

The GSW Scheme

Lecture 13

SEC 595 — Encrypted Computing

- 1 The Gadget Matrix
- 2 Approximate Eigenvectors
- 3 The Scheme
- 4 Homomorphic Addition
- 5 Homomorphic Multiplication
- 6 Toward Bootstrapping and TFHE

The gadget matrix G

Take $q = 2^k$ and base $B = 2$. The gadget vector is $g = (1, 2, 4, \dots, 2^{\log q - 1})^\top$ (Lecture 4).

Definition

$$G = \begin{pmatrix} g & 0 & \cdots & 0 \\ 0 & g & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g \end{pmatrix} \in \mathbb{Z}_q^{\ell \times (n+1)}, \quad \ell = (n+1) \log q.$$

One block of $\log q$ rows per coordinate of the key. G is fixed and public.

The last row of G is $(0, \dots, 0, q/2)$.

The gadget inverse G^{-1}

G is not square and has no inverse. G^{-1} replaces each entry by its $\log q$ binary digits.

Gadget inverse

$$G^{-1}(C) \cdot G = C,$$

and $G^{-1}(C)$ is $\log q$ times wider than C .

Flatness (Lecture 4)

For every $a \in \mathbb{Z}_q$: $\langle g, G^{-1}(a) \rangle = a$ and $\|G^{-1}(a)\|_\infty \leq B/2$, **uniformly in** a .

With $B = 2$ the entries of $G^{-1}(a)$ are bits.

An error $e \in \mathbb{Z}_q^\ell$ multiplied by an arbitrary $C \in \mathbb{Z}_q^{\ell \times \ell}$:

$$\|C\|_\infty \approx q/2 \implies \|Ce\|_\infty \text{ is of order } q.$$

Multiplied by $G^{-1}(C)$ instead:

$$\|G^{-1}(C) e\|_\infty \leq \ell \cdot \frac{B}{2} \cdot \|e\|_\infty.$$

The factor depends on the *parameters*, not on the ciphertext. This is all of GSW's noise behaviour.

Example — decomposing by hand

$q = 64$, $B = 2$, so $g = (1, 2, 4, 8, 16, 32)$.

- a. Compute $G^{-1}(45)$ and check $\langle g, G^{-1}(45) \rangle = 45$.
- b. Compare $\|G^{-1}(45)\|_{\infty}$ with $\|45\|_{\infty}$ (centred).
- c. Redo it with $B = 4$, $g = (1, 4, 16)$, balanced digits. What changed?

Let $C_1 v = \lambda_1 v$ and $C_2 v = \lambda_2 v$. Then

$$(C_1 + C_2) v = (\lambda_1 + \lambda_2) v, \quad (C_1 C_2) v = C_1(\lambda_2 v) = \lambda_1 \lambda_2 v.$$

Read C_i as a ciphertext, λ_i as its message, v as the key. Adding and multiplying matrices adds and multiplies messages, and the key never appears.

Not secure: eigenvectors are computable by linear algebra.

Fix: make the relation hold only *approximately*, with LWE noise as the slack.

The approximate-eigenvector relation

Definition

$C \in \mathbb{Z}_q^{\ell \times (n+1)}$ encrypts $M \in \mathbb{Z}_2$ under $s \in \mathbb{Z}_q^{n+1}$ when

$$C \cdot s = M \cdot Gs + e, \quad \|e\|_\infty \text{ small.}$$

Eigenvector picture	GSW
matrix C	the ciphertext
eigenvector v	Gs , fixed by the key
eigenvalue λ	the message bit M
exact equality	equality up to e

The scheme is $\text{GSW}(m, n, q, \chi)$:

- n — the LWE dimension;
- $q = 2^k$ — the modulus;
- χ — the error distribution on $\mathbb{Z}_q$ (Lecture 6);
- m — the number of LWE samples (rows). It must match the height of G , so $m = \ell = (n + 1) \log q$.

A ciphertext is an $\ell \times (n + 1)$ matrix over $\mathbb{Z}_q$: about $(n + 1)^2 \log q$ elements to carry one bit.

KeyGen(1^n)

Sample $\hat{s} \leftarrow \mathbb{Z}_q^n$ uniformly; output $s = \begin{pmatrix} -\hat{s} \\ 1 \end{pmatrix} \in \mathbb{Z}_q^{n+1}$.

The last coordinate 1 makes an LWE sample annihilate itself:

$$(a \parallel \langle a, \hat{s} \rangle + e) \begin{pmatrix} -\hat{s} \\ 1 \end{pmatrix} = e.$$

No evaluation key is needed: neither homomorphic operation uses key material.

$\text{Enc}(s, M), M \in \mathbb{Z}_2$

- 1 Sample $A \leftarrow \mathbb{Z}_q^{\ell \times n}$ and $e \leftarrow \chi^\ell$.
- 2 $B = (A \parallel A\hat{s} + e)$, so $B \cdot s = e$.
- 3 Output $C = B + M \cdot G$.

A GSW ciphertext is $M \cdot G$ **plus an encryption of zero**.

Security comes from B alone (LWE, Lecture 6); structure comes from the public G alone.

Proposition

For $C \leftarrow \text{Enc}(s, M)$: $C \cdot s = M \cdot Gs + e$.

Proof.

$$Cs = (A \parallel A\hat{s} + e) \begin{pmatrix} -\hat{s} \\ 1 \end{pmatrix} + M \cdot Gs = (-A\hat{s} + A\hat{s} + e) + M \cdot Gs. \quad \square$$

The approximate-eigenvector relation is a consequence of the construction, not an assumption.

$\text{Dec}(s, C)$

Compute $v = C \cdot s$. Output 0 if $\|v\|_\infty < q/4$ (centred), else 1.

Correctness

If $\|e\|_\infty < q/4$ then $\text{Dec}(s, C) = M$.

$M = 0$: $v = e$, all entries small.

$M = 1$: the last entry of Gs is $q/2$ (last row of G times $s_{n+1} = 1$), so $|v_\ell| \geq q/2 - \|e\|_\infty > q/4$.

One matrix–vector product and a comparison: a shallow circuit.

Example — a toy instance

$n = 1$, $q = 64$, $B = 2$, $\ell = 12$, χ uniform on $\{-1, 0, 1\}$, $\hat{s} = 21$.

- a. Write s and $G^\top$. What is the threshold $q/4$?
- b. Compute Gs (centred). Which entry has magnitude $q/2$ for every key?
- c. Encrypt $M = 1$ with

$$A_1 = (37, 5, 52, 19, 61, 8, 44, 27, 3, 50, 16, 39)^\top,$$

$$e_1 = (1, -1, 0, 1, -1, 1, 0, -1, 1, 0, -1, 1)^\top.$$

Example — decrypting

Same parameters. C_1 is from the previous example; C_2 encrypts 1 with $e_2 = (-1, 1, 1, 0, -1, 0, 1, 1, -1, -1, 0, 1)^\top$:

$$C_2^\top = \begin{pmatrix} 12 & 60 & 28 & 15 & 62 & 1 & 62 & 15 & 29 & 41 & 6 & 54 \\ 38 & 3 & 57 & 19 & 5 & 53 & 24 & 62 & 36 & 36 & 14 & 15 \end{pmatrix}.$$

- a. Compute $v_1 = C_1 s$, check $v_1 - Gs = e_1$, and decrypt.
- b. Let $B_2 = C_2 - G$. Compute $B_2 s$ and decrypt.

$$(C_1 + C_2)s = (M_1Gs + e_1) + (M_2Gs + e_2) = (M_1 + M_2)Gs + (e_1 + e_2).$$

Noise

$\|e_1 + e_2\|_\infty \leq \|e_1\|_\infty + \|e_2\|_\infty$. Summing k fresh ciphertexts with $\|e\|_\infty \leq E$ decrypts while $k < q/(4E)$.

The noise grows by a summand, not a factor. No key material, no post-processing.

Addition is not quite XOR

$M_1 + M_2$ is computed in $\mathbb{Z}_q$. For $M_1 = M_2 = 1$ the eigenvalue is 2, and Dec returns 1, not $\text{XOR}(1, 1) = 0$.

$C_1 + C_2$ is XOR only when $M_1 M_2 = 0$.

Genuine gates (using multiplication, next section)

$$\text{XOR} : C_1 + C_2 - 2 G^{-1}(C_1) C_2, \quad \text{NAND} : G - G^{-1}(C_1) C_2.$$

Eigenvalues $M_1 + M_2 - 2M_1 M_2$ and $1 - M_1 M_2$.

Example — adding

Use C_1 , C_2 and $B_2 = C_2 - G$ from the toy instance.

- a. Compute $(C_1 + B_2)s$ and its noise. Compare with $e_1 + e_2$. Decrypt.
- b. Decrypt $C_1 + C_2$. Is the result $\text{XOR}(1, 1)$?
- c. How many fresh ciphertexts may be summed at $q = 64$, $E = 1$?

The product ciphertext

$C_1 C_2$ does not typecheck: both are $\ell \times (n + 1)$.

$G^{-1}(C_1)$ is $\ell \times \ell$, so $G^{-1}(C_1) C_2$ is $\ell \times (n + 1)$ — a ciphertext.

Definition

$$C_{\times} = G^{-1}(C_1) \cdot C_2.$$

Asymmetric: C_1 is decomposed, C_2 is not.

What multiplication does

$$\begin{aligned}(G^{-1}(C_1) C_2) s &= G^{-1}(C_1)(M_2 Gs + e_2) \\&= M_2 C_1 s + G^{-1}(C_1) e_2 \\&= (M_1 M_2) Gs + \underbrace{(M_2 e_1 + G^{-1}(C_1) e_2)}_{e_{\text{new}}}\end{aligned}$$

The only step that uses anything is $G^{-1}(C_1) G = C_1$.

The message becomes $M_1 M_2$: the **AND** gate. With $\text{NAND} = G - C_{\times}$ we have a universal gate.

Proposition ($B = 2$)

$$\|e_{\text{new}}\|_{\infty} \leq \|e_1\|_{\infty} + \ell \cdot \|e_2\|_{\infty}, \quad \ell = (n+1) \log q.$$

$M_2 \in \{0, 1\}$, so the first term is at most $\|e_1\|_{\infty}$. Each entry of $G^{-1}(C_1) e_2$ is a sum of at most ℓ entries of e_2 .

The factor is the parameter ℓ , not $\|C_1\|_{\infty} \approx q/2$.

Subtracting from the noiseless G (NAND) adds no noise.

The noise of C_1 is *added*; the noise of C_2 is *multiplied by* ℓ . Chain d multiplications with fresh ciphertexts ($\|e\|_\infty \leq E$):

Accumulator in the decomposed slot (C_1)

$N_d \leq N_{d-1} + \ell E$, so $N_d \leq E(1 + d\ell)$ — **linear** in d .

Accumulator in the other slot (C_2)

$N_d \leq E + \ell N_{d-1}$ — **exponential** in d .

Bootstrapping (Lecture 14) exploits this: a long chain of multiplications by fresh key encryptions grows the noise only linearly.

Example — multiplying by hand

Use C_1, C_2 from the toy instance. Row 1 of C_1 is $(38, 10)$.

- a. Write row 1 of $G^{-1}(C_1)$ and check it recomposes to $(38, 10)$.
- b. Bound $\|e_{\text{new}}\|_{\infty}$. Is decryption guaranteed? Would it be at $q = 32$?
- c. Given $C_{\times}s = (-18, 22, -22, 24, -17, -30, 1, 0, 4, 10, 14, -31)^{\top}$, find e_{new} and decrypt.
- d. Decrypt $G - C_{\times}$.

Example — how deep a chain?

$$n = 8, q = 2^{20}, E = 1.$$

- a. Compute ℓ and $q/4$.
- b. Find the largest d with $E(1 + d\ell) < q/4$.
- c. Repeat with the accumulator in the other slot.

The last row and binary keys

Row ℓ of $Cs = M \cdot Gs + e$, with $(Gs)_\ell = q/2$:

$$\langle c_\ell, s \rangle = M \cdot \frac{q}{2} + e_\ell.$$

The last row alone is a **Regev ciphertext** (Lecture 7): decryption is one inner product and “nearer 0 or $q/2$?”.

Binary keys

Nothing so far needs $\hat{s}$ uniform: take $\hat{s} \in \{0, 1\}^n$. LWE stays hard (for a somewhat larger n), and $\sum_j c_{\ell j} \hat{s}_j$ only *selects* known numbers.

Lecture 14 bootstraps GSW by evaluating this inner product.

The external product

An LWE ciphertext $c \in \mathbb{Z}_q^{n+1}$: $\langle c, s \rangle = \mu + e_c$.

External product

$$C \boxdot c = G^{-1}(c) \cdot C, \quad \langle C \boxdot c, s \rangle = M\mu + (Me_c + G^{-1}(c) e)$$

with noise at most $\|e_c\|_\infty + \ell \|e\|_\infty$.

- One row of the product of Section 5: ℓ times cheaper, and the result is again a vector.
- c sits in the decomposed slot: a chain of products grows its noise only additively.

TFHE (Lecture 15) is this operation over $\mathbb{Z}_q[X]/(X^N + 1)$.

Example — an external product

$s = (43, 1)$, C_1 encrypts 1 (toy instance). Let $c = (54, 15)$, the last row of C_2 .

- a. Compute $\langle c, s \rangle$. Which bit does c encrypt, and with what noise?
- b. Write $G^{-1}(c)$.
- c. Compute $C_1 \boxdot c$ by adding the selected rows of C_1 .
- d. Decrypt the result and find its noise.

- ① G stacks copies of $g = (1, 2, \dots, 2^{\log q-1})$, so $\ell = (n+1) \log q$; G^{-1} is flat: $\|G^{-1}(a)\|_{\infty} \leq B/2$ for every a .
- ② Key $s = (-\hat{s}, 1)$; ciphertext $C = B + M \cdot G$ with B a stack of LWE samples. Security is B , structure is G .
- ③ $Cs = M \cdot Gs + e$: an approximate eigenvector. Dec tests $\|Cs\|_{\infty} < q/4$.
- ④ Addition $C_1 + C_2$: noises add. Multiplication $G^{-1}(C_1) C_2$: AND, noise $\leq \|e_1\|_{\infty} + \ell \|e_2\|_{\infty}$; $G - G^{-1}(C_1) C_2$ is NAND.
- ⑤ Noise growth is asymmetric: keep the accumulator in the decomposed slot and d products cost $E(1 + d\ell)$.
- ⑥ Toward Lectures 14–15: the last row is a Regev ciphertext, keys may be binary, and the external product $G^{-1}(c) C$ keeps vectors.