Ring − LWE

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

Homomorphisms

Dfn: R, S rings

$f: R \to S$ is a homomorphism if

$f(x+y) = f(x)+f(y)$

$f(xy) = f(x)f(y)$.

Brakerski + Vaikuntanathan

Key gen

Alice generates a small poly $s(x)$.

Encryption

1) A gen random $a(x)$ random small $e(x)$

2) message string of bits. Encode as poly $m(x)$ w/ coeffs 0 or 1

01001 1

$\to x^4+x+1$

3) $c_1 = -a(x)$

$c_0 = a(x)s(x)+2e(x)+m(x)$

4) transmits (c_0, c_1).

Decryption

1) Bob gets (c_0, c_1)

2) computes

$c_0(x)+c_1(x)s(x)$

3) Reduces mod 2 gets $m(x)$.

$c_0(x)+c_1(x)s(x)$

$= a(x)s(x)+2e(x)+m(x)-a(x)s(x)$

$= 2e(x)+m(x)$

$\equiv m(x) \mod(2)$.

b/c $m(x)$ has coeffs in $\{0,1\}$.

Need e small

$q = 17$

$m = x^2 + x + 1$

$e = 8x^2 + 9x + 2$

$2e + m = 17x^2 + 19x + 5$ *

$\equiv 2x + 5$

$\equiv 0x^2 + 0x + 1$

$\to (0, 0, 1)$

mod 17 and mod 2 don't commute, but ok if e is small

$n = 4$

$q = 17$

encrypt $(1\,0\,1\,0)$

$\to m(x) = x^3 + x$

Key $s(x) = x^3 - x^2 + 2x$

Alice

$a(x) = 3x^2 + 7x - 5$

$e(x) = 2x^3 + x + 1$

$c_1 = -a(x) = -3x^2 - 7x + 5$

$c_0 = a(x)\,s(x) + 2e(x) + m(x)$

$= (3x^2 + 7x - 5)(x^3 - x^2 + 2x)$

$+ 4x^3 + 2x + 2 + x^3 + x$

$= 3x^5 + 4x^4 - x^3 + 19x^2 - 7x + 2$

$= \quad -x^3 + 2x^2 - 10x - 2$

$= -x^3 + 2x^2 + 7x - 2.$

Transmits

$(-x^3 + 2x^2 + 7x - 2, -3x^2 - 7x + 5)$

$3x^5 - 7x = -3x - 7x = -10x = 7x$

Bob:

$c_0 + c_1 s$

$= -x^3 + 2x^2 + 7x - 2$

$+ (-3x^2 - 7x + 5)(x^3 - x^2 + 2x)$

$= -3x^5 - 4x^4 + 5x^3 - 17x^2 + 17x - 2$

$= 5x^3 + 3x + 2$

$\equiv x^3 + x \mod 2.$

Public Key scheme

Keygen

1) Alice gens
   random a_0
   random small s, e_0
2) computes
   $b_0 = a_0 s + 2e_0$
   public key = (a_0, b_0)

Encryption

1) Bob gens small
   v, e_1, e_2
2) computes
   $a_1 = a_0 v + 2e_1$
   $b_1 = b_0 v + 2e_2$
3) $c_0 = b_1 + m$
   $c_1 = -a_1$
4) Transmits
   (c_0, c_1)

Decryption

1) Alice gets
   (c_0, c_1)
2) $M = c_0 + s c_1$
3) $m = M \bmod 2$.

$$
\begin{aligned}
M &= c_0 + s c_1 \\
&= b_1 + m + s c_1 \quad (*) \\
&= b_0 v + 2e_2 + m - s a_1 \\
&= a_0 s v + 2e_0 v + 2e_2 + m - s a_0 v - 2 s e_1 \\
&= m + 2(e_0 v + e_2 - e_1 s) \\
&\equiv m \bmod 2.
\end{aligned}
$$

Somewhat homomorphic encryption

Prop: $d(e(m) + e(m')) = m + m'$

Pf/ $m \to (c_0, c_1)$

$m' \to (c_0', c_1')$

$(c_0, c_1) + (c_0', c_1') = (c_0 + c_0', c_1 + c_1')$

$= (as + 2e + m + a's + 2e' + m', -a - a')$

$c_0 + c_0' + (c_1 + c_1')s$

$= as + 2e + m + a's + 2e' + m' + (-a - a')s$

$= 2e + m + 2e' + m'$

$\equiv m + m' \mod 2.$

$d(e(m) + e(m')) = m + m'.$

$c_0 c_0' = (as + 2e + m)(a's + 2e' + m')$

$= mm' + asm' + a'sm + aa's^2$

$+ 2(ea's + 2ee' + em' + e'as + e'm)$

New algorithm

a ciphertext

$$\vec{c} = (c_0, c_1, \ldots, c_d) \in R_q^{d+1}$$

Add ciphertexts pointwise

$$\vec{c} = (c_0, c_1, c_2)$$

$$\vec{c}' = (c_0', c_1', c_2')$$

$$\vec{c} + \vec{c}' = (c_0 + c_0', c_1 + c_1', c_2 + c_2').$$

OR

$$\vec{c} = (c_0, c_1, c_2)$$

$$\vec{c}' = (c_0', c_1', c_2', c_3')$$

$$\vec{c} + \vec{c}' = (c_0 + c_0', c_1 + c_1', c_2 + c_2', 0 + c_3')$$

If I want to multiply

introduce symbol v

$$\vec{c} = \sum_{i=0}^{d} c_i v^i = c_0 + c_1 v + c_2 v^2 + \cdots + c_d v^d$$

polynomial in $R_q[v]$

e.g. $(x^2+1) + (x^3+3x-3)v + (x^2+4x+2)v^2$

$$\vec{c} \times \vec{c}' = (\hat{c}_0, \hat{c}_1, \ldots, \hat{c}_{d+d'})$$

$$\left(\sum_{i=0}^{d} c_i v^i\right)\left(\sum_{i=0}^{d'} c_i' v^i\right) = \sum_{i=0}^{d+d'} \hat{c}_i v^i.$$

Ex1 Alice has encrypted

$S(x) = x^3 - x^2 + 2x$

$m = x^3 + x$

$e(m) = (-x^3 + 2x^2 + 7x - 2 - 3x^2 - 7x + 5)$

$m' = x^2 + x$

$a' = 4x^3 - 5x^2 + 2$

$e' = 1$

$C_1' = -a' = -4x^3 + 5x^2 - 2x$

$C_0' = a' S + 2e' + m'$

$= -8x^3 - 5x^2 - 3x + 6$

$\vec{C} = (-x^3 + 2x^2 + 7x - 2) + (-3x^2 - 7x + 5) v$

$\vec{C}' = (-8x^3 - 5x^2 - 3x + 6) + (-4x^3 + 5x^2 - 2x) v$

$\vec{C}\vec{C}' = (-x^3 + 2x^2 + 7x - 2)(-8x^3 - 5x^2 - 3x + 6)$

$+ (-x^3 + 2x^2 + 7x - 2)(-4x^3 + 5x^2 - 2x) v$

$+ (-3x^2 - 7x + 5)(-8x^3 - 5x^2 - 3x + 6) v$

$+ (-3x^2 - 7x + 5)(-4x^3 + 5x^2 - 2x) v^2$

$= (8x + 10x^2 + 3x^3)$

$+ (15 + 3x + 11x^2 + 15x^3) v$

$+ (11 + 2x + 14x^2 + 13x^3) v^2.$

To decrypt:

Know s

can compute s^i

if I get ciphertext

$$\vec{c} = (c_0, c_1, \ldots, c_D) \in R_q^{D+1}$$

$$\sim c_0 + c_1 v + c_2 v^2 + \cdots + c_D v^D$$

compute

$$\vec{s} = (1, s, s^2, \ldots, s^D) \in R_q^{D+1}$$

$$\langle \vec{c} | \vec{s} \rangle = \sum_{i=0}^{D} c_i s^i$$

$$= c_0 + c_1 s + c_2 s^2 + \cdots + c_n s^n$$

$$\equiv m \mod 2.$$

As long as errors are small enough