small” ideal lattice I. The plaintext space is R/I (cosets of I in R), meaning π is a coset ($\pi = r + I$ for some $r \in R$).

$(\mathbf{B}_{\mathsf{sk}}, \mathbf{B}_{\mathsf{pk}}) \leftarrow \mathsf{KeyGen}$

- $\mathbf{B}_{\mathsf{sk}}$ is a good basis and $\mathbf{B}_{\mathsf{pk}}$ is a bad basis of a “big” ideal lattice J
- J is relatively prime to I, meaning $I + J = R$

Somewhat Homomorphic Encryption

Parameters: Ring $R = \mathbb{Z}[x]/f(x)$ and basis $\mathbf{B}_I$ of a "small" ideal lattice I. The plaintext space is R/I (cosets of I in R), meaning π is a coset ($\pi = r + I$ for some $r \in R$).

$(\mathbf{B}_{\mathsf{sk}}, \mathbf{B}_{\mathsf{pk}}) \leftarrow \mathsf{KeyGen}$

- $\mathbf{B}_{\mathsf{sk}}$ is a good basis and $\mathbf{B}_{\mathsf{pk}}$ is a bad basis of a "big" ideal lattice J
- J is relatively prime to I, meaning $I + J = R$

$\psi \leftarrow \mathsf{Encrypt}(\mathbf{B}_{\mathsf{pk}}, \pi)$

- Choose random, small representative $\pi' \in \pi + I$ with $\|\pi'\| \leq r_{\mathsf{Enc}}$
- Compute $\psi = \pi' \bmod \mathbf{B}_{\mathsf{pk}}$, mapping to the half-open parallelepiped

$\pi \leftarrow \mathsf{Decrypt}(\mathbf{B}_{\mathsf{sk}}, \psi)$

- Compute $\pi' = \psi \bmod \mathbf{B}_{\mathsf{sk}}$, which recovers the small representative if $\|\pi'\| \leq r_{\mathsf{Dec}}$
- Compute $\pi = \pi' \bmod \mathbf{B}_I$, which obtains the proper coset in R/I

Somewhat Homomorphic Encryption

Parameters: Ring $R = \mathbb{Z}[x]/f(x)$ and basis $\mathbf{B}_I$ of a "small" ideal lattice I. The plaintext space is R/I (cosets of I in R), meaning π is a coset ($\pi = r + I$ for some $r \in R$).

$(\mathbf{B}_{\mathsf{sk}}, \mathbf{B}_{\mathsf{pk}}) \leftarrow \mathsf{KeyGen}$

- $\mathbf{B}_{\mathsf{sk}}$ is a good basis and $\mathbf{B}_{\mathsf{pk}}$ is a bad basis of a "big" ideal lattice J
- J is relatively prime to I, meaning $I + J = R$

$\psi \leftarrow \mathsf{Encrypt}(\mathbf{B}_{\mathsf{pk}}, \pi)$

- Choose random, small representative $\pi' \in \pi + I$ with $\|\pi'\| \leq r_{\mathsf{Enc}}$
- Compute $\psi = \pi' \bmod \mathbf{B}_{\mathsf{pk}}$, mapping to the half-open parallelepiped

$\pi \leftarrow \mathsf{Decrypt}(\mathbf{B}_{\mathsf{sk}}, \psi)$

- Compute $\pi' = \psi \bmod \mathbf{B}_{\mathsf{sk}}$, which recovers the small representative if $\|\pi'\| \leq r_{\mathsf{Dec}}$
- Compute $\pi = \pi' \bmod \mathbf{B}_I$, which obtains the proper coset in R/I

$\psi_+ \leftarrow \mathsf{Add}(\mathbf{B}_{\mathsf{pk}}, \psi_1, \psi_2)$ outputs $(\psi_1 + \psi_2) \bmod \mathbf{B}_{\mathsf{pk}}$
$\psi_\times \leftarrow \mathsf{Multiply}(\mathbf{B}_{\mathsf{pk}}, \psi_1, \psi_2)$ outputs $(\psi_1 \cdot \psi_2) \bmod \mathbf{B}_{\mathsf{pk}}$

Somewhat Homomorphic Encryption

Encryption-decryption:

$$\begin{aligned}
\mathsf{Decrypt}(\mathbf{B}_{\mathsf{sk}}, \mathsf{Encrypt}(\mathbf{B}_{\mathsf{pk}}, \pi)) &= \mathsf{Decrypt}(\mathbf{B}_{\mathsf{sk}}, \pi' \bmod \mathbf{B}_{\mathsf{pk}}) \\
&= ((\pi' \bmod \mathbf{B}_{\mathsf{pk}}) \bmod \mathbf{B}_{\mathsf{sk}}) \bmod \mathbf{B}_I \\
&= \pi' \bmod \mathbf{B}_I && (\text{if } \|\pi'\| \leq r_{\mathsf{Enc}}, r_{\mathsf{Dec}}) \\
&= \pi + I
\end{aligned}$$

Somewhat Homomorphic Encryption

Encryption-decryption:

$$\begin{aligned}\mathrm{Decrypt}(\mathbf{B}_{\mathrm{sk}}, \mathrm{Encrypt}(\mathbf{B}_{\mathrm{pk}}, \pi)) &= \mathrm{Decrypt}(\mathbf{B}_{\mathrm{sk}}, \pi' \bmod \mathbf{B}_{\mathrm{pk}}) \\ &= ((\pi' \bmod \mathbf{B}_{\mathrm{pk}}) \bmod \mathbf{B}_{\mathrm{sk}}) \bmod \mathbf{B}_I \\ &= \pi' \bmod \mathbf{B}_I && (\text{if } \|\pi'\| \leq r_{\mathrm{Enc}}, r_{\mathrm{Dec}}) \\ &= \pi + I\end{aligned}$$

$\mathrm{Add}(\mathbf{B}_{\mathrm{pk}}, \psi_1, \psi_2)$ can be decrypted if $\pi'_1 + \pi'_2$ is in the $\mathbf{B}_{\mathrm{sk}}$ parallelepiped.

$$\|\pi'_1 + \pi'_2\| \leq \|\pi'_1\| + \|\pi'_1\| \leq 2 \cdot r_{\mathrm{Dec}}$$

$\mathrm{Multiply}(\mathbf{B}_{\mathrm{pk}}, \psi_1, \psi_2)$ can be decrypted if $\pi'_1 \cdot \pi'_2$ is in the $\mathbf{B}_{\mathrm{sk}}$ parallelepiped.

$$\|\pi'_1 \cdot \pi'_2\| \leq \gamma_{\mathrm{Multiply}}(R) \cdot \|\pi'_1\| \cdot \|\pi'_1\|$$

Bootstrapping

Key idea: What if the scheme could homomorphically evaluate its own decryption circuit?

Bootstrapping

Key idea: What if the scheme could homomorphically evaluate its own decryption circuit?

$$\begin{tikzcd}
 & \psi_{\mathsf{sk}} = \mathsf{Encrypt}(\mathsf{pk}, \mathsf{sk}) \arrow[d] & \\
\psi \arrow[r] & \boxed{\mathsf{Refresh}(\psi_{\mathsf{sk}}, \psi)} \arrow[r] & \psi' \\
\text{Saturated CT} & & \text{Error-Free CT}
\end{tikzcd}$$

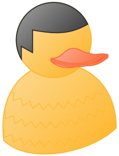
Fully Homomorphic Encryption

3 modifications to SHE scheme:

1. Simplify $\text{Decrypt}(\mathbf{B}_{\text{sk}}, \psi) = (\psi \bmod \mathbf{B}_{\text{sk}}) \bmod \mathbf{B}_{\text{I}}$ to $\text{Decrypt}(\mathbf{v}_{\text{sk}}, \psi) = (\psi - \mathbf{v}_{\text{sk}} \times \psi) \bmod 2$
2. Reduce decryption radius $r_{\text{Dec}} \to r_{\text{Dec}}/2$. This means ψ is required to be within $r_{\text{Dec}}/2$ of a lattice point.
3. Use a hint to "squash" the decryption: the encrypter helps with computing the decryption even before messages are received.

Fully Homomorphic Encryption

Alice (Sender)
Plaintexts: $\boldsymbol{\pi}$
Keys: $(\mathsf{sk}, \mathsf{pk})$



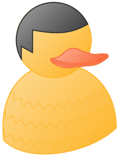
Bob (Evaluator)
Arithmetic Circuit: C
Key: pk

$$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_t)$$
$$\boldsymbol{\psi} = (\psi_1, \ldots, \psi_t)$$

Fully Homomorphic Encryption

$$\color{red}{\psi_{\mathsf{sk}} = \mathsf{Encrypt}(\mathsf{pk}, \mathsf{sk})}$$
$$\boldsymbol{\psi} = \mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{\pi})$$
$$\longrightarrow$$

Alice (Sender)
Plaintexts: $\boldsymbol{\pi}$
Keys: $(\mathsf{sk}, \mathsf{pk})$



Bob (Evaluator)
Arithmetic Circuit: C
Key: pk

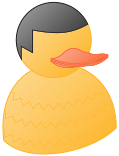
$$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_t)$$
$$\boldsymbol{\psi} = (\psi_1, \ldots, \psi_t)$$

Fully Homomorphic Encryption

$$\psi = \text{Evaluate}(\text{pk}, C_{i\dots i+k}, \boldsymbol{\psi})$$
$$\psi = \text{Refresh}(\psi_{\text{sk}}, \psi)$$
$$\dots$$

$$\psi_{\text{sk}} = \text{Encrypt}(\text{pk}, \text{sk})$$
$$\boldsymbol{\psi} = \text{Encrypt}(\text{pk}, \boldsymbol{\pi})$$

Alice (Sender)
Plaintexts: $\boldsymbol{\pi}$
Keys: (sk, pk)



Bob (Evaluator)
Arithmetic Circuit: C
Key: pk

$$\boldsymbol{\pi} = (\pi_1, \dots, \pi_t)$$
$$\boldsymbol{\psi} = (\psi_1, \dots, \psi_t)$$

Fully Homomorphic Encryption

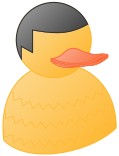
$$\boxed{\begin{array}{c}\psi = \text{Evaluate}(\text{pk}, C_{i\ldots i+k}, \boldsymbol{\psi}) \\ \psi = \text{Refresh}(\psi_{\text{sk}}, \psi) \\ \ldots\end{array}}$$

$$\psi_{\text{sk}} = \text{Encrypt}(\text{pk}, \text{sk})$$

$$\boldsymbol{\psi} = \text{Encrypt}(\text{pk}, \boldsymbol{\pi})$$

$\longrightarrow$

$\longleftarrow$



$$\psi = \text{Evaluate}(\text{pk}, C, \boldsymbol{\psi})$$

Alice (Sender)
Plaintexts: $\boldsymbol{\pi}$
Keys: (sk, pk)

Bob (Evaluator)
Arithmetic Circuit: C
Key: pk

$$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_t)$$

$$\boldsymbol{\psi} = (\psi_1, \ldots, \psi_t)$$

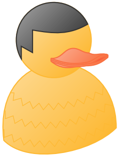
Fully Homomorphic Encryption

$\psi = \text{Evaluate}(\text{pk}, C_{i \ldots i+k}, \boldsymbol{\psi})$
$\psi = \text{Refresh}(\psi_{\text{sk}}, \psi)$
$\ldots$

$C(\boldsymbol{\pi}) = \text{Decrypt}(\text{pk}, \psi)$

$\psi_{\text{sk}} = \text{Encrypt}(\text{pk}, \text{sk})$
$\boldsymbol{\psi} = \text{Encrypt}(\text{pk}, \boldsymbol{\pi})$ $\longrightarrow$

$\longleftarrow$ $\psi = \text{Evaluate}(\text{pk}, C, \boldsymbol{\psi})$



Alice (Sender)
Plaintexts: $\boldsymbol{\pi}$
Keys: (sk, pk)

Bob (Evaluator)
Arithmetic Circuit: C
Key: pk

$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_t)$
$\boldsymbol{\psi} = (\psi_1, \ldots, \psi_t)$

References

Gentry, C. (2009).
Fully homomorphic encryption using ideal lattices.
In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, STOC '09, pages 169–178, New York, NY, USA. Association for Computing Machinery.