Homomorphic Cryptosystems

Edlyn Teske-Wilson

University of Waterloo

Ottawa, 27 June 2011

Focus: Fully homomorphic encryption

- C. Gentry: Fully homomorphic encryption using ideal lattices. *STOC 2009*.
- N.P. Smart, F. Vercauteren: Fully homomorphic encryption with relatively small key and ciphertext sizes. *PKC 2010.*
- C. Gentry, S. Halevi: Implementing Gentry's fully-homomorphic encryption scheme. *Eurocrypt 2011*.
- M. van Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan: Fully homomorphic encryption over the integers. *Eurocrypt 2010*.

The homomorphic property

Basic RSA:
Homomorphic with respect to multiplication....

$$E_{\mathsf{RSA}}(m) = m^e \pmod{n} \qquad (n = pq)$$

and

$$E_{\mathsf{RSA}}(ab) \equiv (ab)^e = a^e b^e \equiv E_{\mathsf{RSA}}(a) \cdot E_{\mathsf{RSA}}(b) \pmod{n},$$

...but not with respect to addition:

$$(a + b)^e \not\equiv a^e + b^e \pmod{n}.$$

Fully homomorphic encryption

A fully homomorphic encryption scheme is
a scheme $E = (\text{KeyGen}_E, \text{Encrypt}_E, \text{Decrypt}_E)$
with an additional efficient algorithm Evaluate_E
that,
for any valid public key pk,
and for **any** circuit C
(not just a circuit consisting of multiplication gates
as in RSA),
and any ciphertexts

$$c_i \leftarrow \text{Encrypt}_E(pk, m_i),$$

outputs

$$c \leftarrow \text{Evaluate}_E(pk, C, c_1, \ldots, c_t),$$

a **valid** encryption of $C(m_1, \ldots, m_t)$ under pk.

Valid, i.e.

$$\text{Decrypt}_E(sk, c) = C(m_1, \ldots, m_t).$$

Note: this definition allows the trivial solution......
so we also require *circuit privacy* and *compactness*.

Why homomorphic cryptosystems?

Encryption schemes:

- Searching an encrypted database on a remote server.
- Compute on encrypted data.
- Spam filtering of encrypted emails.
- Outsource any kind of private computation.

Signature schemes:

- Signatures for network coding
  (only linear functions needed).
- Computing on signed data.

(Necessary to shape new security notions.)

Selected references: Homomorphic signatures:

- D. Charles, K. Jain, and K. Lauter. Signatures for network coding. *IJICOT, 2006*.

- D. Boneh, D. Freeman, J. Katz and B. Waters: Signing a linear subspace: Signature schemes for network coding. *PKC 2009*.

- R. Gennaro, D. Katz, H. Krawczyk, T. Rabin: Secure network coding over the integers. *PKC 2010*.

- D. Boneh, D. Mandell Freeman: Homomorphic signatures for polynomial functions. *Eurocrypt 2011*.

Gentry's Breakthrough (2009):

Use an *ideal lattice*, J.

That is, J is a lattice that is also an ideal.

Easy to construct a *somewhat* homomorphic system.

"somewhat" ...because of "noise":

Plaintext: $m \in \{0, 1\}$.
Ciphertext:
$c = j + 2r + m$ where $j \in J$ and r small.
Decrypt: retrieve $e = 2r + m$.
This works if e is small enough.
Then find $m = e \bmod 2$.

Now, when adding or multiplying ciphertexts, the noise e increases.....
....until it becomes too large and decryption is not correct.

Bootstrapping

Noise increases while computing on encrypted data.

So, need to “refresh” the ciphertext every once in a while.

This is easy if secret key is available:
decrypt, then encrypt again.

Without secret key: “bootstrapping”.

Bootstrapping

Noise increases while computing on encrypted data.

So, need to “refresh” the ciphertext every once in a while.

This is easy if secret key is available:
decrypt, then encrypt again.

Without secret key: “bootstrapping”.

Bootstrap?

to bootstrap:
to better oneself by one’s own unaided efforts.

bootstrapping:
a series of selfsustaining processes that proceed without external help.

Bootstrapping (cont.)

Assume our somewhat homomorphic system can handle circuits up to a certain depth, say D.

If the so-called “augmented” decrypt circuit has depth $\leq D$, then the system is “bootstrappable”.

If we can bootstrap, then we can refresh ciphertexts, via *recryption*.

Recryption – refreshing ciphertexts:
(simplified)

The idea:
Take two public-secret key pairs

$$(sk_1, pk_1) \quad \text{and} \quad (sk_2, pk_2).$$

That is:

$$\text{Decrypt}_E(sk_1, \text{Encrypt}_E(pk_1, m)) = m$$

for any message m.
Ditto for the second pair.

Assume the scheme E is homomorphic with respect to the decryption circuit.

Take an encryption of sk_1 under the public key pk_2:

$$\mathsf{Encrypt}_E(pk_2, sk_1).$$

Also, take an encryption of the initial ciphertext under the public key pk_2:

$$\mathsf{Encrypt}_E(pk_2, \mathsf{Encrypt}_E(pk_1, m)).$$

Consider

$$\mathsf{Dec}_E(\mathsf{Enc}_E(pk_2, sk_1), \mathsf{Enc}_E(pk_2, \mathsf{Enc}_E(pk_1, m)).$$

$$\mathsf{Dec}_E(\mathsf{Enc}_E(pk_2, sk_1), \mathsf{Enc}_E(pk_2, \mathsf{Enc}_E(pk_1, m))$$

$$= \mathsf{Encrypt}_E(pk_2, m).$$

(Well, you need to do this bit-wise, really....)

So, one can remove the inner encryption......
.....creating a newly encrypted (under pk_2) ciphertext.

Now assume the scheme E can homomorphically evaluate

$$\mathsf{Decrypt}_E(sk, c_1) + \mathsf{Decrypt}_E(sk, c_2)$$

and

$$\mathsf{Decrypt}_E(sk, c_2) \cdot \mathsf{Decrypt}_E(sk, c_2).$$

Then we say E is *bootstrappable*.

Gentry (2009):

E bootstrappable

$\Rightarrow$ fully homomorphic encryption scheme E.

The new scheme inherits semantic security against chosen plaintext attacks from E.

Back to ideal lattices:

Why they are good:

- **Very low circuit complexity** of decrypt algorithm.
  (Compare with RSA, ElGamal).

- Natural Add/Mult. operations. (Think of ideals in polynomial rings.)

- Security can be based on standard problems over ideal lattices, that **seem** to be as hard as standard well-studied problems over general lattices.

But, problem:

- Decryption circuit is not shallow enough!
  I.e., its depth is larger than what Evaluate$_E$ function can handle.

The problem:

Evaluate$_E$ function can handle a certain set C of circuits.

But C does not contain (augmented) decryption circuit.

Solution 1:

- Modify the scheme to enlarge C.
- But this possibly complexifies Decrypt$_E$.

Solution 2: Squash the decryption circuit:

- While encrypting, include extra data to help decrypter for decryption
  (think of server-aided cryptography...).
- Extra data = secret-key info, presented as subset sum problem.

Some math, at last

Smart-Vercauteren (PKC 2010):

Let $n = 2^q$, $f(x) = x^n + 1$.

$R = \mathbb{Z}[x]/(f(x)) \cong \mathbb{Z}^n$.

Consider the principal ideal $J = (v)$ generated by $v \in R$.

Note: Coefficient vectors associated to the elements of J form a lattice with *rotation basis*

$$\vec{v_i} := \{\vec{v} \times x^i \bmod f(x) \, : \, i \in [0, n-1]\} \ ,$$

We call $J = (\vec{v})$ an *ideal lattice*.

Smart-Vercauteren: no lattice talk!

Let K be the number field $\mathbb{Q}(\Theta)$,
where Θ is a root of $f(x) = x^n + 1$.

Take $v(x) \in \mathbb{Z}[x]$, $\deg(v) = n - 1$,
$v(x) \equiv 1 \bmod 2$, $\|v\|_\infty \leq \eta$ such that

$$p := \text{Resultant}(v(x), f(x))$$

is prime.

$J = (v(\Theta))$ is a degree-one prime ideal in $\mathbb{Z}[\Theta]$.

Let r be the common root of $v(x)$, $f(x)$ mod p.

Two-element representation of J: $(p, \Theta - r)$.
(p, r) **is the public key.**

Let $z(x) = \sum_{i=0}^{n} z_i x^i$ be the scaled inverse of $v(x)$:

$$z(x)v(x) = p \pmod{f(x)} .$$

Let $w = z_0 \pmod{2p}$.
(p, w) **is the secret key.**

Encryption:

Let $m \in \{0,1\}$.
Take $u(x) \in_R \mathbb{Z}[x]$, deg$(u) = n-1$, $\|u\|_\infty \leq \mu/2$.

$$c \leftarrow \text{Encrypt}((p,r),m) = (m + 2u(r)) \pmod{p}.$$

(This is reduction of $m + 2u(\Theta)$ modulo J.)

Decryption:

$$\text{Decrypt}((p,w),c) = c - \lfloor c \cdot w/p \rceil \pmod{2}.$$

Why this works:

$$m + 2u(\Theta) - c \in J.$$

Let $q(\Theta) \in \mathbb{Z}[\Theta]$ such that

$$m + 2u(\Theta) - c = q(\Theta)v(\Theta).$$

So $m - c = q_0 \pmod 2$.
(Remember, $v(x) \equiv 1$ mod 2.)

Recall, $z(x)v(x) = p \pmod{f(x)}$ and $f(\Theta) = 0$,
so devide by $v(\Theta)$:

$$\frac{(m + 2u(\Theta))z(\Theta)}{p} - \frac{cz(\Theta)}{p} = q(\Theta),$$

$$-\frac{(m + 2u(\Theta))z(\Theta)}{p} + q(\Theta) = -\frac{cz(\Theta)}{p}.$$

Thus, $q_0 = -\lfloor c \cdot z_0/p \rceil$, IF

$$\left\| -\frac{(m + 2u(x))z(x)}{p} \right\|_\infty < \frac{1}{2}.$$

Putting things together:

$$\begin{aligned} m &= c + q_0 \pmod{2} \\ q_0 &= -\lfloor c \cdot z_0 / p \rceil \\ w &= z_0 \pmod{2p} \end{aligned}$$

So $m = c - \lfloor c \cdot z_0 / p \rceil \pmod{2}$.

Summary:

$J = (v(\Theta))$ where $v(x) \in \mathbb{Z}[x]$, $\|v\|_\infty \leq \eta$.

Encrypt: $c \leftarrow (m + 2u(r)) \pmod{p}$
$(\|u\|_\infty \leq \mu/2)$.

Decrypt: $m \leftarrow c - \lfloor c \cdot w/p \rceil \pmod{2}$

This works if

$$\left\| -\frac{(m + 2u(x))z(x)}{p} \right\|_\infty < \frac{1}{2}.$$

The latter holds if (after some calculation.....):

$$\|m + 2u(x)\|_\infty < \frac{\eta}{2\sqrt{n}} \ .$$

So to decrypt correctly, one needs:

$$\|m + 2u(x)\|_\infty < \frac{\eta}{2\sqrt{n}} \ .$$

Now consider computing on encrypted data:

Add:

$c_1 + c_2 = (m_1 + m_2) + 2(u_1(r) + u_2(r)) \pmod{p}.$

Multiply:

$c_1 \cdot c_2 = (m_1 \cdot m_2) + 2(m_1 u_2(r) + m_2 u_1(r) + 4u_1(r)u_2(r)).$

$\pmod{p}.$

"Noise" increases.......:

Initially, $\|m + 2u(x)\|_\infty \leq \mu + 1.$

Do some calculations.....and obtain:

After executing a circuit with multiplicative depth D, obtain ciphertext $c' = m' + 2u'(r)$ with

$$\|c'(x)\|_\infty \leq T \quad \text{where} \quad T \approx (n\mu)^{2D}.$$

Recall: to decrypt correctly, one needs:

$$\|m + 2u(x)\|_\infty < \frac{\eta}{2\sqrt{n}} \ .$$

More calculation:
The output of a circuit of depth D can be correctly decrypted if

$$D \log 2 < \log\log\left(\frac{\eta}{2\sqrt{n}}\right) - \log\log(n\mu).$$

With $n = 2^{11}$, $\eta = 2^{\sqrt{n}}$, $\mu = 2$
this allows for $D = 1.7$.

Need p with 92681 bits for that (for security reasons). This is the largest circuit depth that could be achieved...... not enough for bootstrapping to work, even with squashed decryption circuit.

So SV2010 scheme cannot be implemented.

Gentry-Halevi (2011):
generalize Smart-Vercauteren constructions:

- Switch back to lattice presentation.
- Choose $\vec{v}$ such that $\det \vec{v}$ odd and square-free.
- Algorithm to compute z_0 only, not all of $z(x)$.

Security – underlying problems

- Small principal ideal problem (SV 2010).
  To recover the private key.

- Bounded distance decoding problem / Closest vector problem (G 2009, SV 2010).
  To recover the message from a given ciphertext.

- Polynomial coset problem (SV 2010) / Ideal coset problem (G 2009).
  To break semantic security.

- Sparse subset sum problem.
  To recover secret key from additional data due to squashing the decryption circuit.

- Approximate integer GCD (DGHV 2010).

More work has been done.......
(all in 2011)

Groth, Smart-Vercauteren, Gu ($\times 5$), Boneh-Segev-Waters, Armknecht-Augot-Perret-Sadeghi, Gentry, Gentry-Halevi, ...

....and more work is being done

- Ideal lattices easier than general lattices?
- Improve efficiency of key generation.
- Improve efficiency of encrypt and decrypt, and reducing key length and ciphertext size.
- Improve squashing mechanism/get rid of squashing mechanism.
- Get rid of bootstrapping.
- Find new applications.

Thank you!

Security

- IND-CCA2 security (i.e., indistinguishability of ciphertexts under adaptive chosen ciphertext attack):
  impossible, due to the malleability of ciphertexts.

- IND-CCA1 security (non-adaptive): open problem.

- Indistinguishability against chosen plaintext attacks: Yes.
  - security of the somewhat homomorphic scheme.
  - security after addition of the secret key hint to the public key.