Lecture 20:

FHE pt. 2 / course conclusion

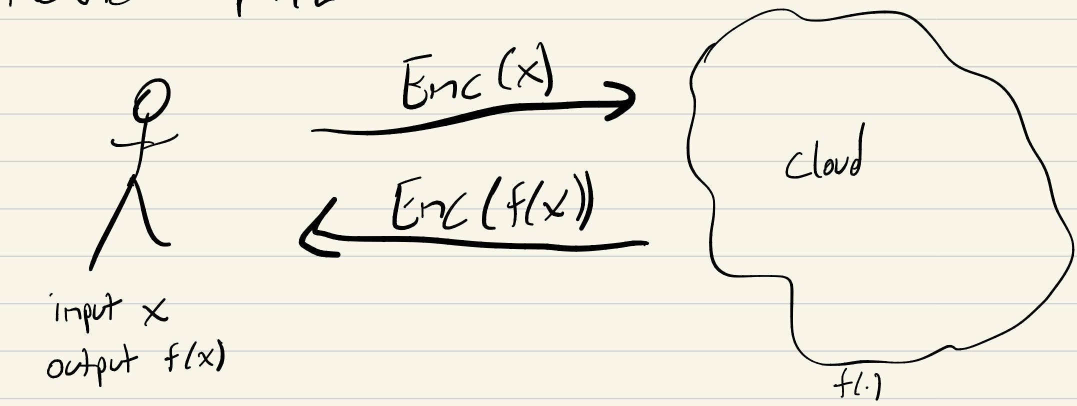
Plan

- Review leveled FHE
- Leveled FHE → FHE (bootstrapping!)
- CS 355 course conclusion

Logistics

Please fill out course evaluations!
HW5

Review: FHE



History

1978: problem first posed

Context: Diffie & Hellman introduced public-key crypto in 1976

⋮

2009: Craig Gentry builds first FHE ← Both at Stanford!

↑ CS 355 TA fall '07
office: Gates 492

Last time: "Leveled" FHE

intuition: encryption based on noisy eigenvectors

KeyGen$(1^n) \to \quad \tilde{s} \xleftarrow{R} \mathbb{Z}_q^{n-1} \qquad \vec{s} \leftarrow \begin{pmatrix} \tilde{s} \\ -1 \end{pmatrix} \in \mathbb{Z}_q^n$

Enc$(\vec{s}, \mu) \to \quad A \xleftarrow{R} \mathbb{Z}_q^{m \times (n-1)}$ for $m = n \log q$

$\vec{e} \xleftarrow{R} \chi_B^m$

$$C = \underbrace{(A, A\tilde{s} + \vec{e}) + \mu G}_{m \times n}$$

output $ct \leftarrow \underbrace{\hat{C}}_{m \times m}$

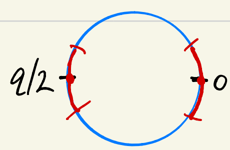
Recall $\hat{()}$ operation is bit decomposition

$\hat{x} = (x_0, \ldots, x_{\log q - 1}) \in \{0,1\}^{\log q - 1}$ s.t. $x = \sum_{i=0}^{\log q - 1} x_i \cdot 2^i$

Dec$(\vec{s}, \hat{c})$: $\hat{C} \cdot G \cdot \vec{s}$

Eval$(\text{"}\times\text{"}, \hat{C}_1, \hat{C}_2) = \hat{C}_1 \cdot \hat{C}_2$

Trick: using bit decomposition vector $\hat{c}$ means that $\hat{c} \cdot \vec{e}$ noise term in multiplication stays small



See last time's notes for how we go from this to supporting any circuit.

What does G look like?

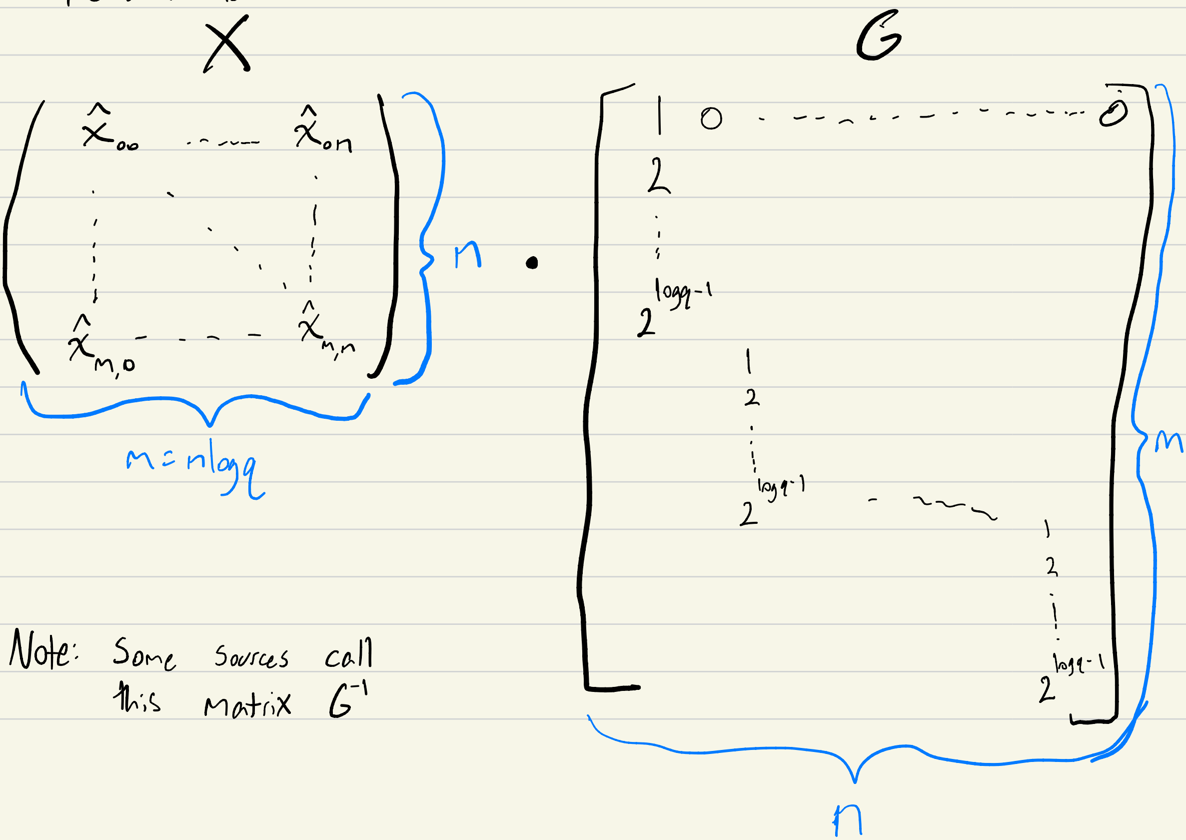
G converts bit decomposition to a single element of $\mathbb{Z}_q$

i.e. it computes $\sum_{i=0}^{\log q - 1} x_i \cdot 2^i$

So for one element:

$$1 \; \underbrace{(x_0, \ldots, x_{\log q - 1})}_{\log q} \bullet \underbrace{\begin{pmatrix} 2^0 \\ 2^1 \\ \vdots \\ 2^{\log q - 1} \end{pmatrix}}_{1} \Big\} \log q$$

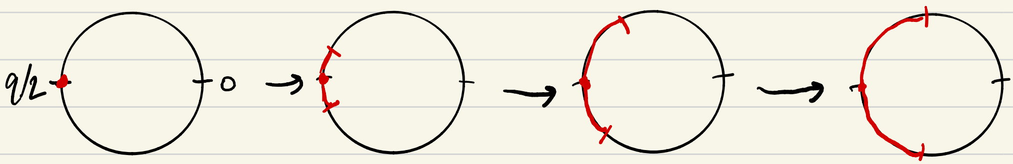
For a whole matrix:



Note: Some sources call this matrix G^{-1}

Why "Leveled"?

Each FHE operation increases noise.



Eventually, the noise gets so big that you can't tell if the message is 0 or 1 anymore.

More formally:

$$\text{Dec}(\vec{s}, \hat{C}_1 \cdot \hat{C}_2) = \hat{C}_1 \cdot \hat{C}_2 \cdot G \cdot \vec{s}$$

(See steps in notes from last time)

$$= M_1 \cdot M_2 \cdot G \cdot \vec{s} + M_2 \cdot \vec{e}_1 + \hat{C}_1 \cdot \vec{e}_2$$

$M_2 \cdot \vec{e}_1$ ↑ at most noise from $\hat{c}_1$

$\hat{C}_1 \cdot \vec{e}_2$ ↑ at most M times noise from $\hat{c}_2$

Note that in previous attempt, a single multiplication caused noise to grow to $O(q)$, but now we can do many multiplications before the noise gets too big — but not an unlimited number.

How do we go from this to a full FHE?

Bootstrapping

A technique to refresh a very noisy ciphertext into an only slightly noisy ciphertext.

Observation: FHE decryption is just some circuit

$$f(\cdot) = \mathrm{Dec}(\cdot, ct)$$

such that $f(\vec{s}) = \mu$

Key insight: We can evaluate this circuit inside an FHE!

$$\mathrm{Eval}(f, \mathrm{Enc}(\vec{s}, \vec{s})) \rightarrow \mathrm{Enc}(\vec{s}, \mu)$$

- f: $\mathrm{Dec}(\cdot, ct)$ with potentially noisy ciphertext
- $\mathrm{Enc}(\vec{s}, \vec{s})$: fresh encryption with no noise
- $\mathrm{Enc}(\vec{s}, \mu)$: encryption with whatever noise is generated by evaluating Dec. But noise <u>does not</u> depend on noise in ct

Leveled FHE → full FHE: Evaluate whatever you want on the input ciphertexts, but whenever noise gets too high, pause and bootstrap to reduce noise.

Caveat 1: this doesn't work if $Dec(\cdot, ct)$ itself is so deep that evaluating it makes a fresh ciphertext too noisy.

Let max depth of leveled FHE be L
Let depth of $Dec(\cdot, ct)$ be d

Then bootstrapping works if $L > d$

Our leveled FHE is good for this because decryption has depth $O(\log q)$: multiplicative depth of each matrix mult is 1, and comparison operation has depth $O(\log q)$.

Caveat 2: $Enc(\vec{s}, \vec{s})$ is an encryption of a secret key under itself. To prove this secure we need to make an additional "circular security" assumption.

Caveat 3: why aren't we using FHE all over the place?

Performance cost of FHE, especially for bootstrapping, is quite high.

performance note: Although FHE is slow, lattice crypto in general is not. In fact, lattice crypto is sometimes faster than elliptic curve crypto, at the cost of larger ciphertexts/noisiness.

Course Review

Lec 1-3: OWF → PRFs, hybrid arguments, RO model, commitments

Lec 4-5: Cryptanalysis

Lec 6-7: Elliptic Curve Crypto & Pairings

Lec 8-12: Zero Knowledge (IP, ZK, Schnorr, Sigma protocols, PoK, NIZKs, Fiat-Shamir, SNARGs)

Lec 13-17: Privacy enhancing technologies (2PC, Secret sharing, MPC, DP, PIR)

Lec 18-20: Lattice-based Crypto & FHE

You now have the tools to:

- reason about security in systems you design & build
- recognize crypto tools that can benefit your projects
- understand technical aspects of policy debates around privacy & encryption
- expand your knowledge of crypto by following the latest developments and making your own research contributions.

Some interesting topics we did not cover

Some theoretical crypto topics: attribute based encryption

functional encryption

obfuscation

Some applied crypto topics: Group Signatures

Anonymous credentials

CCA-secure PKE & Signcryption

Other post-quantum crypto approaches

Where to read about the latest & greatest in crypto:

Flagship IACR conferences:

CRYPTO
Eurocrypt
Asiacrypt

Top Security conferences:

IEEE Security & Privacy ("Oakland")
Usenix Security
ACM CCS

See also
Annual Real World crypto symposium (RWC)
Cryptology ePrint Archive: eprint.iacr.org

More Crypto at Stanford

Crypto/Security events at Stanford: (each one has a mailing list you can sign up for)

- Security lunch : Wed 12-1pm securitylunch.stanford.edu
- Security Seminar: crypto.stanford.edu/seclab/sem.html
- Bay Area Crypto Day: ~twice a year, alternates b/w Stanford and Berkeley
  bacrypto.github.io

Feel free to reach out to us if you have questions about getting involved in research or doing a PhD in CS.

Reminder: Course evaluations

Thank you!!