# Fully homomorphic encryption

## PCMI 2022 Undergraduate Summer School

## Lecture 9

Christelle Vincent

University of Vermont

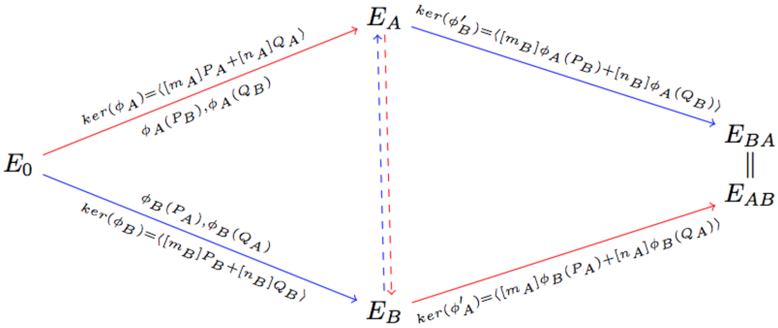
August 1, 2022

# Breaking news!!

SIDH/SIKE is broken as it stands!!

(Castryck-Decru, preliminary report posted on Saturday)

# SIDH in one slide



# Fully homomorphic encryption

## PCMI 2022 Undergraduate Summer School

## Lecture 9

Christelle Vincent

University of Vermont

August 1, 2022

# A dream from 1978

A public key cipher such that

- for any function $f$, and
- access only to encryptions $\mathtt{Enc}(m_1), \mathtt{Enc}(m_2), \ldots, \mathtt{Enc}(m_t)$

we can compute an encryption of $f(m_1, m_2, \ldots, m_t)$.

# Applications of FHE

- Query/search on encrypted database
- Private query/search on database
- Analysis of/machine learning on private data

# The punchline

Gentry came up with a construction in 2009 based on RLWE.

# Homomorphic encryption

A **homomorphic** cipher allows one operation on ciphertexts.

Usually this is $+$ or $\times$ on the integers (modulo $N$).

# Homomorphic encryption example: RSA

RSA encryption: $c \equiv m^e \pmod{N}$

If $c_i \equiv m_i^e \pmod{N}$ and $m \equiv m_1 m_2 \pmod{N}$, then

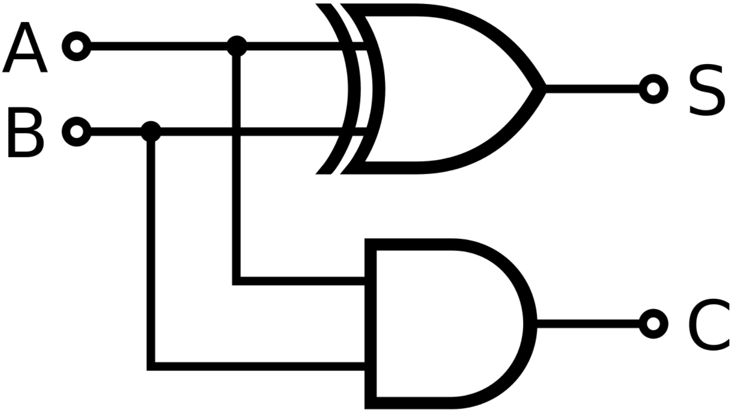
$$c_1 c_2 \equiv m_1^e m_2^e \equiv m^e \equiv c \pmod{N}.$$

A **fully homomorphic** cipher allows arbitrary operations on ciphertexts.

# Some circuit facts

Computer operations are encoded as **circuits** consisting of **gates**.

# Example: bit addition with carry



# Truth tables

Gates/programs can be expressed as **truth tables**.

| Input | Output |
|---|---|
| 00 | 0 |
| 01 | 1 |
| 10 | 1 |
| 11 | 0 |

XOR gate (sum)

| Input | Output |
|---|---|
| 00 | 0 |
| 01 | 0 |
| 10 | 0 |
| 11 | 1 |

AND gate (carry)

# Universal gates

A set of gates is **functionally complete** if any truth table can be expressed with these gates.

# One functionally complete set

The gates { AND, NOT } are enough to express anything.

| Input | Output |
|---|---|
| 00 | 0 |
| 01 | 0 |
| 10 | 0 |
| 11 | 1 |

AND gate

| Input | Output |
|---|---|
| 0 | 1 |
| 1 | 0 |

NOT

# Fully homomorphic encryption, again

It used to mean "respects $+$ and $\times$."

Now can also respect just one universal gate, like NAND or NOR.

# Main issue

Known constructions add **noise** to the ciphertext for security.

Operations increase the noise.

# Somewhat homomorphic encryption

A cipher that respects a certain number of $+$ and $\times$ is called **somewhat homomorphic**.

# One answer

Restrict how many operations can be done: **leveled fully homomorphic encryption**.

# Gentry's idea: bootstrapping

If the decryption circuit has $N$ operations, build a cipher that can handle at least $N+1$ operations.

# Gentry's analogy: Alice's jewelry store

Alice does not trust her employees, so gets lockboxes with gloves:

# Gentry's analogy: Alice's jewelry store

Unfortunately, the gloves get stiff with use.

Thankfully, the boxes have a **one-way insertion slot**, and are stretchy enough so one box can be **put inside another**.

# Gentry's solution: Alice's jewelry store

Several boxes, and the $i$th box contains the key of the $(i-1)$st box.

Work in box $i-1$ until almost stiff, put inside box $i$, unlock, work in box $i$ until almost stiff, and so on.

# Gentry's solution: FHE

Generate enough pairs $(\texttt{sk}_i, \texttt{pk}_i)$, and use $\texttt{pk}_i$ to encrypt $\texttt{sk}_{i-1}$.

When noise gets too big, "recrypt" ciphertext with next set of keys.

# Recryption example

Let $D$ be the decryption circuit: If $c$ is an encryption of $m$ under $\mathtt{pk}$ then

$$D(\mathtt{sk}, c) = m.$$

# Recryption example

Let

- $c_1$ encrypt $m$ under $\mathtt{pk}_1$,
- $\overline{\mathtt{sk}_1}$ encrypt $\mathtt{sk}_1$ under $\mathtt{pk}_2$, and
- $\overline{c_1}$ be an encryption of $c_1$ under $\mathtt{pk}_2$.

Then $D(\overline{\mathtt{sk}_1}, \overline{c_1})$ is $m$ encrypted under $\mathtt{pk}_2$

# Consequences for algorithms

- Must specify size of output of the circuit
- No random access memory
- Develop low depth algorithms

Thank you!