

Bootstrapping

Lecture 14

SEC 595 — Encrypted Computing

- 1 The Ceiling
- 2 Recryption
- 3 Bootstrappable Schemes
- 4 Circular Security
- 5 Bootstrapping GSW

Every scheme so far is levelled

BFV, BGV and CKKS (Lectures 10–12) all carry noise.

- Addition adds noise; multiplication multiplies it.
- Decryption fails once the noise passes a threshold.
- So the multiplicative depth is fixed at KeyGen.

A larger modulus raises the ceiling. It never removes it.

$$\text{Dec}_{\mathcal{E}}(\text{Sk}, \text{ct}) = m$$

$\text{Dec}_{\mathcal{E}}$ is fixed, public and finite, so it is a valid choice of Π in

$$\text{Evaluate}(\text{Pk}, \Pi, \text{ct}_1, \dots, \text{ct}_k).$$

The only obstacle: Evaluate needs its inputs *encrypted*, and one input is the secret key.

So encrypt the secret key.

Bootstrapping (Gentry, 2009)

Publish $\overline{Sk} = \text{Enc}(Pk, Sk)$ once. To refresh ct :

$$ct^\# = \text{Evaluate}(Pk, \text{Dec}_\mathcal{E}, \text{Enc}(Pk, ct), \overline{Sk})$$

No new algorithm: it is `Evaluate` with $\Pi = \text{Dec}_\mathcal{E}$.

$\text{Enc}(Pk, ct)$ has two layers: the outer one we compute under, the inner one the evaluated decryption removes.

Correct: $ct^\#$ encrypts $\text{Dec}_\mathcal{E}(\text{Sk}, ct) = m$. Nobody sees m , and the evaluator never holds Sk .

Refreshed: $ct^\#$ comes from one run of Evaluate on a fixed circuit with fresh inputs.

The noise of $ct^\#$ depends only on $\text{Dec}_\mathcal{E}$, **not** on the history of ct .

The old noise sits inside the plaintext of $\text{Enc}(\text{Pk}, ct)$, and the decryption being evaluated removes it.

The condition

A fresh ciphertext supports D levels. Evaluating $\text{Dec}_{\mathcal{E}}$ costs d .

Bootstrappable

$\mathcal{E}$ can evaluate its own decryption plus one more operation:

$$d < D.$$

evaluate $\rightarrow$ bootstrap $\rightarrow$ evaluate $\rightarrow \dots$

reaches any depth: levelled + bootstrapping = **FHE**.

Depth, not gate count, is the cost: decryption must be *shallow*. And D and d grow together with the parameters.

Example — where the budget goes

A fresh ciphertext supports $D = 10$ levels; $\text{Dec}_{\mathcal{E}}$ costs $d = 7$.

- a. How many useful levels does each bootstrap buy?
- b. How many bootstraps does a circuit of depth 100 need?
- c. What happens if $d = D$?

Encrypting the key under itself

$\overline{\text{Sk}} = \text{Enc}(\text{Pk}, \text{Sk})$ encrypts a key under its *own* public key. IND-CPA (Lecture 8) says nothing about that.

Circular security

$\mathcal{E}$ is **circular secure** if it stays secure when the attacker is also given $\text{Enc}(\text{Pk}, \text{Sk})$.

Assumed, not proved: no reduction from LWE, and no attack.

A chain of keys, $\text{Enc}(\text{Pk}_{i+1}, \text{Sk}_i)$, avoids the cycle — but allows only L bootstraps: the scheme is levelled again.

GSW decryption is one inner product

A GSW ciphertext (Lecture 13): $Cs = M \cdot Gs + e$, $\|e\|_\infty < q/4$.

Only the last row is needed:

$$(Cs)_\ell = \sum_{j=1}^{n+1} C_{\ell j} s_j = M \cdot \frac{q}{2} + e_\ell.$$

Is it nearer 0 or $q/2$? That is Regev decryption (Lecture 7).

C_{Dec} = one inner product + a top-bit test.

Take the secret binary, $\hat{s} \in \{0, 1\}^n$.

- **Selection:** $C_{\ell j} \hat{s}_j$ — a known number, kept or zeroed by a key bit. Free.
- **Adder tree:** add $n + 1$ numbers of $\log q$ bits.
- **Test:** read the top bit.

GSW supplies the gates: $G^{-1}(C_1)C_2$ is AND, $G - G^{-1}(C_1)C_2$ is NAND, and NAND is universal.

About $(n + 1) \log q$ full adders, AND-depth $O(\log n + \log \log q)$: wide and shallow.

The bootstrapping key

$$ek = (\text{Enc}(s, \hat{s}_1), \dots, \text{Enc}(s, \hat{s}_n))$$

one GSW ciphertext per key bit, under that same key.

Refreshing C : evaluate C_{Dec} with C hard-wired as constants and ek as the encrypted inputs.

The result $\tilde{C}$ decrypts to the same bit with new noise. Publishing ek needs circular security.

GSW multiplication is asymmetric (Lecture 13):

$$G^{-1}(C_1) C_2 : \quad \|e_{\text{new}}\|_{\infty} \leq \|e_1\|_{\infty} + \ell \|e_2\|_{\infty}.$$

Evaluate C_{Dec} **sequentially**: the running value (accumulator) in the slot C_1 , a fresh key ciphertext in C_2 .

Bootstrapping condition

After g gates with key noise E :

$$\|e_{\tilde{C}}\|_{\infty} \leq E(1 + g\ell) < \frac{q}{4}.$$

Additive, and independent of the input's noise. A balanced tree would give $E(\ell + 1)^d$.

Example — the bootstrapping budget

$n = 500$, key noise $E = 2^{10}$, $\ell = (n + 1) \log q$.

- a. $q = 2^{60}$: find the largest g with $E(1 + g\ell) < q/4$.
- b. C_{Dec} has about 30 000 gates. Is GSW bootstrappable here?
- c. Repeat (a) for $q = 2^{30}$.

- ① Levelled schemes have a fixed depth: noise grows with the history.
- ② Bootstrapping = Evaluate(Pk , $\text{Dec}_{\mathcal{E}}$, $\text{Enc}(\text{Pk}, \text{ct})$, $\text{Enc}(\text{Pk}, \text{Sk})$): same m , noise fixed by $\text{Dec}_{\mathcal{E}}$.
- ③ Bootstrappable: $d < D$. Then levelled becomes fully homomorphic.
- ④ Publishing $\text{Enc}(\text{Pk}, \text{Sk})$ needs circular security.
- ⑤ GSW decryption is one inner product and a top-bit test; with the accumulator in the decomposed slot the noise stays below $E(1 + g\ell)$.