Applied Cryptography CPEG 472/672 Lecture 12B

Instructor: Nektarios Tsoutsos

Fully Homomorphic Encryption

- Encrypt message m1, m2 to c1, c2
- Apply a function on c1 and c2, get c3
- Decrypt c3 to m3
- Apply same function on m1,m2, get m4
- FHE ensures that m3==m4

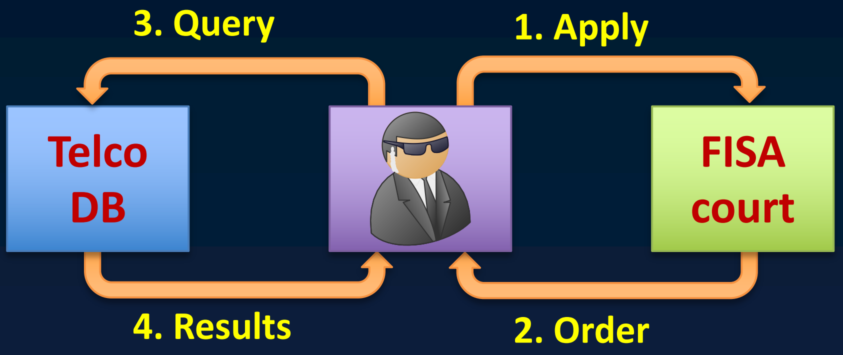
Homomorphic encryption analogy

- Solving a puzzle in a locked glovebox while blindfolded, using thick gloves

FHE applications

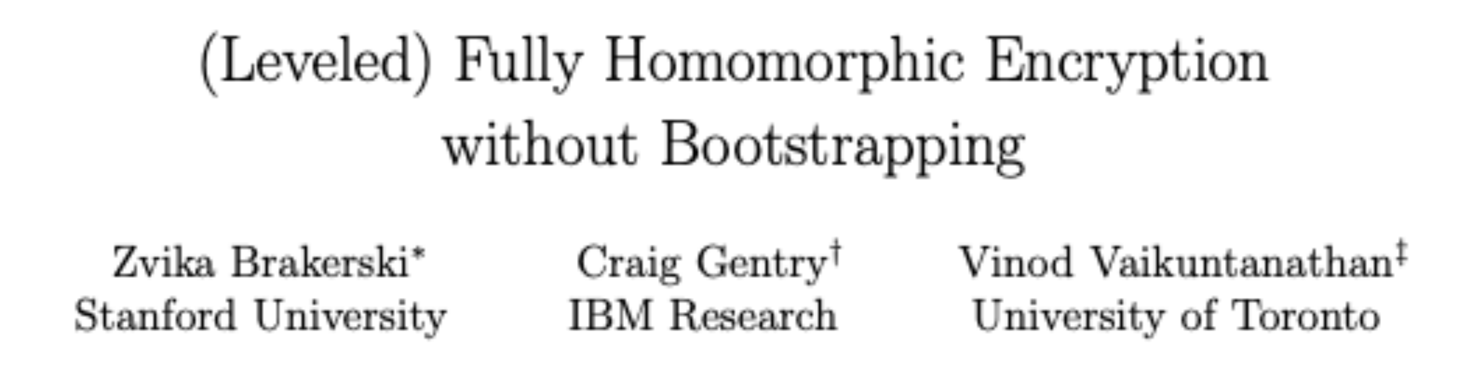
- Census Data

  - Alice wants to compute on sensitive PII to get aggregate results. Bob holds census PII
    - Also: signal filtering, electronic voting etc.
- Publicly verifiable FISA application
  - Does the application conform to the law?
  - Is the law applied correctly?



  - Are the database results legally compliant?

BGV Scheme (2012)



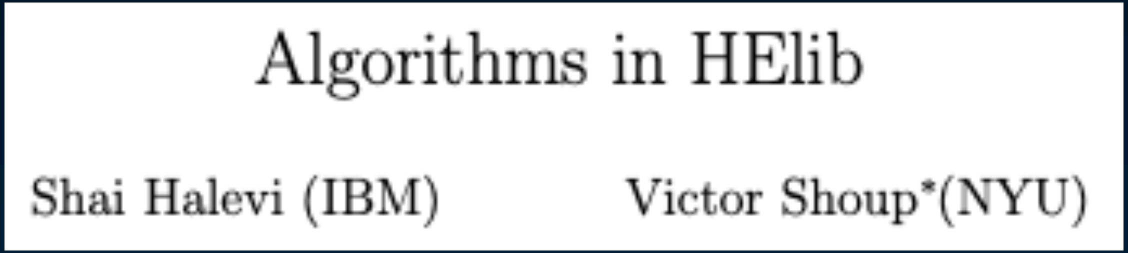
(Leveled) Fully Homomorphic Encryption without Bootstrapping

Zvika Brakerski* Stanford University

Craig Gentry† IBM Research

Vinod Vaikuntanathan‡ University of Toronto

- First implemented in Helib

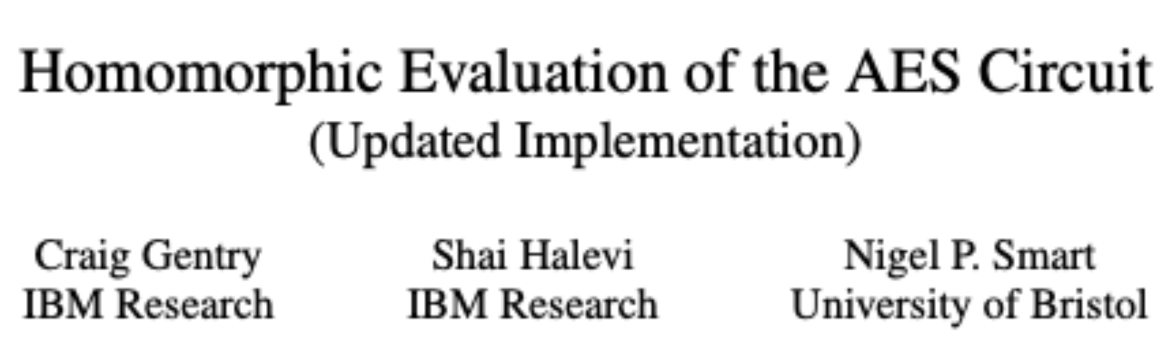


Algorithms in HElib

Shai Halevi (IBM)

Victor Shoup*(NYU)

- Optimized by GHS



Homomorphic Evaluation of the AES Circuit (Updated Implementation)

Craig Gentry IBM Research

Shai Halevi IBM Research

Nigel P. Smart University of Bristol

How does BGV-GHS work?

- Select plaintext modulus *p*
- A plaintext is a “list of *n* integers”
  - They represent coefficients of a polynomial
  - Range of coefficients: -p/2 to p/2
- A ciphertext is 2 lists of *n* integers each
  - Each list represents polynomial coefficients
  - Uses a ciphertext modulus *q*
  - Range of coefficients: -q/2 to q/2
- Special rules to add/mult such lists

General rules to + and * these lists

- Addition (simple)
  - You can add the coefficients individually
  - Need modular reduction (range −p/2 to p/2)
- Multiplication (more involved)
  - The output list size must equal input size
  - Treat input lists as polynomials:
    - Perform polynomial multiplication
      - The list is now twice as long
    - Get remainder of polynomial division with X^n+1
  - Reduce coefficients range to −p/2 to p/2

BGV/GHS ctxt addition

- Each ctxt is a tuple
  - Ctxt0 = (C00,C01)   Ctxt1=(C10,C11)
  - Each component (e.g., C00) is a list of integers from −qi/2 to qi/2
- The homomorphic operation is addition
  - Add C00+C10, Add C01+C11
  - Reduce back to range −qi/2 to qi/2

BGV/GHS ctxt multiplication

- Multiplication result is 3 values
  - The result of multiplying the Ctxt0 tuple with the Ctxt1 tuple returns a triple
    - C0=(scaling factor)*C00*C10
    - C1=(scaling factor)*(C00*C11+C01*C10)
    - C2=(scaling factor)*C01*C11
- We need to get back to a 2-tuple
  - Use 2 special algorithms: ModSwitch, Relin

Final Exam

- Thursday **May 21, 2020**
- 1:00PM - 3:00PM
- Format will be similar to the midterm
- Online: Zoom, Canvas

Hands-on exercises

- HElib demo