Fully Homomorphic Encryption

陳榮傑
交通大學資工系
Cryptanalysis Lab
2013/07/03

Agenda

- Fully homomorphic encryption
- Query encrypted data
- Query data privately

FHE (1/6)

- Fully homomorphic encryption
  - Store your encrypted data on the public cloud, an untrusted server
  - Query the data
    - Make Boolean queries on the data
  - Get a useful response from the server
    - Without the server just sending all of the data to you

FHE (2/6)

- Homomorphic encryption
  - Idea: privacy homomorphism
    - Rivest, Adleman, and Dertouzos proposed the concept in 1978.

FHE (3/6)

- Partial homomorphic encryption schemes
  - RSA (multiplication mod *m*) 1977

$$\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) = x_1^e x_2^e \bmod m = (x_1 x_2)^e \bmod m = \mathcal{E}(x_1 \cdot x_2)$$

  - Goldwasser-Micali (XOR) 1984

$$\mathcal{E}(b_1) \cdot \mathcal{E}(b_2) = x^{b_1} r_1^2 x^{b_2} r_2^2 = x^{b_1 + b_2} (r_1 r_2)^2 = \mathcal{E}(b_1 \oplus b_2)$$

  - Paillier (addition mod *m*) 1999

$$\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) = (g^{x_1} r_1^m)(g^{x_2} r_2^m) = g^{x_1 + x_2} (r_1 r_2)^m = \mathcal{E}(x_1 + x_2 \bmod m)$$

FHE (4/6)

- Fully homomorphic encryption
  - A public-key encryption scheme
    - KeyGen, Enc, Dec
    - Evaluate

$$\text{Evaluate}(pk, C, \psi_1, \ldots, \psi_t) \approx \text{Enc}(pk, C(\pi_1, \ldots, \pi_t))$$

for all pk, all circuits C, all ψ_i = Encrypt(pk, π_i).

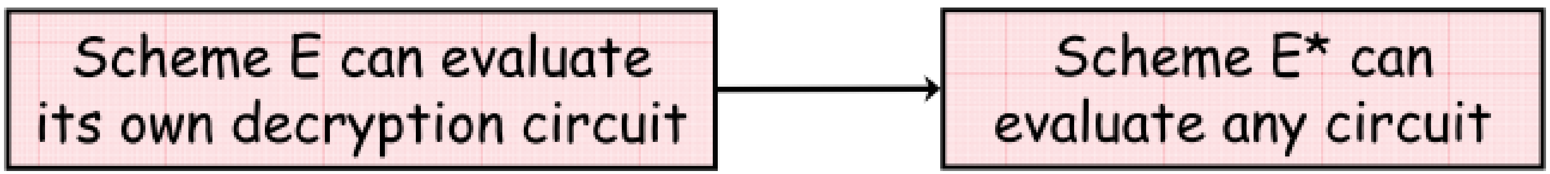
FHE (5/6)

- Gentry proposed the first FHE 2009
  - Support *AND* and *XOR* gates on ciphertext
    - For c_1=Enc(m_1), c_2=Enc(m_2)
      Evaluate(pk, *AND*, c_1, c_2) = Enc(pk, (m_1 *AND* m_2))
      Evaluate(pk, *XOR*, c_1, c_2) = Enc(pk, (m_1 *XOR* m_2))
  - Universal property
    - AND: a *AND* b
    - OR: (a *AND* b) *XOR* (a *XOR* b)
    - NOT: a *XOR* 1

FHE (6/6)

<u>3 Steps</u>

- Step 1 – Bootstrapping:



- Step 2 – Ideal Lattices: Decryption in lattice-based systems has low circuit complexity. *Ideal* lattices used to get + and × ops.

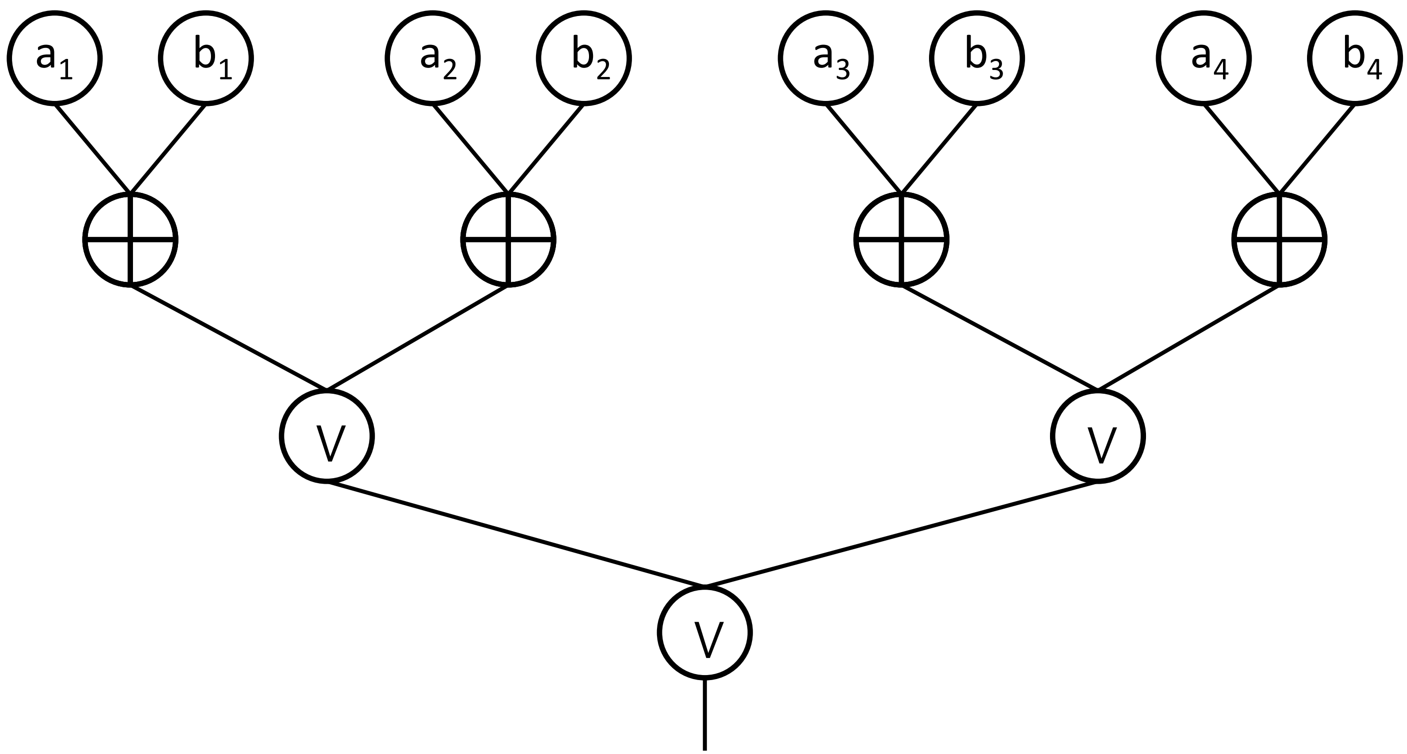
- Step 3 – Squashing the Decryption Circuit: the encrypter helps make decryption circuit smaller by starting decryption itself! Like server-aided decryption.

Query Encrypted Data (1/2)

- Searchable encryption
  - PEKS (Public-key encryption with keyword search)
    - Equality
  - HVE (Hidden vector encryption)
    - Conjunctive of equality, range, and subset
  - IPE (Inner product encryption)
    - Conjunctive/disjunctive

Query Encrypted Data (2/2)

- Test if $a_1a_2a_3a_4 = b_1b_2b_3b_4$



  - Output: 0(match), 1(mismatch)

Query Data Privately (1/2)

- Send an encrypted query regarding stored data
  - E.g., on Google's servers
- Get a useful concise response

Query Data Privately (2/2)

- Alice wants to search something on Google's search engine privately
  - Alice encrypts her query
  - Google encrypts data on the server by Alice's public key and evaluates search circuit for Alice
  - Google returns the encrypted results to Alice
  - Alice decrypts the encrypted results