Homomorphic Encryption from Ring Learning with Errors

Michael Naehrig

Technische Universiteit Eindhoven
michael@cryptojedi.org

joint work with
Kristin Lauter (MSR Redmond)
Vinod Vaikuntanathan (University of Toronto)

MSR Cambridge, 10 January 2012

Homomorphic encryption

Example 1: RSA public key encryption

- Let $n = p \cdot q$, $p \neq q$ primes, $\varphi(n) = (p-1)(q-1)$,
- $\mathrm{pk} = (n, e)$, $\gcd(e, \varphi(n)) = 1$,
- $\mathrm{sk} = d = e^{-1} \mod \varphi(n)$.
- Encrypt message $m \in \mathbb{Z}_n$:

$$c = m^e \mod n.$$

- Decrypt ciphertext c: $m = c^d \mod n$.

- Multiplicative homomorphism:
  If $c_1 = m_1^e \mod n$, $c_2 = m_2^e \mod n$, then

$$c_1 \cdot c_2 = (m_1 \cdot m_2)^e \mod n.$$

Homomorphic encryption

Example 2: ElGamal public key encryption in a group $G = \langle g \rangle$

- $\mathrm{sk} = x \in \mathbb{Z}_{|G|}$,
- $\mathrm{pk} = h = g^x$.
- Encrypt $m \in G$: choose $r \in \mathbb{Z}_{|G|}$ at random and compute

$$(c, d) = (g^r, m \cdot h^r).$$

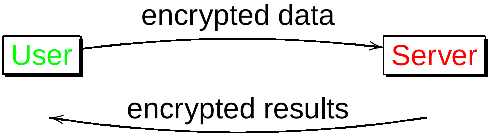
- Decrypt: $m = d \cdot (c^x)^{-1}$.
- Multiplicative homomorphism:
  If $(c_1, d_1) = (g^{r_1}, m_1 \cdot h^{r_1})$, $(c_2, d_2) = (g^{r_2}, m_1 \cdot h^{r_2})$, then

$$\begin{aligned}(c_1 \cdot c_2, d_1 \cdot d_2) &= (g^{r_1} \cdot g^{r_2}, (m_1 \cdot h^{r_1}) \cdot (m_2 \cdot h^{r_2})) \\ &= (g^{r_1+r_2}, (m_1 \cdot m_2) h^{r_1+r_2}).\end{aligned}$$

Homomorphic encryption

- Many crypto systems have homomorphic properties: RSA, ElGamal, Benaloh, Paillier, but only provide additive or multiplicative homomorphism, not both.
- With addition and multiplication, can do arbitrary computations.
- First system that could do both: Boneh-Goh-Nissim 2005 many additions and one multiplication (uses pairings).
- Fully homomorphic encryption allows to do arbitrary computations on encrypted data without knowing the secret key,
- in particular it allows doing an arbitrary number of additions and multiplications.

Application scenario



operates on encrypted data: e.g. search, statistics, . . .

Server never sees data in the clear.

But does a fully homomorphic encryption scheme exist?
And if so, is it efficient?

Fully homomorphic encryption

Gentry proposed the first fully homomorphic encryption scheme in 2009 based on ideal lattices.

- The basis is a somewhat homomorphic encryption scheme that can evaluate low-degree polynomials on encrypted data.
- Ciphertexts are "noisy" and the noise grows slightly during addition and strongly during multiplication.
- If the SWHE scheme can evaluate its own decryption circuit, then a bootstrapping step can refresh ciphertexts by homomorphically decrypting using an encrypted secret key.
- Only works by "squashing" the decryption circuit.
- So far quite inefficient.

Fully homomorphic encryption

- Recently, many improvements, but still inefficient. Implementation (Gentry, Halevi 2011),
  - toy setting: encrypt a bit in 0.2s, recrypt in 6s, public key: 17MB
  - large setting: encrypt in 3min, recrypt in 31min, public key: 2.3GB
- New variants, mostly following Gentry's blueprint.
- Recent variants based on the LWE problem or RLWE problem.

- Applications might not need fully homomorphic encryption, somewhat homomorphic could be sufficient.
- This talk: somewhat homomorphic encryption scheme by Brakerski and Vaikuntanathan (Crypto 2011) based on RLWE.

The Learning with Errors (LWE) Problem

(Regev 2005)

Let $n \in \mathbb{N}$, $q \in \mathbb{Z}$, χ a probability distribution on $\mathbb{Z}$.

Distinguish the following distributions of pairs $(a_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$:

Uniform distribution

- Sample $(a_i, b_i) \in \mathbb{Z}_q^{n+1}$ uniformly at random.

LWE distribution

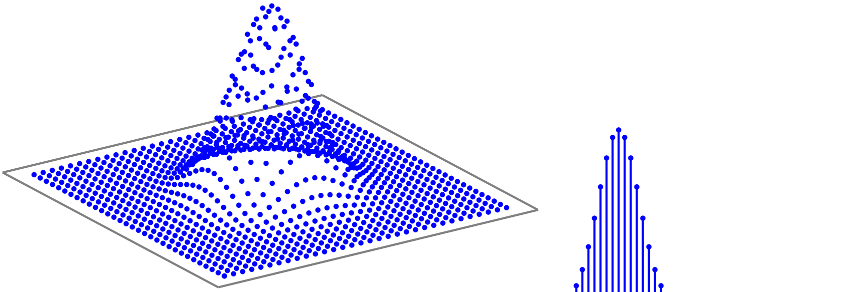
- Draw $s \in \mathbb{Z}_q^n$ uniformly at random.
- Sample $a_i \in \mathbb{Z}_q^n$ uniformly at random,
- sample $e_i \leftarrow \chi$, $\overline{e_i} \in \mathbb{Z}_q$,
- set $b_i = \langle a_i, s \rangle + \overline{e_i}$.

The b_i are noisy inner products of random a_i with a secret s.

The Learning with Errors (LWE) Problem

(Regev 2005)

- Regev gave a quantum reduction of certain approximate SVP to LWE, i.e. if one can solve LWE, then there's a quantum algorithm to solve certain approximate SVP.
- Peikert (2009) gave a reduction using classical algorithms
- Assumption: q prime, χ is a discrete Gaussian error distribution



The Ring Learning with Errors (RLWE) Problem

(Lyubashevsky, Peikert, Regev 2010)

Here: special case.

- Let $n = 2^k$,
$$f(x) = x^n + 1$$
($2n$-th cyclotomic polynomial).
- Define ring
$$R = \mathbb{Z}[x]/(f)$$
(ring of integers in $2n$-th cyclotomic number field).
- Let q be prime, define
$$R_q = R/qR \cong \mathbb{Z}_q[x]/(\overline{f}).$$
- Let χ be an error distribution on R.

The Ring Learning with Errors (RLWE) Problem

(Lyubashevsky, Peikert, Regev 2010)

Distinguish the following distributions of pairs $(a_i, b_i) \in R_q^2$:

Uniform distribution on R_q^2

- Sample $(a_i, b_i) \in R_q^2$ uniformly at random.

RLWE distribution

- Draw $s \in R_q$ uniformly at random.
- Sample $a_i \in R_q$ uniformly at random,
- sample $e_i \leftarrow \chi$, $\overline{e_i} \in R_q$,
- set $b_i = a_i \cdot s + \overline{e_i}$.

The b_i are noisy ring (number field) products of random a_i with a secret s.

Toy(!) example parameter setting

Let's take $k = 3$, i.e. $f = x^8 + 1$, $q = 97$.

- A typical (random) element in R_q looks like this:

$$a = 27x^7 - 11x^6 - 33x^5 + 41x^4 - 18x^3 - 5x^2 - 37x - 16.$$

- A small element sampled coefficient-wise from a narrow Gaussian, might look like this:

$$e = -2x^6 - 2x^3 + 2x^2 - x + 1.$$

- Addition in R_q:

$$\begin{aligned} a + e &= 27x^7 - 13x^6 - 33x^5 + 41x^4 - 20x^3 - 3x^2 - 38x - 15, \\ a + a &= -43x^7 - 22x^6 + 31x^5 - 15x^4 - 36x^3 - 10x^2 + 23x - 32. \end{aligned}$$

- Multiplication in R_q:

$$\begin{aligned} x \cdot a &= 27x^8 - 11x^7 - 33x^6 + 41x^5 - 18x^4 - 5x^3 - 37x^2 - 16x \\ &= -11x^7 - 33x^6 + 41x^5 - 18x^4 - 5x^3 - 37x^2 - 16x - 27. \end{aligned}$$

The Ring Learning with Errors (RLWE) Problem

(Lyubashevsky, Peikert, Regev 2010)

- Believed to be as hard as general LWE problem, i.e. would be solved with the same algorithms.
- Though there's a lot more structure!
- Recent results indicate RLWE problem easier than LWE, (Schneider 2011 claims in practice speedup is linear in n).
- But much more efficient.
- Smaller keys, very efficient arithmetic in R_q.

Can be used to build a fully homomorphic encryption scheme.

Slight modifications

- In both LWE and RLWE problems, it is okay to sample $s \leftarrow \chi$ (and not uniformly at random).
  - Sample until $(a_0, b_0 = a_0 s + e_0)$ with $a_0 \in R_q^*$ (invertible).
  - For every additional sample $(a, b = as + e)$ consider

$$\begin{aligned} (a', b') &= (-a_0^{-1} a, b + a' b_0) \\ &= (a', as + e + a'(a_0 s + e_0)) \\ &= (a', as + e - as + a' e_0) = (a', a' e_0 + e) \end{aligned}$$

  - If one can solve RLWE with small secret, then one can solve it with uniform secret.
- It is also okay to use small multiples of the error terms, i.e. samples $(a_i, b_i = a_i \cdot s + t e_i)$ are still indistinguishable from random. For example, take $t = 2$.

Somewhat homomorphic encryption

(Brakerski, Vaikuntanathan 2011)

$\mathbf{SH.Keygen}$

- Sample small $s \leftarrow \chi$. Set secret key $\mathrm{sk} = s$.

Sample RLWE instance:

- Sample $a_1 \leftarrow R_q$ unif. rand., small error $e \leftarrow \chi$.

Set public key

- $\mathrm{pk} = (a_0 = -(a_1 s + te), a_1)$.

In the example setting: $t = 2$

- $s = -x^7 - x^6 - x^5 + x^4 + x^3 + x^2 + x - 1$,
- $e = -2x^6 - 2x^3 + 2x^2 - x + 1$,
- $a_1 = 27x^7 - 11x^6 - 33x^5 + 41x^4 - 18x^3 - 5x^2 - 37x - 16$,
- $a_0 = 10x^7 - 25x^6 + 46x^5 - 37x^4 + 23x^3 + 27x^2 - 43x + 31$,
- $\mathrm{pk} = (10x^7 - 25x^6 + 46x^5 - 37x^4 + 23x^3 + 27x^2 - 43x + 31,$ $27x^7 - 11x^6 - 33x^5 + 41x^4 - 18x^3 - 5x^2 - 37x - 16)$.

Somewhat homomorphic encryption

(Brakerski, Vaikuntanathan 2011)

Message space:

$$R_t = \mathbb{Z}_t[x]/(x^n + 1),$$

t rel. prime to q, e.g. $t = 2$. Encode messages as elements in R_q with coefficients mod t.

- Can encode n bits at once.
- For example encode 01011001 as $m = x^6 + x^4 + x^3 + 1$.

$\mathbf{SH}.\mathbf{Enc}$

Given $\mathrm{pk} = (a_0, a_1)$ and a message $m \in R_q$,

- sample $u \leftarrow \chi$, and $g, h \leftarrow \chi$,

Set ciphertext

- ct $= (c_0, c_1) := (a_0 u + tg + m, a_1 u + th)$.

Somewhat homomorphic encryption

Example encryption

- Sample small elements

$$
\begin{aligned}
u &= -2x^6 + 3x^5 + 2x^3 - x, \\
g &= -x^6 - x^2 + 2x, \\
h &= -x^7 + x^5 + x^4 + x + 1.
\end{aligned}
$$

- From $\mathrm{pk} = (a_0, a_1)$ as above and $m = x^6 + x^4 + x^3 + 1$ compute

$$
\begin{aligned}
c_0 &= a_0 \cdot u + 2 \cdot g + m \\
&= 21x^7 + 2x^6 + 10x^5 + 6x^4 + 9x^3 + 3x^2 - 14x + 1 \\
c_1 &= a_1 \cdot u + 2 \cdot h \\
&= -44x^7 + 15x^6 - 43x^5 + 37x^4 + 37x^3 - 30x^2 - 22x + 42.
\end{aligned}
$$

- The ciphertext is

$$
\begin{aligned}
(c_0, c_1) = \; & (21x^7 + 2x^6 + 10x^5 + 6x^4 + 9x^3 + 3x^2 - 14x + 1, \\
& -44x^7 + 15x^6 - 43x^5 + 37x^4 + 37x^3 - 30x^2 - 22x + 42).
\end{aligned}
$$

Somewhat homomorphic encryption

(Brakerski, Vaikuntanathan 2011)

$\mathbf{SH.Dec}$

Given $\mathrm{sk} = s$ and a ciphertext ct $= (c_0, c_1)$,

- compute $\widetilde{m} = c_0 + c_1 s \in R_q$.

Output the message

- $\widetilde{m} \mod t$.

Correctness:

$$\begin{aligned} \widetilde{m} = c_0 + c_1 s &= (a_0 u + tg + m) + (a_1 u + th)s \\ &= -(a_1 s + te)u + tg + m + a_1 us + ths \\ &= m + t(g + hs - eu). \end{aligned}$$

Reduction modulo t gives back m as long as the error terms are not too large. Gives bound on standard deviation of the Gaussian.

Somewhat homomorphic encryption

Example decryption

- Use $\mathrm{sk} = s = -x^7 - x^6 - x^5 + x^4 + x^3 + x^2 + x - 1$ and ciphertext

$$
\begin{aligned}
(c_0, c_1) &= (21x^7 + 2x^6 + 10x^5 + 6x^4 + 9x^3 + 3x^2 - 14x + 1, \\
&\quad -44x^7 + 15x^6 - 43x^5 + 37x^4 + 37x^3 - 30x^2 - 22x + 42).
\end{aligned}
$$

- Compute

$$
\begin{aligned}
\tilde{m} &= c_0 + c_1 \cdot s \\
&= 24x^7 + 21x^6 + 4x^5 + 21x^4 + 15x^3 + 16x^2 - 28x - 21.
\end{aligned}
$$

- Reduce modulo $t = 2$ and get

$$x^6 + x^4 + x^3 + 1 = m.$$

Somewhat homomorphic encryption

(Brakerski, Vaikuntanathan 2011)

Homomorphic operations

$\mathbf{SH.Add}$

Given $\mathsf{ct} = (c_0, c_1)$ and $\mathsf{ct}' = (c'_0, c'_1)$, set the new ciphertext

- $\mathsf{ct}_{\mathsf{add}} = (c_0 + c'_0, c_1 + c'_1)$
  $= (a_0(u + u') + t(g + g') + (m + m'), a_1(u + u') + t(h + h'))$.

$\mathbf{SH.Mult}$

Given $\mathsf{ct} = (c_0, c_1)$ and $\mathsf{ct}' = (c'_0, c'_1)$,

- compute
  $(c_0 + c_1 X)(c'_0 + c'_1 X) = c_0 c'_0 + (c_0 c'_1 + c'_0 c_1) X + c_1 c'_1 X^2$
- $\mathsf{ct}_{\mathsf{mlt}} = (c_0 c'_0, c_0 c'_1 + c'_0 c_1, c_1 c'_1)$

Errors multiply!

$(m + t(g + hs - eu))(m' + t(g' + h's + eu')) = mm' + t(\dots)$

Somewhat homomorphic encryption

(Brakerski, Vaikuntanathan 2011)

- Homomorphic operations increase size of error terms.
- Homomorphic multiplication increases the size of the ciphertext.
- Homomorphic addition, multiplication, and decryption generalize to longer ciphertexts.

$\mathbf{SH.Dec}$

Given $\mathrm{sk} = s$ and a ciphertext $\mathsf{ct} = (c_0, c_1, \ldots, c_\delta)$,

- compute $\widetilde{m} = \sum_{i=0}^{\delta} c_i s^i \in R_q$.

Output the message

- $\widetilde{m} \pmod{t}$.

Relinearization

(Brakerski, Vaikuntanathan 2011)

There is a way to go from 3-element ciphertext $\mathsf{ct} = (c_0, c_1, c_2)$ back to a 2-element ciphertext.

- We have
$$c_2 s^2 + c_1 s + c_0 = t e_{\mathsf{mult}} + m m'$$
- Publish a "homomorphism key"
$$h_i = (a_i, b_i = -(a_i s + t e_i) + t^i s^2) \quad \text{for } i = 0, \ldots, \lceil \log_t q \rceil - 1$$
- Write c_2 in its base-t representation $c_2 = \sum c_{2,i} t^i$.

Relinearization

(Brakerski, Vaikuntanathan 2011)

- Replace ct by $(c_0^{\mathsf{relin}}, c_1^{\mathsf{relin}})$ with

$$c_1^{\mathsf{relin}} = c_1 + \sum_{i=0}^{\lceil \log_t q \rceil - 1} c_{2,i} a_i, \quad c_0^{\mathsf{relin}} = c_0 + \sum_{i=0}^{\lceil \log_t q \rceil - 1} c_{2,i} b_i$$

- Then

$$c_0^{\mathsf{relin}} + c_1^{\mathsf{relin}} s = c_0 + c_1 s + c_2 s^2 - t e_{\mathsf{relin}}$$

$$c_0^{\mathsf{relin}} + c_1^{\mathsf{relin}} s = t(e_{\mathsf{mult}} - e_{\mathsf{relin}}) + mm'$$

- Okay, ciphertext is smaller, but error has increased!
- Decryption still correct if final error $e_{\mathsf{mult}} - e_{\mathsf{relin}}$ is small enough.

Specific parameter choices

Choosing parameters to “guarantee” security and correctness.

Correctness:

- q must be large enough when compared to the size of the error terms and t.
- I.e. parameters are chosen s.t. the scheme can evaluate polynomials of a certain fixed degree D ($D-1$ multiplications and a bunch of additions).

Security:

- Against distinguishing attack with advantage 2^{-32} by Micciancio/Regev 2009.
- Adjust analysis of Lindner/Peikert 2011 to our setting.
- Still assume RLWE is no easier than LWE.

Specific parameters, key and ciphertext sizes

t	D	n	$\lceil \lg(q) \rceil$	$\lg(\mathrm{T})$	$l_{R_q}/10^3$	$(2 + \log_t q) \cdot l_{R_q}/10^3$
2	1	512	19	123	10	205
	2	1024	38	107	39	1557
	3	2048	64	134	132	8651
	5	4096	120	145	492	59966
	10	8192	264	117	2163	575276
1024	1	1024	30	164	31	154
	2	2048	58	164	119	927
	3	4096	95	215	390	4475
	5	8192	171	242	1401	26756
	10	16384	368	214	6030	233938

Message encoding

Homomorphic operations reflect operations in R_t.

- Want operations on integers.
- Encode an integer $m = (m_0, m_1, \ldots, m_l)_2$, $m_i \in \{0, 1\}$ as a polynomial of degree l with coefficients m_i. Get back m by evaluating at 2.
- $t = 2$ not useful for addition and multiplication since operations mod 2 are different from integer operations.
- Choose t large enough to allow for enough additions.
- Reduction modulo $x^n + 1$ screws up integer multiplication.
- Choose l small enough to allow a certain number of multiplications before reaching degree n.

Reference implementation

Implementation using the computer algebra system Magma

- Uses polynomial arithmetic in Magma,
- no specific optimization for multiplication, no DFT,
- no optimization for specific parameters (sizes),
- decryption for arbitrary length ciphertexts.

Big potential to improve efficiency

- Main cost is polynomial multiplication modulo $x^n + 1$ in R_q.

Timings

Intel Core 2 Duo @ 2.1 GHz

				S_χ	Enc	Dec		Mult	Mult
					prec.	deg 1	deg 2		degred
t	D	n	$\lceil \lg(q) \rceil$	ms	ms	ms	ms	ms	s
2	1	512	19	27	2	2	–	–	–
	2	1024	38	55	9	6	10	15	0.3
	3	2048	64	110	29	18	33	56	2.0
	5	4096	120	223	85	49	94	163	10.6
	10	8192	264	438	425	227	454	887	114.6
1024	1	1024	30	54	5	4	–	–	–
	2	2048	58	110	24	15	26	41	0.2
	3	4096	95	221	81	46	88	154	1.0
	5	8192	171	440	275	148	288	526	5.3
	10	16384	368	868	1260	664	1300	1593	48.2

- Compute the ciphertext of the sum of 100 numbers of size 128 bits from the single ciphertexts (for mean computation): < 20ms
- Ciphertexts for the sum and sum of squares of 100 such numbers (for mean and variance): < 6s

Questions?

- Regev: On Lattices, Learning with Errors, Random Linear Codes, and Cryptography, STOC 2005, J. ACM 2009.
- Lyubashevsky, Peikert, Regev: On Ideal Lattices and Learning with Errors Over Rings, Eurocrypt 2010.
- Brakerski, Vaikuntanathan: Fully Homomorphic Encryption from Ring LWE and Security for Key Dependent Messages, Crypto 2011.
- Lauter, N., Vaikuntanathan: Can homomorphic encryption be practical?, ACM CCSW 2011.
- `michael@cryptojedi.org`