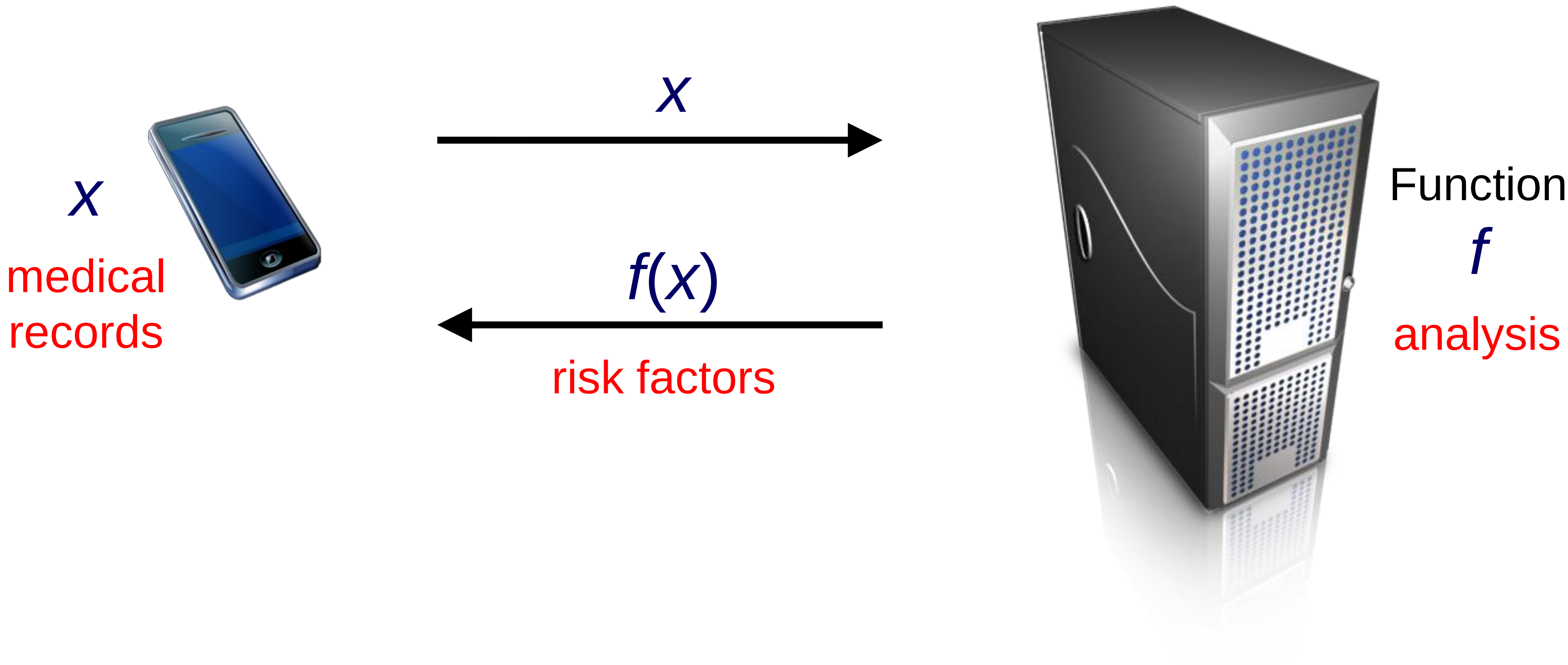
Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages

Zvika Brakerski
(Weizmann)

Vinod Vaikuntanathan
(University of Toronto)

CRYPTO 2011

Outsourcing Computation



Want Privacy!

Outsourcing Computation – Privately

Knows nothing of x.

Enc(x) →

x

← y

Function f

Dec(y)=f(x)

WANTED

Eval: f, Enc(x) → Enc(f(x))

homomorphic evaluation

Fully Homomorphic Encryption (FHE) [RAD78]

sk

x

Enc(x)

y = Eval$_{pk}$(f, Enc(x))

pk

Function f

Correctness guarantee:

Dec$_{sk}$(y) = f(x)

Privacy guarantee (semantic security [GM82]):

Enc(x) $\cong$ Enc(0)

"Fully" = Evaluate all (efficient) f

Evaluating binary $+, \times$ is sufficient.

Gentry's Breakthrough [G09,G10]
First Candidate FHE

Bootstrapping Theorem [G09]**:**

d-HE + dec. depth < *d* + circular security ⇒ **FHE**

Eval for any depth *d* circuit (aka “somewhat” HE)

=key dependent message security

Adversary sees $Enc(sk)$.

(more generally: $Enc(f(sk))$)

Gentry's construction:

d-HE with dec. depth **>** *d*

Ideal lattice assumption.

“Squash” to dec.

Novel use of ideal lattices! Previous works (e.g. [NTRU, MR04, LM06, M07]) used for efficiency, here used for functionality.

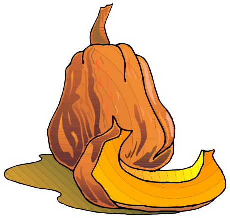
Sparse Subset-Sum assumption.

+

Explicit circular security assumption

Since Gentry

- Another candidate [vDGHV10]:

d-HE with dec. depth **>** *d*  approx. GCD assumption.	+	“Squash” to dec. depth < *d*  Sparse Subset-Sum assumption.	+	Explicit circular security assumption 

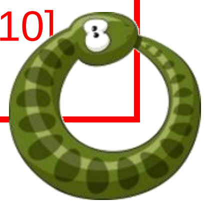
- Efficiency improvements of Gentry's scheme [SV10, SS10, GH11].

Our Scheme

Simple

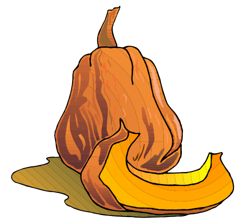
d-HE with dec. depth **>** *d*

Ring-LWE [LPR10] assumption.



\+

"Squash" to dec. depth < *d*



Sparse Subset-Sum assumption.

\+

Explicit circular security assumption

- First circular secure "somewhat" HE.
  - Circular security extends to polynomials of key (a la [MTY11]).
  - Caveat: circular scheme is not bootstrappable.
- Simple construction! Simple key generation.
  - Combine the "two callings" of ideal lattices: efficiency and functionality.

People are implementing!

Ring-LWE [LPR10]
(simplified)

Ring of polynomials:

$$R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$$

Degree $(n-1)$ polynomials with coefficients in $\mathbb{Z}_q$ (q large odd prime).

***RLWE*$_{n,q}$ assumption:** For random s

any coefficient

$$\{\, (a_i, b_i = a_i s + \ 2\, e_i) \,\} \approx \{\, (a_i, u_i) \,\}$$

For uniform a_i, u_i and for “small” e_i.

Distinguish $RLWE_{n,q}$ $\Rightarrow_{\text{quant.}}$ short vectors in ideal lattice

[LPR10]

Toy Example: "Ring-LWOE"

Ring "learning without errors" on ring R:

$$\{\,(a_i, b_i = a_i s)\,\} \approx \{\,(a_i, u_i)\,\}$$

(ob

Circular security:

$$\begin{aligned} Enc_s(s) &= (a, -as + s) \\ &= (a, -(a-1)s) \\ &= \big((a'+1), -a's\big) \\ &= Enc_s(0) + (1,0) \end{aligned}$$

Ring-LWOE based (symmetric) er

- **Key generation:** uniformly s
- **Encrypt $m \in \{0,1\}$:** $c = (a,\ \ b = -as + m)$.
- **Decrypt $c = (a, b)$:** $m = (as + b) \quad (mod\ 2)$.

modular operation needed for actual scheme

Toy Example: Homomorphic Add.

$$c = (a, b) \qquad + \qquad c' = (a', b')$$

s.t. $as + b = m$ s.t. $a's + b' = m'$

$$\Rightarrow \quad c_{add} = (a + a', b + b')$$

Correctness:

$$\begin{array}{r} as + b = m \\ a's + b' = m' \end{array} \quad +$$

$$\overline{(a + a')s + (b + b') = m + m'}$$

Toy Example: Homomorphic Mult.

$c = (a, b)$ s.t. $as + b = m$ $\times$ $c' = (a', b')$ s.t. $a's + b' = m'$

$$\Rightarrow \quad c_{mult} = (h_2, h_1, h_0)$$

$$\frac{\begin{array}{c} as + b = m \\ a's + b' = m' \end{array} \quad \times}{\begin{array}{c} (as + b) \cdot (a's + b') = m \cdot m' \\ h_2 s^2 + h_1 s + h_0 = m \cdot m' \end{array}}$$

$$\begin{aligned} \boldsymbol{Dec_s}(h_2, h_1, h_0) &= h_2 s^2 + h_1 s + h_0 \pmod 2 \\ &= m \cdot m' \pmod 2 \end{aligned}$$

The Actual Scheme

Just add noise…

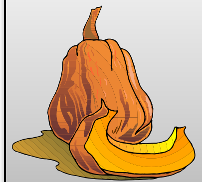
- **Key generation:** uniformly sample $\mathrm{sk} = s$.
- **Encrypt** $m \in \{0,1\}$**:** $c = (a,\ \ b = -as + 2e + m)$.
- **Decrypt** $c = (h_d, \dots, h_1, h_0)$**:** $m = \sum h_i s^i \pmod 2$

$$= \langle \vec{h}, \vec{s} \rangle \pmod 2.$$

After hom. eval. of deg. d function

(where $\vec{s} = (s^d, \dots, s, 1)$.)

Noise grows exponentially with d $\Rightarrow$ $d < \log q \approx n^{\epsilon}$.



Squashing: Represent $\vec{s}$ as sparse subset sum a la Gentry.

Follow-Up Works

- FHE from standard LWE without squashing [BV11b].
  - Techniques apply for RLWE as well.
- Better noise management and further efficiency improvements [BGV11].
- Implementation of (“somewhat homomorphic”) scheme [LNV11].

Conclusion

- We showed circular secure somewhat homomorphic encryption.
  - Q: Circular secure *bootstrappable* encryption?

- Our scheme is basis for implementations (combined with follow-up) – hope for more efficient schemes.

Thank you