ENCRYPTED MESSAGES FROM THE HEIGHTS OF CRYPTOMANIA

Craig Gentry, IBM

Joint work with Sanjam Garg, Shai Halevi, Amit Sahai, Brent Waters

Supported by IARPA contract number D11PC20202

TCC 2013

Tokyo, Japan

Fully Homomorphic Encryption (FHE)

- Awesome!
  - I give the cloud encrypted program E(P)
  - For (possibly encrypted) x, cloud can compute E(P(x))
  - I can decrypt to recover P(x)
  - Cloud learns nothing about P, or even P(x)

- Problem…
  - What if I **want** the cloud to learn P(x) (but still not P)?
  - So that the cloud can take some action if P(x) = 1.

Obfuscation

- Obfuscation
  - I give the cloud an "encrypted" program E(P).
  - For any input x, cloud can compute E(P)(x) = P(x).
  - Cloud learns "nothing" about P, except $\{x_i, P(x_i)\}$.
- Barak et al: "On the (Im)possibility of Obfuscating Programs"
- Difference between obfuscation and FHE:
  - In FHE, cloud computes E(P(x)) and can't decrypt to get P(x).
- Step in right direction? Modify FHE so that cloud can detect when some special value, say '0', is encrypted
  - A *zero test* (or *equality test*)

FHE with a Zero Test

- Seems as powerful as FHE (if message space is large).
- To regain semantic security:
  - Use a composite N = pq message space
  - Mod-p part for message, mod-q part for randomness

- Perhaps more powerful
  - Control when cloud extracts information
  - E.g, when residues mod-p and mod-q “align” to 0.

- Difficulty:
  - Can we enable zero-testing without breaking the FHE scheme?

Black Box Fields (BBFs) [BL96]

- BBFs:
  - Each element x encoded by arbitrary string [x] (maybe more than 1)
  - Given [x], [y], BBF oracle provides [x+y] and [x·y]
  - Equality test: Given [x], [y], Eq([x],[y]) outputs 1 iff x = y.
- Sort of like FHE scheme with zero test

Attacks on Black Box Fields

- BBF Problem: Given encoding [x] of x in F_p, output x.
  - Solvable in sub-exponential time.
    - Technique: Solve $DL_A(x,y)$ over elliptic curve with smooth order.
  - Solvable in quantum polynomial time [vDHI03]

- Corollary: FHE over F_p with a zero test is breakable in subexponential or quantum polynomial time.

- Not fatal, but troubling.
- Anyway, we don't have a construction of FHE with zero test.

Somewhat HE (SWHE) with a Zero Test

- SWHE
  - Can evaluate functions of degree bounded by some polynomial in the security parameter
- SWHE with zero test
  - Boneh-Lipton subexponential attack does not apply. Nor does quantum attack.
  - Turns out to be like a multilinear map!

Bilinear Maps

- Cryptographic bilinear map (for groups)
  - Groups G_1, G_2 of order p with generators g_1,g_2
  - Bilinear map:

$$e : G_1 \times G_1 \longrightarrow G_2 \quad \text{where}$$

    - $e(g_1^a, g_1^b) = g_2^{ab}$ for all a,b 2 F_p.
- Bilinear DDH: Given $g_1^{a_1}$, $g_1^{a_2}$, $g_1^{a_3}$ 2 G_1, and h2 G_2, distinguish whether $h = g_2^{a_1 a_2 a_3}$ or is random.
- Bilinear group ≈ Degree-2 HE with equality test
  - $Enc_i(a) \longrightarrow g_i^a$

Multilinear Maps

- Cryptographic k-multilinear map (for groups)
  - Groups $G_1, \ldots, G_k$ of order p with generators $g_1, \ldots, g_k$
  - Family of maps:

$$e_{i,j} : G_i \times G_j \to G_{i+j} \quad \text{for } i+j \leq k, \text{ where}$$

    - $e_{i,j}(g_i^a, g_j^b) = g_{i+j}^{ab}$ for all a,b 2 F_p.
  - Notation Simplification: $e(g_{i_1}, \ldots, g_{i_t}) = g_{i_1+\ldots+i_t}$.
- k-linear DDH: Given $g_1^{a_1}, \ldots, g_1^{a_{k+1}}$ 2 G_1, and h 2 G_k, distinguish whether $h = g_k^{a_1 \cdots a_{k+1}}$ or is random.
- k-linear group ≈ Degree-k SWHE with a zero test
  - $Enc_i(a) = g_i^a$. Eval degree-k polys on level-1 encodings.

Probabilistic Encodings and Extraction

- For multilinear groups, encoding is deterministic
  - Zero test is immediate
  - Extraction: Parties that arrive at the same encoding can easily extract a shared key

- For a SWHE scheme with a zero test, encoding is probabilistic
  - A zero test doesn't imply an extraction procedure.
  - So, let's assume an extraction procedure for now.

Multilinear Maps: Applications

Thanks to Brent for some of these slides

Applications

- Easy Application: (k+1)-partite key agreement using k-linear map [Boneh-Silverberg '03]:
  - Party i generates level-0 encoding of a_i.
  - Party I broadcasts level-1 encoding of a_i.
  - Each party separately computes key $e(g_1, \ldots, g_1)^{a_1 \cdots a_{k+1}}$.
  - Secure assuming k-linear DDH: Given $g_1^{a_1}, \ldots, g_1^{a_{k+1}}$ 2 G_1, and h2 G_n, hard to distinguish whether $h = g_k^{a_1 \cdots a_{k+1}}$.

- More interesting applications:
  - Attribute-based encryption for circuits [GGHSW12].
  - Witness encryption [GGSW13]

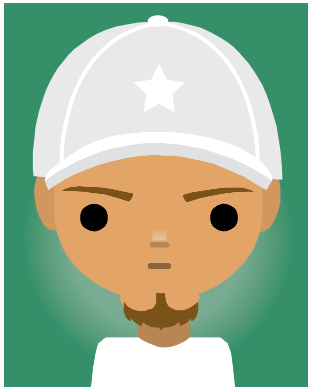
Attribute Based Encryption (ABE)

Setup$(1^\lambda, F)$: takes as input a security parameter and a class of functions $F = \{f : \{0,1\}^n \to \{0,1\}\}$.

Outputs master secret and public keys MSK, MPK.

KeyGen(MSK,f): Authority uses MSK to generate a key SK_f for the function f.

f represents a user's "key policy" that specifies when it can decrypt.



Encryption(MPK, A, M): Outputs CT that encrypts M under string $A \in \{0,1\}^n$.

"A" may be "attributes" needed by decrypter.

Decryption(SK_f, CT): Decrypter recovers M iff $f(A)=1$.

Prior Work on ABE

- F = simple functions in prior ABE schemes
  - Example: F = formulas.
  - For F = circuits, prior schemes have exponential complexity
- Tools:
  - Bilinear maps [SW05,GOSW06,…]
  - Lattices (learning with error (LWE)) [Boyen13].
- Big open problem: Efficient *ABE for circuits*.
  - Just like *HE for circuits* was open.
  - Note: Monotone circuits → general circuits.

ABE for Circuits using MMaps [GGHSW12]

AND gate: similar to OR gate

$L = \#$ levels; $k = L+1$; n-bit inputs

k-linear map: $G_1, \ldots, G_k$; $g_1, \ldots, g_k$

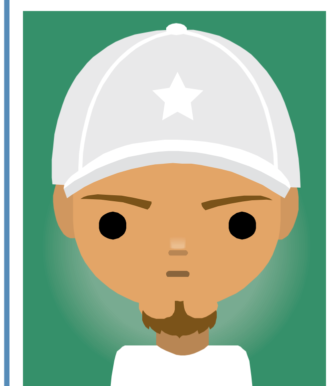
MSK $= g_1^{\alpha}$ for uniform α in F_p

MPK $= g_1, h_1, \ldots h_n \in G_1$, $g_k^{\alpha} \in G_k$

KeyGen: Random $r_w \leftarrow F_p$ for each wire w in circuit, except $r_w = \alpha$ for output wire.

OR gate: Input wires x,y and output wire w at depth j. Choose random a_w, b_w in F_p. Give $g_1^{a_w}, g_j^{r_w - a_w r_x}$, $g_1^{b_w}, g_j^{r_w - b_w r_y}$.

AND gate: Give $g_1^{a_w}, g_1^{b_w}, g_j^{r_w - a_w r_x - b_w r_y}$.



Encryption: Enc. M for attributes $A \in \{0,1\}^n$

$s \leftarrow F_p$, $CT = M \cdot g_k^{\alpha s}$, g_1^{s}, $\forall y \in A$, h_y^{s}

Decryption: Gate-by-gate to output wire, compute $g_{j+1}^{r_w s}$ for wires at depth j

Summary of ABE for Circuits

- Now we have ABE for arbitrarily complex policies
  - The scheme is quite simple.
  - Ciphertexts are "succinct"
    - Do not grow with size of circuit.
    - Grow with size of input.
    - Grow with depth of circuit (due to our construction of mmaps)
  - Security: based on k-linear DDH

- Interesting concurrent work:
  - [GVW13] ABE for circuits based on LWE

Witness Encryption

Can we encrypt a message so that it can opened only by a recipient who knows a *witness to a NP relation*?

Like a proof of the Riemann Hypothesis.

- Unlike ABE:
  - No "authority" in the system
  - No "secret key" per se
- Related concepts:
  - Rudich'89: Comp. secret sharing for NP-comp access structures

Witness Encryption: Definition

NP language L with witness relation $R(\cdot,\cdot)$

$\text{Encrypt}(1^\lambda, x, M) \rightarrow CT$

$\text{Decrypt}(CT, w) \rightarrow (M \cup \perp)$

Notice the gap.
No immediate security
promises when x in L.

Correctness

$\forall \lambda, M, x \in L$ s.t. $R(x,w) =$ [illegible] we have $\text{Dec}(\text{Enc}(1^\lambda, x, M), w) = M$

Security

If x is not in L, then $\text{Enc}(1^\lambda, x, M_0) \approx_c \text{Enc}(1^\lambda, x, M_1)$

Exact Cover Problem [Karp72]

- Problem: x includes n and subsets $T_1, \ldots, T_m \subseteq [n]$
- Witness: $I \subseteq [m]$ s.t. $\{T_i : i \in I\}$ partitions $[n]$

- Examples: 4, ({2,3}, {2,4}, {1,4})
  4, ({2,3}, {2,4}, {1})

Our WE Construction (for Exact Cover)

- Encrypt$(1^\lambda, \ (n, (T_1, \dots, T_m \subseteq [n])), \ M \in G_n)$
  - n-linear group family $G_1, \dots, G_n$, generators $g_1, \dots, g_n$.
  - Choose random $a_1, \dots, a_n \in F_p$.

$C = M \cdot g_n^{a_1 \dots a_n}$ $\qquad C_i = \ (g_{|T_i|})^{\prod_{j \in T_i} a_j}$ for all $i \in [m]$

- Decrypt(CT, $w = I = (i_1, \dots, i_t)$)

$$C / e(C_{i_1} C_{i_2}, \dots, C_{i_t})$$

Limitations in Proving

- Suppose we have a black box reduction of WE to some non-interactive assumption. Either:
  - Assumption depends on NP instance
  - Reduction uses enough computation to decide relation R

- Decision No Exact Cover Problem Family

$$\left(n, (T_1, \ldots, T_m \subseteq [n])\right), \qquad \mathcal{G}(1^\lambda, n) \to (G_1, \ldots, G_n)$$

$$a_1, \ldots, a_n, r \leftarrow F_p, \quad C_i = (g_{|T_i|})^{\prod_{j \in T_i} a_j} \text{ for all } i \in [m]$$

$$\text{Distinguish } C = g_n^{a_1 \ldots a_n} \text{ from } g_n^r.$$

Fun Application of WE:
Public Key Enc with Super-Fast KeyGen

- KeyGen(1^λ):
  - Let $F : \{0,1\}^\lambda \longrightarrow \{0,1\}^{2\lambda}$ be a PRG.
  - SK = PRG seed $s \in \{0,1\}^\lambda$.   PK = F(s).

- Encrypt(PK, M)
  - Karp-Levin reduction $x \in L$ iff PK is in range of F.
  - $\text{Encrypt}_{WE}(1^\lambda, x, M) \longrightarrow CT$

- Decrypt(SK = s, CT)
  - $s \longrightarrow$ witness w
  - $\text{Decrypt}_{WE}(CT, w) \longrightarrow M$

Proof Sketch for PKE Scheme

- PRG security → indistinguishable whether PK is a PRG output or truly random

- If PK truly random, then x not in L (with high prob), and we can rely on soundness of WE scheme

Multilinear Maps from Ideal Lattices

Cryptographic Multilinear Maps: Do They Exist?

- Boneh and Silverberg '03 say it's unlikely cryptographic m-maps can be constructed from abelian varieties:

"We also give evidence that such maps might have to either come from outside the realm of algebraic geometry, or occur as '*unnatural*' *computable maps arising from geometry*."

- Unnatural geometric maps: Why not the 'noisy' mappings of lattice-based crypto?

Overview of Our Noisy M-Maps

- Encoding: m → g_i^m (groups) becomes m → $Enc_i(m)$ for us.
  - $Enc_i(m)$ is a "level-i encoding of m".
  - Our encoding system builds on the NTRU encryption scheme.

- Zero test: For k-linear maps, we use a level-k zero tester to test equality of level-k encodings and extract keys.

- Repairs: Zero testers cause security issues to fix.
  - Certain aspects of the "message space" of our encodings must be kept secret.
  - Our params only enable encoding of random elements.
    - Sufficient for our ABE and WE applications.

Starting Point: the NTRU Cryptosystem

- NTRU's concept: The following are indistinguishable:
  - A random element of $R_q = Z_q[x]/(x^N-1)$. ($q=127, N=257$)
  - A ratio $a/b \in R_q$ of "small" elements. That is, a and b are polynomials in R_q with small coefficients – e.g. in $\{-1,0,1\}$.

- Secret key: uniform $z \in R_q$.
- Public key: $c_1 = a_1/z$, $c_0 = a_0/z \in R_q$ with a_1, a_0 small.
  - Let p be a small integer or ideal generator w/ $\gcd(p,q)=1$ ($p=3$)
  - Make sure $a_1 = 1 \bmod p$ and $a_0 = 0 \bmod p$.

- Ciphertexts: A ciphertext that encrypts $m \in R_p$ has the form $e/z \in R_q$, where e is "small" and $e = m \bmod p$.
  - c_1 encrypts 1, and c_0 encrypts 0.

NTRU Cryptosystem: Encrypt, Decrypt

- Encrypt(PK,m) for "small" m
  - Generate random "small" $r \in R_q$.
  - Output ciphertext $CT = m \cdot c_1 + r \cdot c_0 \in R_q$.
  - Observe: $CT = (ma_1+ra_0)/z \in R_q$, where ma_1+ra_0 is "small" and equals m mod p.
  - Encryption implicitly uses additive homomorphism of NTRU.

- Decrypt(SK,CT):
  - Compute $CT \cdot z = ma_1+ra_0 \in R_q$.
  - Get ma_1+ra_0 exactly (unreduced mod q) since it is "small".
  - Reduce modulo p to recover m.

Basic NTRU: Summary

- Ciphertext that encrypts m has form e/z, where
  - e is small
  - e = m mod p
  - z is the secret key

- To decrypt, multiply by z and reduce mod p.

- Public key has encryptions of 1 and 0 (c_1 and c_0). To encrypt m, multiply m with c_1 and add "random" encryption of 0.

NTRU: Additive Homomorphism

- Given: CT_1, CT_2 that encrypt m_1, m_2 2 R_p.
  - $CT_i = e_i/z$ 2 R_q where e_i is small and $e_i = m_i \bmod p$.

- Set $CT = CT_1 + CT_2$ 2 R_q and $m = m_1 + m_2$ 2 R_p.

  Then CT encrypts m.
  - $CT = (e_1 + e_2)/z$ where $e_1 + e_2 = m \bmod p$ and $e_1 + e_2$ is "sort of small". It works if $|e_i|$ « q.

NTRU: Multiplicative Homomorphism

- Given: CT_1, CT_2 that encrypt $m_1, m_2 \in R_p$.
  - $c_i = e_i / z \in R_q$ where e_i is small and $e_i = m_i \bmod p$.

- Set $CT = CT_1 \cdot CT_2 \in R_q$ and $m = m_1 \cdot m_2 \in R_p$.
  Then CT encrypts m under z^2 (rather than under z).
  - $CT = (e_1 \cdot e_2) / z^2$ where $e_1 \cdot e_2 = m \bmod p$ and $e_1 \cdot e_2$ is "sort of small". It works if $|e_i| \ll \sqrt{q}$.

NTRU: Any Homogeneous Polynomial

- Given: CT_1, …, CT_t encrypting m_1,…, m_t.
  - $CT_i = e_i / z$ 2 R_q where e_i is small and $e_i = m_i$ mod (p).

- Let f be a homogeneous polynomial of degree d. Set CT= f(CT_1, …, CT_t)2 R_q, m = f(m_1, …, m_t)2 R_p

  Then CT encrypts m under z^d.
  - CT = f(e_1, …, e_t)/ z^d where f(e_1, …, e_t)=m mod p and f(e_1, …, e_t) is “sort of small”. It works if $|e_i|$ « $q^{1/d}$.

Homomorphic NTRU: Summary

- Ciphertext that encrypts m at "level d" has form e/z^d:
  - e is small
  - e = m mod p
  - z is the secret key

- To decrypt, multiply by z^d and reduce mod p.

- How homomorphic?: For any degree-d homogeneous $f(x_1, \ldots, x_t)$, we get a "level-d" encryption of $f(m_1, \ldots, m_t)$ from "level-1" encryptions $\{CT_i = e_i/z\}$ of $\{m_i\}$, if e_i's are small enough.

- "Noise" – size of numerator – grows exp. with degree.
  - Works OK if d is (sublinear) polynomial in security param.

Adding a Zero/Equality Test to NTRU

- Given level-k encodings $CT_1 = e_1/z^k$ and $CT_2 = e_2/z^k$, how do we test whether they encode the same m?
- Fact: If they encode same thing, then $e_1-e_2 = 0 \bmod (p)$. Moreover, $(e_1-e_2)/p$ is a "small" polynomial.
- Zero-Testing parameter:
  - $a_{ZT} = h \cdot z^k/p$ for "medium-size" h (e.g. $|h| \approx q^{3/4}$)
  - $a_{ZT}(CT_1-CT_2) = h(e_1-e_2)/p$
    - If CT_1, CT_2 encode same thing, then denominator p disappears
      - $|h(e_1-e_2)/p|$ is "medium-sized", unreduced mod q.
      - $a_{ZT} \cdot CT_1$ and $a_{ZT} \cdot CT_2$ have same most significant bits → extract key
    - Otherwise, denominator p "randomizes" things mod q.
- Small ideal generator p must be secret. Ideal (p) is public.

Summary of Our Noisy M-Maps

- Level-i encoding of $m \in R_p$ has form e/z^i, where
  - e is small
  - $e - m \in$ ideal (p)
  - z is secret
- Public params have encodings of 1 and 0 (c_1 and c_0).
- To encode a random element, sample "small" m, multiply m with c_1 and add "random" encoding of 0.
- Homomorphisms work as in NTRU
- Level-k zero tester $h \cdot z^k/p$ enables zero-testing at level k or below.

Cryptanalysis

Security of NTRU

- Lattice attacks on NTRU apply to our n-linear maps.
  - NTRU semantically secure if ratios $g/f \in R_q$ of "small" elements are hard to distinguish from random elements
  - NTRU can be broken via lattice reduction (eventually)

- [Lenstra, Lenstra, Lovász '82]: Given a rank-n lattice L, the LLL algorithm runs in time poly(n) and outputs a 2^n-approximation of the shortest vector in L.
  - [Schnorr'93]: 2^k-approximates SVP in $2^{n/k}$ time (roughly)

Attacks that Exploit the Zero Tester

- Concept of the attack:
  - The zero-tester is not an "oracle"
  - Zero-testing could actually leak useful information

- Attack in practice
  - Actually, our zero test does leak *useful* information.
  - Our m-maps are imperfect
  - Some assumptions that are true for "generic" m-maps are false for our m-maps

Source Group Decision Assumptions

- Example: Decision Linear Assumption in bilinear groups.
  - Distinguish $(f, g, h, f^x, g^y, h^{x+y})$ from (f, g, h, f^x, g^y, h^z).
  - All elements in source group G_1, none in target group G_2.

- k-linear source group assumption:
  All encodings are at level $\leq$ k-1.

- Source group assumptions false with our m-maps
  - if params includes level-1 encodings of 0

Target Group Decision Assumptions

- Example: k-linear DDH or Decision No Exact Cover.
- Target group assumption for k-linear m-maps:
  The two distributions are statistically the same, except for encodings at level k.
- Target group assumptions for our m-maps seem ok.

k-linear DDH for GGH encodings: Given

- Params: Level-1 encodings c_0, c_1 of 0 and 1 and level-k zero-testing parameter $a_{zt} = hz^k/p$
- Level-1 encodings e_i/z of m_i for $i \in [k+1]$
- Level-k encoding of either $m_1 \cdots m_{k+1}$ or random

Distinguish which is the case.

Flavor of the Attack

- An "attack" on low-level encodings
  - Take a level-i encoding e/z^i for $i \le k-1$ (low-level encoding)
  - Multiply it with
    - A level-(k-i) encoding of 0 (from params)
    - The level-k zero tester
  - Extract useful information about what is encoded

- What is leaked?
  - E mod (p) = m mod (p)
  - Not m itself – i.e., not a small representative of m's coset
  - Not a "level-0 encoding" of m

- Preventing the attack on level-k encodings
  - (p) is public, but small p is secret. No "level-0 encoding" of 0.

Summary and Future Directions

Summary

- "Noisy" cryptographic multilinear maps
  - SWHE with a zero test
  - Built on the NTRU cryptosystem
  - Stronger computational assumptions than NTRU.
- Applications:
  - ABE for Circuits
  - Witness Encryption

Future Directions

- Security
  - Need more cryptanalysis of our m-maps
  - M-maps based on better assumptions (like LWE)?
- Applications
  - Functional encryption?
  - Some types of obfuscation?

Thank You! Questions?

Revisiting Multilinear DDH

- Ineffective attack: Multiply the k+1 contributions to get an encoding at level k+1; not useful (similar to bilinear groups)
  - $(E/z^{k+1}) \cdot (hz^k/p) = Eh/pz$. Can't get rid of denominator.

Attacks that Exploit the Zero Tester

- Additional attacks:
  - The principal ideal I = (p) is not hidden.
    - Recall $a_{zt} = hz^k/p$, $h_0 = a_0/z$ and $h_1 = a_1/z$ with $a_0 = c_0 p$.
    - The terms $a_{zt} \cdot h_0^{i} \cdot h_1^{k-i} = h \cdot c_0^{i} \cdot p^{i-1} \cdot e_1^{k-i}$ likely generate I.
  - But we must hide p itself
    - An attacker can break our scheme with a "small" generator p' of I = (p)
  - An attacker that finds a good basis of I can break our scheme.

What Does Zero Testing Leak?

- Let e/z^i be a level-i encoding of m for i < k.

$$(e/z^i) \cdot c_1^{k-1-i} \cdot c_0 \cdot a_{ZT} = (e/z^i) \cdot (a_1/z)^{k-1-i} \cdot (a_0/z) \cdot (hz^k/p)$$
$$= e \cdot a_1^{k-1-i} \cdot a_0' \cdot h$$

  - $e \cdot a_1^{k-1-i} \cdot a_0' \cdot h$ unreduced mod q.
  - We get e's coset mod p.
  - We get a "bad level-0 encoding" of m.
    - A "good" level-i encoding has a small numerator.

Using a Good Basis of I

- Player i's DH contribution: a level-1 encoding of a_i.

- Easy to compute a_i's coset of I. (Notice: this is different from finding a "small" representative of a_i's coset, a level-0 encoding of a_i.)
  - Compute level-(n-1) encodings of 1 and a_i: e/z^{n-1}, e'/z^{n-1}.
  - Multiply each of them with a_{zt} and $h_0 = c_0 p/z$.
    - We get bec_0 and $be'c_0$.
  - Compute $be'c_0/bec_0 = e'/e$ in R_p to get a_i's coset.

- Spoofing Player i: If we have a good basis of I, player i's coset gives a level-0 encoding of a_i. The attacker can spoof player i.

Dimension-Halving for Principal Ideal Lattices

- There are better attacks on principal ideal lattices than on general ideal lattices. (But still inefficient.)

- [GS'02]: Given
  - a basis of I = (u) for u(x) 2 R and
  - u's relative norm u(x)ū(x) in the index-2 subfield $Q(\zeta_N + \zeta_N^{-1})$,

  we can compute u(x) in poly-time.

- Corollary: Set v(x) = u(x)/ ū(x). We can compute v(x) given a basis of J = (v).
  - We know v(x)'s relative norm equal 1.

Dimension-Halving for Principal Ideal Lattices

- Attack given a basis of $I = (u)$:
  - First, compute $v(x) = u(x)/\bar{u}(x)$.
  - Given a basis $\{u(x)r_i(x)\}$ of I, multiply by $1+1/v(x)$ to get a basis $\{(u(x)+\bar{u}(x))r_i(x)\}$ of $K = (u(x)+\bar{u}(x))$ over R.
  - Intersect K's lattice with subring $R' = Z[\zeta_N + \zeta_N^{-1}]$ to get a basis $\{(u(x)+\bar{u}(x))s_i(x) : s_i(x)\ 2\ R'\}$ of K over R'.
  - Apply lattice reduction to lattice $\{u(x)s_i(x) : s_i(x)\ 2\ R'\}$, which has half the usual dimension.

A "Straight Line Program (SLP)" Model of Attacks on Our M-Maps

- SLP attack model: Attacker can $+,-,\times,\div$ encodings in R_q (until it gets a level-i encoding of 0, $i \leq k$).
  - View encodings as formal rational polynomials P/Q.
  - The ops $+,-,\times,\div$ give more rational polynomials.
  - Which ones can it compute?

- Params: a_1/z, a_0/z, $h \cdot z^k/p$
- Weight the variables
  - Set $w(a_i) = w(z) = w(p) = 1$ and $w(h) = 1\text{-}k$.
  - $w(a_i/z) = 0$. Weight of all terms above is 0.

- Given params, $+,-,\times,\div$ only yield terms of weight 0.

SLP Attacks Don't Break Target Group Assumptions

- SLP attacker against MDDH
  - First attack: Try to compute level-k encoding E/z^k of $m_1 \cdots m_{k+1}$ from params and the parties' encodings e_i/z.
    - E/z^k must have weight zero.
    - E must have weight k.
    - But E must have $e_1 \cdots e_{k+1}$ inside it; else hopeless.
    - Now numerator's weight is too large. Must reduce weight using h (it is the only negative weight term).
    - But h is middle size, so numerator is not small anymore.
  - Second attack: Try to find nontrivial relation among the encodings of the MDDH instance.
    - Analysis is similar: relation must have degree $\geq k+1$.

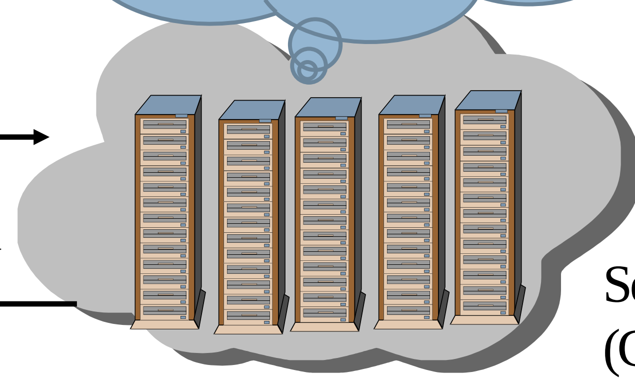
Homomorphic Encryption

The special sauce! For security parameter k, Eval's running should be Time(f)·poly(λ)

"I want 1) the cloud to process my data 2) even though it is encrypted.

Alice
(Input: data x, key k)

Run
Eval[f, Enc_k(x)]
= Enc_k[f(x)]



Server
(Cloud)

Enc_k(x)

function f

This could be encrypted too.

Enc_k[f(x)]

f(x)

Delegation: Should cost less for Alice to encrypt x and decrypt f(x) than to compute f(x) herself.