

The BFV Scheme

Lecture 10

SEC 595 — Encrypted Computing

- 1 Spaces and Parameters
- 2 Encoding
- 3 Key Generation
- 4 Encryption and Decryption
- 5 EvalAdd
- 6 EvalMult
- 7 Relinearization
- 8 A Worked Instance
- 9 Where the Message Sits

Primitive	Returns
ParamGen(λ)	(n, t, q) and the distributions
KeyGen	SK (kept), PK and EK (published)
Encrypt(PK, M)	$ct \in R_q \times R_q$
Decrypt(SK, ct)	$M \in R_t$
EvalAdd($ct^{(1)}, ct^{(2)}$)	encrypts $M^{(1)} + M^{(2)}$
EvalMult($ct^{(1)}, ct^{(2)}$)	encrypts $M^{(1)} M^{(2)}$
Relinearize(EK, ct^*)	a two-element ct , same M

BFV is **levelled**: it evaluates circuits up to a multiplicative depth fixed when (n, t, q) is chosen.

Fan–Vercauteren (2012).

plaintexts $R_t = \mathbb{Z}_t[X]/(X^n + 1)$, ciphertexts $R_q \times R_q$.

- n : ring dimension, a power of two.
- t : plaintext modulus; q : ciphertext modulus; $q \gg t$.
- Coefficients in the centered range; $[\cdot]_q$ reduces into it (Lecture 1).
- $\|v\|$ = largest absolute coefficient; $\|ab\| \leq n\|a\|\|b\|$ (Lecture 3).

$|R_t| = t^n$, $|R_q \times R_q| = q^{2n}$: at $n = 4$, $t = 4$, $q = 2^{14}$, 256 plaintexts against 2^{112} ciphertexts.

Example — plaintexts at $(n, t) = (4, 5)$

- a. Write down the centered coefficient range for $t = 5$.
- b. Which of these are elements of R_5 ?

$$1 + 2X + X^2 - X^3, \quad -1 - 2X - X^2 + 2X^3, \quad 1 + X + X^2 + X^3$$

- c. Count $|R_5|$.

The three distributions

Symbol	What it is	Used for
R_2	ternary, coefficients in $\{-1, 0, 1\}$	SK and u
$\mathcal{X}$	discrete Gaussian (μ, σ, β)	e, e_1, e_2
R_q	uniform over R_q	the mask a

Standard: $\mu = 0$, $\sigma = 8/\sqrt{2\pi} \approx 3.2$, tail-cut $\beta = \lfloor 6\sigma \rfloor = 19$ (Lecture 6).

Because of the tail-cut, $\|e\| \leq \beta$ is treated as a hard bound: noise analysis becomes arithmetic.

Choosing (n, t, q)

λ is a work parameter: $\lambda = 128$ means the best known attack costs about 2^{128} operations.

Rule of thumb: *minimise all three*, subject to the computation fitting.

- larger n : costs time and size, buys security;
- larger q : buys depth, *costs* security at fixed n ;
- larger t : buys plaintext room, shrinks the budget $q/2t$ and inflates the growth factor $2tn^2$.

Encoding maps data into R_t . No keys, no randomness.

Naive encoding

$M = m + 0 \cdot X + \dots + 0 \cdot X^{n-1}$; decode by reading the constant term.

- Wastes $n - 1$ coefficients.
- One coefficient holds every intermediate value, so t must exceed the largest — and a large t costs budget.

Base- k encoding

Write $m = (a_{n-1} \cdots a_1 a_0)_k$ and set $M = a_{n-1}X^{n-1} + \cdots + a_1X + a_0$.
Decode by evaluating: $m = M(k)$.

Digits are read as non-negative representatives in $\{0, \dots, k-1\}$.

Coefficients start at $k-1$, not m — but multiplication raises degrees, and degree wraps at n .

Compatibility

At every intermediate result of the circuit:

- Ⓐ no coefficient reaches t in absolute value;
- Ⓑ the degree stays below n .

If (a) fails, the value is corrupted modulo t .

If (b) fails, $X^n \equiv -1$ folds high terms onto low ones *with a sign flip*.

Boolean circuits over $\mathbb{Z}_2$ tolerate (a) by design.

Example — encoding by hand

- a. Encode $m = 13$ in base 2 and in base 3 at $n = 4$; decode each.
- b. Which base leaves more room before (a) fails? Before (b) fails?
- c. Encode $m = 21$ in base 2 at $n = 4$.

Key generation

Sample $SK \leftarrow R_2$, $a \leftarrow R_q$, $e \leftarrow \mathcal{X}$. Set

$$PK_0 = [- (a \cdot SK + e)]_q, \quad PK_1 = a.$$

$$PK_0 + PK_1 \cdot SK = -e.$$

The public key, evaluated at the secret, is small.

(PK_0, PK_1) is one RLWE sample: distinguishing it from uniform is decision RLWE (Lecture 6).

The scaling factor

$$\Delta = \lfloor q/t \rfloor, \quad q = \Delta t + [q]_t, \quad 0 \leq [q]_t < t.$$

ΔM puts the message in the top $\log_2 t$ bits; the bottom $\log_2(q/t)$ bits are room for noise (Lecture 1).

Decryption scales by t/q and **rounds**.

The residue $[q]_t$ vanishes exactly when $t \mid q$.

Encryption

Sample $u \leftarrow R_2$, $e_1, e_2 \leftarrow \mathcal{X}$. Return

$$c_0 = [\text{PK}_0 \cdot u + e_1 + \Delta M]_q, \quad c_1 = [\text{PK}_1 \cdot u + e_2]_q.$$

Three small polynomials per encryption, never needed again. They make encryption randomised, as IND-CPA requires (Lecture 8).

Decryption

$$M = \left[\left[\frac{t [c_0 + c_1 \cdot \text{SK}]_q}{q} \right] \right]_t.$$

Evaluate the ciphertext at the secret, undo the scaling, round, reduce.

Example — expand the bracket

- a. Substitute Encrypt and then KeyGen into $c_0 + c_1 \cdot \text{SK}$.
- b. What cancels? What is left?

Proposition

For a fresh ciphertext,

$$c_0 + c_1 \cdot \text{SK} = \Delta M + v \pmod{q}, \quad v = -e \cdot u + e_1 + e_2 \cdot \text{SK}.$$

The mask $a u \text{SK}$ cancels. The encryptor creates it inside $\text{PK}_0 u$ without knowing SK ; the decryptor creates it inside $c_1 \text{SK}$ without knowing u .

What remains is the message, scaled by Δ , plus the **noise** v .

Proposition

With $\|e\|, \|e_1\|, \|e_2\| \leq \beta$ and $\|u\|, \|SK\| \leq 1$,

$$\|v\| \leq n\beta + \beta + n\beta = 2n\beta + \beta.$$

If u, SK are only β -bounded: $\|v\| \leq 2n\beta^2 + \beta$.

Proof: $\|ab\| \leq n\|a\|\|b\|$ on eu and e_2SK . □

At $n = 4096$, $\beta = 19$:

$$2n\beta + \beta = 155\,667 \approx 2^{17.25}, \quad 2n\beta^2 + \beta = 2\,957\,331 \approx 2^{21.50}.$$

The second, conservative bound is the one quoted in the literature.

Theorem

If $c_0 + c_1\text{SK} = \Delta M + v \pmod{q}$ with $M \in R_t$ and

$$\|v\| < \frac{q}{2t},$$

then decryption returns M .

Proof: over $\mathbb{Z}$, $\frac{t}{q}(c_0 + c_1\text{SK}) \approx M + \frac{t}{q}v + tr$; rounding removes $\frac{t}{q}v$,
reduction mod t removes tr . □

Only the rounding can fail — and the failure is **silent**: a coefficient comes back wrong, with no warning.

The gap between $\|v\|$ and $q/2t$ is the **noise budget**.

Example — one failure

$t = 8$, $q = 1024$, one coefficient $m = 5$.

- a. Compute Δ and the bound $q/(2t)$.
- b. Decrypt at $v = 30$ and at $v = 70$: form $\left\lfloor \frac{t}{q}(\Delta m + v) \right\rfloor$.
- c. What comes back at $v = 70$? Could the decryptor tell?

Homomorphic addition

$$\text{EvalAdd}(\text{ct}^{(1)}, \text{ct}^{(2)}) = \left([c_0^{(1)} + c_0^{(2)}]_q, [c_1^{(1)} + c_1^{(2)}]_q \right).$$

The result has the shape of a fresh encryption of $M^{(1)} + M^{(2)}$, with randomness $u^{(1)} + u^{(2)}$, $e_i^{(1)} + e_i^{(2)}$.

Adding the decryption identities:

$$c_0^{(3)} + c_1^{(3)} \text{SK} = \Delta(M^{(1)} + M^{(2)}) + (v_1 + v_2), \quad \|v_{\text{add}}\| \leq \|v_1\| + \|v_2\|.$$

Noise grows **additively**: L fresh ciphertexts sum to $\|v\| \leq L(2n\beta + \beta)$.

At $n = 4096$, $\beta = 19$, fresh noise $\approx 2^{17.25}$; a budget of 2^{178} permits about 2^{160} additions.

Adding a plaintext M'

Add $\Delta M'$ to c_0 , leave c_1 . The noise is unchanged.

The whole difficulty of FHE is multiplication.

Session 1 ends here

Session 2: EvalMult and its noise growth, the depth budget, relinearization, and a full instance at $n = 4$, $t = 4$, $q = 16384$.

Multiply the two identities

Read ct as $c_0 + c_1 S \in R_q[S]$; decryption evaluates at $S = \text{SK}$ (Lecture 4).
Over $\mathbb{Z}$, $\text{ct}^{(i)}(\text{SK}) = \Delta M^{(i)} + v_i + q r_i$.

$$\begin{aligned}(\text{ct}^{(1)}\text{ct}^{(2)})(\text{SK}) &= \Delta^2 M^{(1)} M^{(2)} + \Delta(M^{(1)} v_2 + M^{(2)} v_1) \\&\quad + q(v_1 r_2 + v_2 r_1) + q\Delta(M^{(1)} r_2 + M^{(2)} r_1) \\&\quad + v_1 v_2 + q^2 r_1 r_2.\end{aligned}$$

The leading term carries Δ^2 . One factor must come off.

Dividing by Δ fails on $q^2 r_1 r_2$: $\Delta \nmid q^2$, and the non-integral residue survives reduction mod q .

- Scaling by t/q sends $q^2 r_1 r_2 \mapsto q t r_1 r_2$, which vanishes mod q .
- $t/q \approx 1/\Delta$, so the leading term still comes out right.
- t/q is not an integer, so we must **round**, adding an error r_e .

The result

With $M^{(1)}M^{(2)} = [M^{(1)}M^{(2)}]_t + t r_M$ and $v_1 v_2 = [v_1 v_2]_\Delta + \Delta r_v$:

$$\begin{aligned} \frac{t}{q}(\text{ct}^{(1)}\text{ct}^{(2)})(\text{SK}) &= \Delta[M^{(1)}M^{(2)}]_t + (M^{(1)}v_2 + M^{(2)}v_1) \\ &\quad + t(v_1 r_2 + v_2 r_1) + r_v \\ &\quad - [q]_t(r_M + M^{(1)}r_2 + M^{(2)}r_1) + r_e. \end{aligned}$$

The standard shape $\Delta M^{(3)} + v_3$ with $M^{(3)} = [M^{(1)}M^{(2)}]_t$.

Dominant term: $M^{(1)}v_2 + M^{(2)}v_1$. *The messages multiply the noises.*

Multiplicative noise growth

Up to lower-order terms,

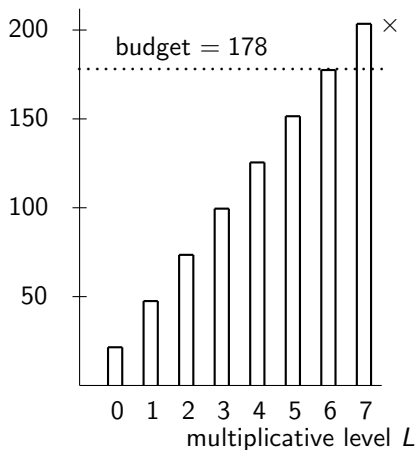
$$\|v_{\text{mult}}\| \lesssim \|v_i\| \cdot \underbrace{2 t n^2 \|SK\|}_G,$$

with $\|v_i\|$ the larger input noise.

After L sequential multiplications: $\|v_0\| \cdot G^L$ — exponential in the **depth**, and in nothing else.

At $n = 4096$, $t = 2$, ternary SK: $G = 2 \cdot 2 \cdot 4096^2 = 2^{26}$, i.e. 26 bits of budget per level.

A depth budget, end to end



$\log_2 \|v\|$ after L multiplications: $n = 4096$, $t = 2$, $\beta = 19$, $q = 2^{180}$; fresh noise $2^{21.50}$, $G = 2^{26}$.

Example — how many multiplications?

$n = 4096$, $t = 2$, $\beta = 19$, ternary SK, $q = 2^{180}$; use the conservative fresh-noise bound $2n\beta^2 + \beta$.

- a. Fresh noise $\|v_0\|$ and growth factor G , as powers of two.
- b. Budget $q/(2t)$, and the largest L with $\|v_0\| G^L < q/(2t)$. Check L and $L + 1$.
- c. How large must $\log_2 q$ be for $L = 12$?
- d. What is L at $t = 2^{16}$?

EvalMult: tensor, scale, round

$$\left(\left[\left[\left\lfloor \frac{t c_0^{(1)} c_0^{(2)}}{q} \right\rfloor \right] \right]_q, \left[\left[\left\lfloor \frac{t(c_0^{(1)} c_1^{(2)} + c_1^{(1)} c_0^{(2)})}{q} \right\rfloor \right] \right]_q, \left[\left[\left\lfloor \frac{t c_1^{(1)} c_1^{(2)}}{q} \right\rfloor \right] \right]_q \right)$$

The coefficients of S^0, S^1, S^2 in $(c_0^{(1)} + c_1^{(1)}S)(c_0^{(2)} + c_1^{(2)}S)$, formed over $\mathbb{Z}$ from centered representatives *before* scaling.

Two problems with the output

- 1 The ciphertext has grown from **two** ring elements to **three**.
- 2 Decryption now needs SK^2 :

$$c_0^* + c_1^*SK + c_2^*SK^2.$$

Both are repaired by relinearization.

Plaintext times ciphertext

$(M'c_0, M'c_1)$: no tensoring, no rescaling; noise grows to $n\|M'\|\|v\|$, not by G .

The problem

Given $ct^* = (c_0^*, c_1^*, c_2^*)$, find $\hat{ct} = (\hat{c}_0, \hat{c}_1)$ with

$$c_0^* + c_1^*SK + c_2^*SK^2 \approx \hat{c}_0 + \hat{c}_1SK \pmod{q}.$$

The evaluator cannot compute $c_2^*SK^2$ — it has no SK.

So SK^2 is **supplied in masked form**: the evaluation key.

EK

Sample fresh $a \leftarrow R_q$, $e \leftarrow \mathcal{X}$:

$$\text{EK} = (\text{EK}_0, \text{EK}_1) = \left(\left[- (a \cdot \text{SK} + e) + \text{SK}^2 \right]_q, a \right).$$

$$\text{EK}_0 + \text{EK}_1 \cdot \text{SK} = \text{SK}^2 - e.$$

An encryption of SK^2 under SK : BFV needs a circular-security assumption on top of RLWE.

Relinearize, basic form

$$\hat{c}_0 = [c_0^* + EK_0 \cdot c_2^*]_q, \quad \hat{c}_1 = [c_1^* + EK_1 \cdot c_2^*]_q.$$

$$\hat{c}_0 + \hat{c}_1 SK = c_0^* + c_1^* SK + c_2^* SK^2 - c_2^* \cdot e.$$

But c_2^* is spread over all of $\mathbb{Z}_q$:

$$\|c_2^* \cdot e\| \leq n \|c_2^*\| \beta \approx \frac{1}{2} n q \beta > q.$$

Basic relinearization destroys the ciphertext.

The fix: gadget decomposition

Fix a base w , $\ell = \lceil \log_w q \rceil$ (Lecture 4):

$$c_2^* = \sum_{i=0}^{\ell-1} w^i d_i, \quad \|d_i\| < w.$$

Relinearize with ℓ keys

$$\text{EK}^{(i)} = ([-(a_i \text{SK} + e_i) + w^i \text{SK}^2]_q, a_i),$$

$$\hat{c}_j = c_j^* + \sum_i \text{EK}_j^{(i)} d_i, \quad \|\mathbf{v}_{\text{Relinearize}}\| \leq n \ell w \beta.$$

Why the trade is favourable

A blow-up of order q is traded for an *additive* term of order w , at ℓ times the key material and ℓ times the work.

EvalMult has just multiplied the noise by $G \approx 2^{26}$, so the relinearization error is almost never the binding constraint.

Choosing w trades key size against noise. That is the only knob.

Relinearization is an instance of **key switching** ($SK^2 \rightarrow SK$); the same machinery rotates slots in BGV and CKKS.

Example — the instance

$$n = 4, \quad t = 4, \quad q = 2^{14} = 16384, \quad \beta = 1,$$

$$\text{SK} = 1 - X + X^3, \quad e = 1 - X^2 + X^3, \quad a = 7561 + 3002X + 15115X^2 + 8443X^3.$$

Encryption randomness: $u^{(1)} = 1 - X^2 + X^3$, $e_1^{(1)} = X - X^2$, $e_2^{(1)} = 1 - X$;
 $u^{(2)} = -1 + X + X^2$, $e_1^{(2)} = 1 - X^3$, $e_2^{(2)} = -X + X^2 + X^3$.

- a. Compute Δ , the budget $q/2t$, the fresh-noise bound and G .
- b. Compute PK.
- c. Encode $m_1 = 5$, $m_2 = 2$ in base 2; encrypt both.
- d. Find v_1, v_2 and decrypt.

Example — one addition, one multiplication

- a. Run EvalAdd. Decrypt and decode. Compare $\|v\|$ with $\|v_1\| + \|v_2\|$.
- b. Run EvalMult. Decrypt with $c_0^* + c_1^*SK + c_2^*SK^2$ and decode.
- c. Compare the observed noise with the worst case $\|v_i\| \cdot G$.

Example — relinearize

$$a' = 11923 + 604X + 9981X^2 + 14002X^3, \quad e' = X - X^2.$$

- a. Build EK and check $EK_0 + EK_1SK - SK^2 = -e'$.
- b. Relinearize with the basic procedure and decrypt. Compare $\|v\|$ with the budget and with $n\|c_2^*\|\beta$.
- c. Relinearize with base decomposition at $w = 4$. How many keys? Compare the observed noise with $n\ell w\beta$.

Example — where it breaks

Multiply the relinearized product by $\text{ct}^{(2)}$ again, to compute $5 \times 2 \times 2 = 20$.

- a. Is the noise still inside the budget?
- b. Is $M^{(4)}$ the correct product in R_t ?
- c. Decode at $X = 2$. What do you get, and why?
- d. Which of the two encoding conditions failed?

Two independent failure modes

In the last example **BFV works perfectly**: $M^{(4)} = M^{(3)}M^{(2)}$ in R_t , and the noise stays far inside the budget.

- $20 = 10100_2$ needs degree 4; $n = 4$ does not have it.
- $X^4 \equiv -1$ turns the leading bit into -1 in the constant term, which is 3 modulo $t = 4$.

A computation can fail through the noise budget or through the plaintext geometry — and neither announces itself.

Where the message sits

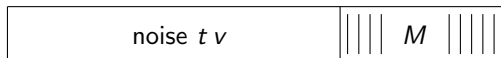
high bits

low bits

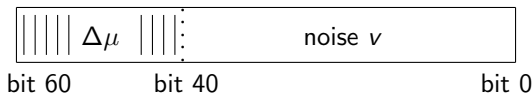
BFV



BGV



CKKS



	BFV	BGV	CKKS
Dec.	$c_0 + c_1 SK$ $= \Delta m + v$	$c_0 + c_1 SK$ $= M + tv$	$c_0 + c_1 SK$ $= \Delta \mu + v$
Recover by	scaling by t/q , rounding, then mod t	reducing mod t ; no rounding	dividing by Δ and accepting the error
Correct when	$\ v\ < q/(2t)$	$\ v\ < q_\ell/(2t)$	never exactly; precision about $\log_2 \Delta - \log_2 \ v\ $, in $\mathbb{C}^{n/2}$

- 1 Plaintexts in R_t , ciphertexts in $R_q \times R_q$; $PK_0 + PK_1SK = -e$.
Base- k encoding needs coefficients below t and degree below n .
- 2 $c_0 + c_1SK = \Delta M + v$, $v = -eu + e_1 + e_2SK$; fresh noise $\leq 2n\beta + \beta$ (conservatively $2n\beta^2 + \beta$).
- 3 Decryption is correct while $\|v\| < q/2t$; failure is silent.
- 4 EvalAdd adds noises. EvalMult scales the tensor product by t/q and multiplies noise by $G = 2tn^2\|SK\|$: depth is the scarce resource.
- 5 Relinearize absorbs $c_2^*SK^2$ with EK; gadget decomposition cuts its error from $\approx nq\beta$ to $n\ell w\beta$.