De Cifris Trends
in
Cryptographic Protocols

University of Trento and *De Componendis Cifris*

October 2023

Lecture 6
Fully-homomorphic Encryption

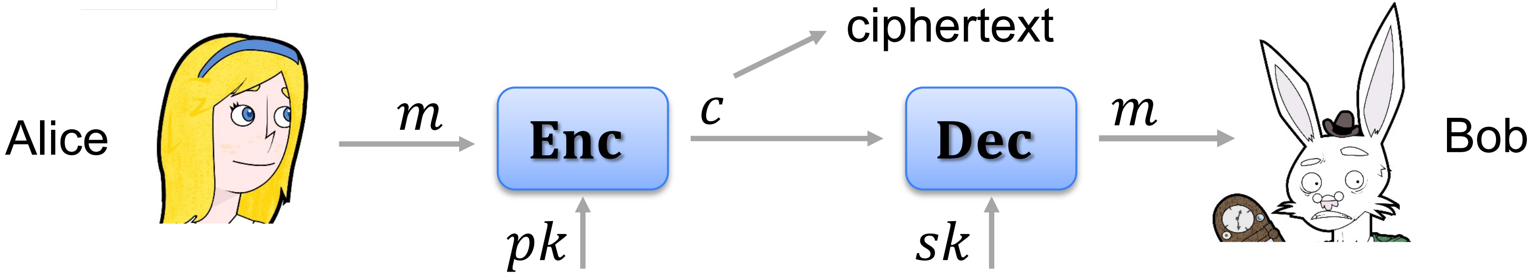
Daniele Venturi **Sapienza University of Rome**

Lecture 6
Fully-homomorphic Encryption

Daniele Venturi

Sapienza University of Rome

Public-key encryption



- **Proposed** by Diffie and Hellman in their seminal paper
  - W. Diffie, M. E. Hellman. New directions in cryptography. IEEE Trans. Inf. Theory 22(6), 1976
- First **realization** based on the hardness of **factoring**
  - R. L. Rivest, A. Shamir, L. M. Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Commun. ACM 21(2), 1978

Chosen-plaintext attacks security

Eve

pk ←

x_0, x_1 →

c ←

guess b →

Challenger

Generate pk, sk

Pick random b

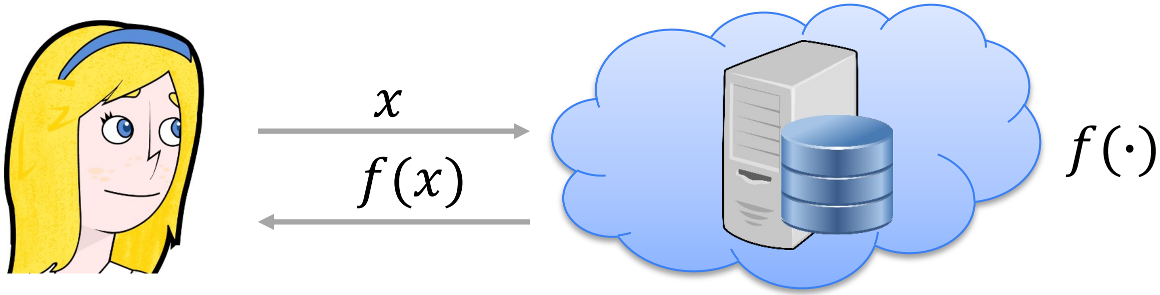
$c \leftarrow \mathbf{Enc}(pk, x_b)$

- The attacker cannot even guess a **single bit** of the plaintext
  - The messages are chosen by the adversary
  - CPA security implies hardness of **recovering the message**
  - CPA security implies hardness of **recovering the secret key**

Computing over encrypted data

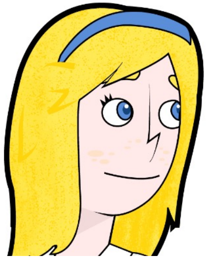
- Can we have a (public-key) encryption scheme which allows to run **computations** over **encrypted data**?
- Question dating back to the late 70s
  - Ron Rivest and "privacy homomorphisms"
- Partial solutions known
  - E.g., RSA and Elgamal enjoy **limited** forms of homomorphism
- First solution by Craig Gentry after **30 years**
  - C. Gentry. Fully-Homomorphic Encryption Using Ideal Lattices. STOC 2009
- The "Swiss Army knife of cryptography"

Outsourcing of computation



- Email, web search, navigation, social networking, …
- What about **private** x?

Fully-homomorphic encryption



pk, sk

$c = \mathbf{Enc}(pk, x)$

$y = \mathbf{Eval}(pk, f, c)$

pk $\quad f(\cdot)$

Correctness:

$\mathbf{Dec}(sk, y) = f(x)$

Privacy:

$\mathbf{Enc}(pk, x) \approx \mathbf{Enc}\left(pk, 0^{|x|}\right)$

FHE = Correctness $\forall$ efficient f = Correctness for universal set

Levelled FHE: **Bounded** depth f

- NAND
- $(+,\times)$ over a ring

Trivial FHE

- Let $(\mathbf{KGen}, \mathbf{Enc}, \mathbf{Dec})$ be **any PKE** scheme
- Define the following **"fully-homomorphic"** PKE $(\mathbf{KGen}, \mathbf{Enc}, \mathbf{Eval}, \mathbf{Dec}')$:
  - $\mathbf{Eval}(pk, f, c) = (f, c)$
  - $\mathbf{Dec}'(sk, c) = f(\mathbf{Dec}(sk, c))$

Wish: Complexity of decryption **much less** than running the circuit from scratch

Compactness

- We say that Π is **compact** if there is a **fixed polynomial bound** $B(\cdot)$ such that the size of an **evaluated** ciphertext is $\leq B(\lambda)$ (with high probability)
  - The latter should hold for **all circuits** f (with bounded depth $\tau \in \mathbb{N}$), and for **all inputs** to the circuit
  - The probability is over the randomness for key generation and ciphertext computation
- Note that B **does not depend** on τ (but only depends on the security parameter $\lambda \in \mathbb{N}$)
  - An even weaker condition (dubbed **weak compactness**) is to have $B = B(\lambda, \tau)$, but still say $B(\lambda, \tau) = \mathrm{poly}(\lambda) \cdot o(\log |f|)$

The Gentry-Sahai-Waters scheme

- In what follows we will present the FHE scheme due to:
  - C. Gentry, A. Sahai, B. Waters. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. CRYPTO 2013
- Based on the **Learning with Errors (LWE)** assumption
- Only achieves **levelled homomorphism**
  - But can be **bootstrapped** to **full homomorphism** using a trick by Gentry (under additional assumptions)
- The plaintext space will be $\mathbb{Z}_q = [-q/2, q/2)$, for a large prime q
  - For simplicity we sometimes write $[a]_q$ for $a \bmod q$

Learning with errors

- Introduced by Oded Regev in his seminal paper
  - O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. STOC 2005

$\vec{s}$ $\in \mathbb{Z}_q^n$

A

$+ \; \vec{\eta}$

$= \; \vec{b}$ $\in \mathbb{Z}_q^m$

$\approx$

A

$\vec{b}$

$\equiv \mathbf{U}_q^{(n+1)\times m}$

Small noise $\in \mathbb{Z}_q^m$

$|\eta_i| \le \alpha q; \alpha \ll 1$

Uniform distribution over $\mathbb{Z}_q^{(n+1)\times m}$

Hardness of LWE

- **Brute force** requires $q = 2^{O(n)}$ and $2^{O(n)}$ time
  - By Gaussian elimination we can find a set S of $O(n)$ equations such that $\sum_S \vec{a}_i = (1,0,..,0)$
  - This yields a guess for s_1 which is correct w.p. $1/2 + 2^{-\Theta(n)}$
  - An algorithm by Blum, Kalai, Wasserman reduces this to $2^{O(n/\log n)}$
- **Worst-case** to average-case hardness
  - Being able to solve LWE over a **random choice** of the secret, allows to solve it for **all possible choices** of the secret
- When $\alpha q > 2\sqrt{n}$, solving LWE is equivalent to approximating **short vectors** in a lattice to within $\tilde{O}(n/\alpha)$
  - It is also equivalent to **decoding a random linear code**

Rearranging notation

$$\boxed{\vec{b} = \vec{s} \times A + \vec{\eta}}$$

$\vec{s}$

$\in \mathbb{Z}_q^n$

A

$+\ \vec{\eta}$

$=\ \vec{b}$

Small noise $\in \mathbb{Z}_q^m$

$|\eta_i| \leq \alpha q; \alpha \ll 1$

$\in \mathbb{Z}_q^m$

New secret $\vec{s} \in \mathbb{Z}_q^{n+1}$

$\vec{s} \quad -1$

A

New matrix

$A' \in \mathbb{Z}_q^{(n+1)\times m}$

$\vec{b}$

$\Leftrightarrow \quad = \ \vec{\eta}$

$$\boxed{\text{LWE: } A' = (A||\vec{b}) \approx_c \mathbf{U}_q^{(n+1)\times m}}$$

Eigenvectors method

- Let C_1 and C_2 be matrices for **eigenvector** $\vec{s}$, and **eigenvalues** x_1, x_2 (i.e., $\vec{s} \times C_i = x_i \cdot \vec{s}$)
  - $C_1 + C_2$ has eigenvalue $x_1 + x_2$ w.r.t. $\vec{s}$
  - $C_1 \times C_2$ has eigenvalue $x_1 \cdot x_2$ w.r.t. $\vec{s}$
- Idea: Let C be the ciphertext, $\vec{s}$ be the secret key and x be the plaintext (say over $\mathbb{Z}_q$)
  - Homomorphism for **addition/multiplication**
  - But **insecure**: Easy to compute eigenvalues

Approximate eigenvectors

- Approximate variant: $\vec{s} \times C = x \cdot \vec{s} + \vec{e} \approx x \cdot \vec{s}$
  - Decryption **works** as long as $\|\vec{e}\|_\infty \ll q$

$$\vec{s} \times C_1 = x_1 \cdot \vec{s} + \vec{e}_1 \qquad \vec{s} \times C_2 = x_2 \cdot \vec{s} + \vec{e}_2$$

$$\|\vec{e}_1\|_\infty \ll q \qquad \|\vec{e}_2\|_\infty \ll q$$

- Goal: Define **homomorphic** operations

$$\begin{aligned} C_{\text{add}} &= C_1 + C_2: \\ \vec{s} \times (C_1 + C_2) &= \vec{s} \times C_1 + \vec{s} \times C_2 \\ &= x_1 \cdot \vec{s} + \vec{e}_1 + x_2 \cdot \vec{s} + \vec{e}_2 \\ &= (x_1 + x_2) \cdot \vec{s} + (\vec{e}_1 + \vec{e}_2) \end{aligned}$$

Noise **grows** a little!

Approximate eigenvectors

- Approximate variant: $\vec{s} \times C = x \cdot \vec{s} + \vec{e} \approx x \cdot \vec{s}$
  - Decryption **works** as long as $\|\vec{e}\|_\infty \ll q$

$$\vec{s} \times C_1 = x_1 \cdot \vec{s} + \vec{e}_1 \qquad \vec{s} \times C_2 = x_2 \cdot \vec{s} + \vec{e}_2$$
$$\|\vec{e}_1\|_\infty \ll q \qquad \|\vec{e}_2\|_\infty \ll q$$

- Goal: Define **homomorphic** operations

$$C_{\text{mult}} = C_1 \times C_2:$$
$$\begin{aligned}\vec{s} \times (C_1 \times C_2) &= (x_1 \cdot \vec{s} + \vec{e}_1) \times C_2 \\ &= x_1 \cdot (x_2 \cdot \vec{s} + \vec{e}_2) + \vec{e}_1 \times C_2 \\ &= x_1 \cdot x_2 \cdot \vec{s} + (x_1 \cdot \vec{e}_2 + \vec{e}_1 \times C_2)\end{aligned}$$

Noise **grows**!
Needs to be **small**!

Shrinking gadgets

- Write entries in C using **binary decomposition**; e.g.

$$C = \begin{bmatrix} 3 & 5 \\ 1 & 4 \end{bmatrix} \pmod 8 \xrightarrow{\text{yields}} \text{bits}(C) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \pmod 8$$

- **Reverse** operation:

$$C = G \times G^{-1}(C) = \begin{bmatrix} 2^{N-1} & \ldots & 2 & 1 & 0 & \ldots & 0 & 0 \\ 0 & \ldots & 0 & 0 & 2^{N-1} & \ldots & 2 & 1 \end{bmatrix} \times \text{bits}(C)$$

$$k \cdot N = k\lceil \log q \rceil$$

$$\boxed{\Rightarrow \vec{s} \times C = \vec{s} \times G \times G^{-1}(C)}$$

The GSW scheme: description

$\vec{s}$

A

public key

$\in \mathbb{Z}_q^{n\times m} =$ $\vec{\eta}$

$\in \mathbb{Z}_2^{m\times N} = \mathbb{Z}_2^{m\times n\cdot\lceil\log q\rceil}$

$\boxed{\mathbf{Enc}(A, x; R) = [A\times R + x\cdot G]_q}$

$\|\vec{e}\|_\infty = \|\vec{\eta}\times R\|_\infty \leq (\alpha q)\cdot m = n\cdot m$

Invariant: $\vec{s}\times C = \vec{e} + x\cdot\vec{s}\times G$

$\mathbf{Dec}(\vec{s}, C) = \vec{s}\times C\times G^{-1}((0,\dots,0,-\lfloor q/2\rfloor))$
$= \vec{e}\times G^{-1}(\cdots) + x\cdot\vec{s}\times G\times G^{-1}((0,\dots,0,-\lfloor q/2\rfloor))$
$= \vec{e}\times G^{-1}(\cdots) + \lfloor q/2\rfloor\cdot x = z$

Output: $0 \Leftrightarrow |z| < q/4$

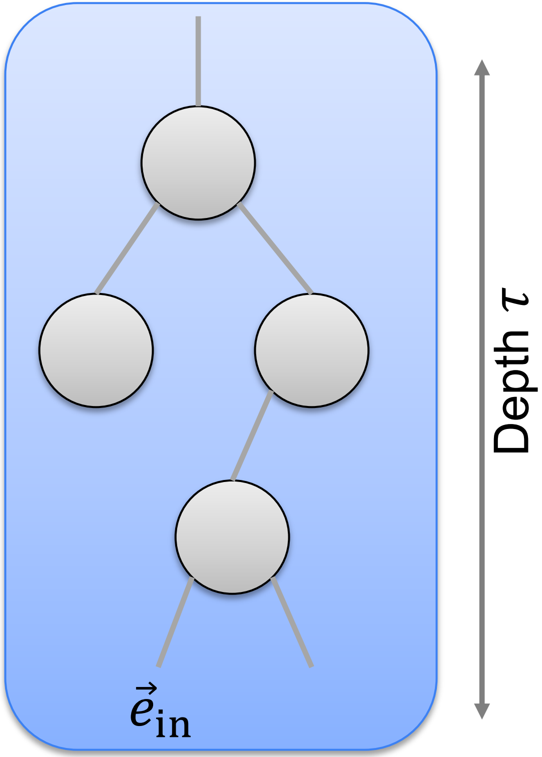
The GSW scheme: homomorphism

$$\boxed{\underline{\textbf{Invariant:}}\ \vec{s}\times C = \vec{e} + x\cdot\vec{s}\times G}$$

$$\boxed{C_{\mathrm{mult}} = C_1\times G^{-1}(C_2)}$$

$$\begin{aligned}
\vec{s}\times C_1\times G^{-1}(C_2) &= (\vec{e}_1 + x_1\cdot\vec{s}\times G)\cdot G^{-1}(C_2)\\
&= \vec{e}_1\times G^{-1}(C_2) + x_1\cdot\vec{s}\times G\times G^{-1}(C_2)\\
&= \vec{e}_1\times G^{-1}(C_2) + x_1\cdot\vec{s}\times C_2\\
&= \vec{e}_1\times G^{-1}(C_2) + x_1\cdot(\vec{e}_2 + x_2\cdot\vec{s}\times G)\\
&= (\vec{e}_1\times G^{-1}(C_2) + x_1\cdot\vec{e}_2) + x_1x_2\cdot\vec{s}\times G\\
&= \vec{e}_{\mathrm{mult}} + x_1x_2\cdot\vec{s}\times G
\end{aligned}$$

The GSW scheme: noise growth



$$\|\vec{e}_{\text{out}}\|_\infty \leq (N+1)^{\tau+1} m \cdot \alpha q$$

Correctness:

$$n \cdot m \cdot (N+1)^{\tau+1} < q/4$$

$$\|\vec{e}_{\text{i+1}}\|_\infty \leq (N+1) \|\vec{e}_{\text{i}}\|_\infty$$

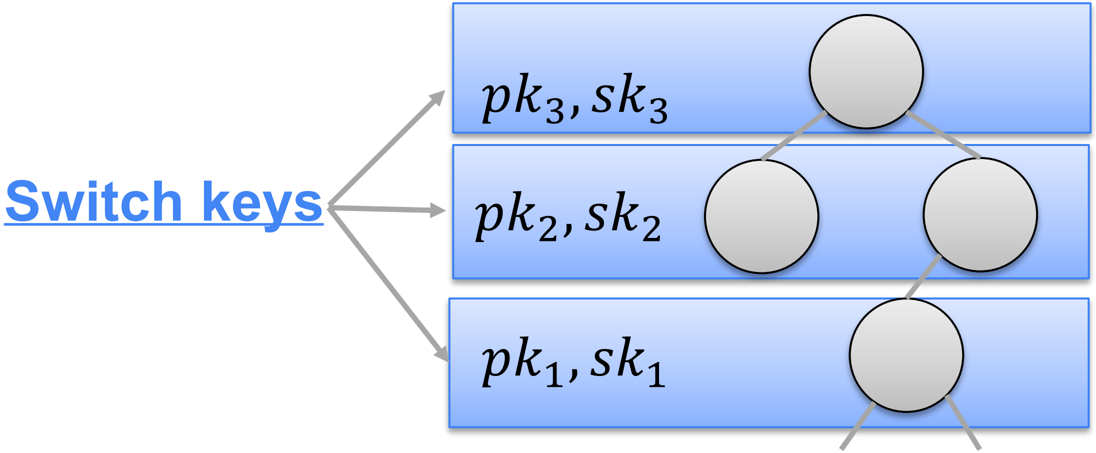
$$\|\vec{e}_{\text{in}}\|_\infty \leq m \cdot n = m \cdot \alpha q$$

The GSW scheme: security

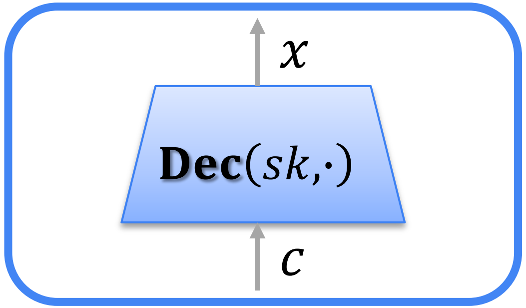
- Similar as in the proof of Regev PKE
- Using LWE we move to a **mental experiment** with $A \leftarrow \mathbb{Z}_q^{n \times m}$
- Hence, by the **leftover hash lemma**, with $m = \Theta(n \log q)$, the statistical distance between $(A, A \times \vec{r})$ and uniform is negligible
  - By a **hybrid argument** over the columns of R, it follows that the statistical distance between $(A, A \times R)$ and uniform is also negligible
  - Thus, the ciphertext **statistically hides** the plaintext

Increasing the homomorphic capacity

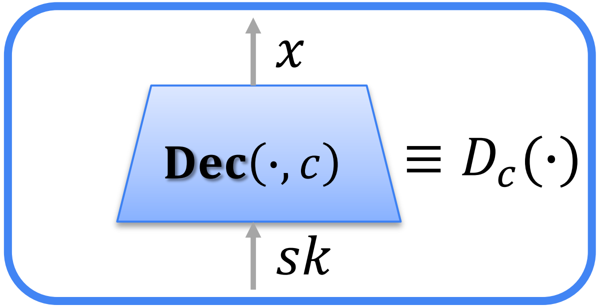
- The only way to increase the homomorphic capacity of GSW is to pick **larger parameters**
- This dependence can be **broken** using a trick by Gentry
- Main idea: Do a few operations, then **switch keys**



How to switch keys



Decryption circuit



Dual view

$$\begin{aligned}\mathbf{Eval}_{pk'}(D_c, aux) &= \mathbf{Eval}_{pk'}\left(D_c, \mathbf{Enc}_{pk'}(sk)\right)\\ &= \mathbf{Enc}_{pk'}(D_c(sk))\\ &= \mathbf{Enc}_{pk'}(x)\end{aligned}$$

Bootstrapping

- Given ciphertexts c_1, c_2, let $D^*_{c_1,c_2}(sk)$ be the **augmented decryption circuit**, defined by

$$\boxed{D^*_{c_1,c_2}(sk) = NAND(D_{c_1}(sk), D_{c_2}(sk))}$$

- We say that Π is **bootstrappable** if its homomorphic capacity includes all the augmented decryption circuits

Theorem. Any **bootstrappable homomorphic** encryption scheme can be transformed into a **compact somewhat homomorphic** encryption scheme

Circular security

- The above scheme is **compact**, but **not fully homomorphic**, as we need a pair of keys **for each level** in the circuit
- A natural idea is to use a **single pair** (pk, sk) and include in pk' a ciphertext $\vec{c}^* \leftarrow \mathbf{Enc}(pk, sk)$
  - Correctness still holds, but the reduction to **semantic security breaks**
- Workaround: Assume **circular security**
  - I.e., $\mathbf{Enc}(pk, 0) \approx \mathbf{Enc}(pk, 1)$ even given $\vec{c}^* \leftarrow \mathbf{Enc}(pk, sk)$
  - GSW is **conjectured** to have this property, but no proof of this fact is currently known

De Componendis Cifris

https:/www.decifris.it

Daniele Venturi Sapienza University of Rome