A Practical Adaptive Key Recovery Attack on the LGM (GSW-like) Cryptosystem

Prastudy Fauzi[1], **Martha Norberg Hovd**[1,2] and Håvard Raddum[1]

Simula UiB[1], Bergen, Norway
University of Bergen[2], Norway

PQCrypto
2021

Introduction: Background

LGM is an *LHE* scheme based on the *FHE* scheme GSW, designed to achieve *IND-CCA1* security.

- *LHE*: Limited evaluation of ciphertexts.
- *FHE*: Unlimited evaluation of ciphertexts.
- *IND-CCA1*: An adversary with limited access to a decryption oracle cannot distinguish between two encrypted messages.

Introduction: IND-CCA1 is hard to achieve

LGM only concrete scheme believed to be IND-CCA1 secure.

LGM

LGM: Secret Key Generation

For $i \in [1, t] : \mathbf{e}_i \leftarrow \chi^m$

$$\mathbf{s}_i = (\mathbf{r}_i \parallel -\mathbf{e}_i^T)^T = (\underbrace{0, \ldots, 1, \ldots, 0}_{\text{length t, 1 in pos. i}}, \underbrace{-\mathbf{e}_i^T}_{\text{length m}})^T$$

Secret key: $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_t)$

LGM: Decryption

$\mathbf{C} \in \mathbb{Z}_q^{(t+m)\times N}$, j such that $2^{j-1} \in (q/4, q/2]$

Sample $(\lambda_1, \ldots, \lambda_t) \in \mathbb{Z}_q^t \setminus \{0\}^t$

$\mathbf{s}' = \sum_{i=1}^t \lambda_i \mathbf{s}_i = (\lambda_1, \ldots, \lambda_t, \sum_{i=1}^t \lambda_i e_{i,1}, \ldots, \sum_{i=1}^t \lambda_i e_{i,m})$

Choose index i such that $\lambda_i \neq 0$ and calculate $I(i)$

Compute $u = \langle \mathbf{C}_I, \mathbf{s}' \rangle \mod q$ and return $|\lfloor u/2^{j-1} \rceil| \in \{0, 1\}$

LGM: Parameters and assumptions

Parameter	Value
Secret keys, t	190 or 400
Length of $\mathbf{e}_i$, m	525
Standard deviation of discrete Gaussian (χ), σ	25
Modulus, q	94980001

Table: Parameter choices for 120-bit security.

We assume a uniform and binary λ-distribution

Attack

Attack: Procedure

- Each ciphertext is queried T times
- Estimate $\frac{1}{2}\sum_{i=1}^{t} e_{i,1}$
- Estimate $e_{i,1} + \frac{1}{2}\sum_{k\neq i}^{t} e_{k,1}$
- Estimate $e_{i,1}$

Attack: Estimation of the Baseline

$$D_\alpha = \begin{bmatrix} \alpha & 0 & \cdots & 0 \\ 0 & \alpha & \cdots & 0 \\ 0 & 0 & \cdots & \alpha \\ 1 & 1 & \cdots & 1 \\ & & \mathbf{0}_{(m-1)\times t} & \end{bmatrix}$$

$$u(D_\alpha) = \alpha + \sum_{i=1}^{t} \lambda_i e_{i,1}$$

$$\alpha_{est} + 1/2 \sum_{i=1}^{t} e_{i,1} = 2^{j-2} + \epsilon$$

Attack: Estimation of a Specific Element

$$R_{a,i} = \begin{bmatrix} & \mathbf{0}_{(i-1)\times t} & \\ a\ a & \cdots & a \\ & \mathbf{0}_{(t-i)\times t} & \\ 1\ 1 & \cdots & 1 \\ & \mathbf{0}_{(m-1)\times t} & \end{bmatrix}$$

$$u(R_{a,i}) = \lambda_i a + \lambda_i e_{i,1} + \sum_{k\neq i} \lambda_k e_{k,1}$$

$$a_{est} + e_{i,1} + 1/2 \sum_{k\neq i} e_{k,1} = 2^{j-2} + \epsilon_i$$

Attack: Recovering $e_{i,1}$

$$\alpha_{est} + 1/2 \sum_{i=1}^{t} e_{i,1} = 2^{j-2} + \epsilon$$

$$a_{est} + e_{i,1} + 1/2 \sum_{k \neq i} e_{k,1} = 2^{j-2} + \epsilon_i$$

$$e_{i,1} = \lfloor 2(\alpha_{est} - a_{est}) + 2(\epsilon_i - \epsilon) \rceil$$

Attack: Results

Secret keys	Sample size	Time	Correctly recovered elements
$t = 190$	$T = 95,000,000$	12 hours	$516/525$
$t = 400$	$T = 200,000,000$	48 hours	$525/525$

Table: The attacks were performed on a server with 75 CPUs running in parallel.

Possible countermeasures?

- Fix $(\lambda_1, \ldots, \lambda_t)$ for a given ciphertext matrix $\mathbf{C}$.
  - Can make small changes to both D_α and $R_{a,i}$ without affecting the attack.
- Decrypt a ciphertext multiple times and return a value only if the decryptions are consistent.
  - The attack is the same, only with three return values: $(0, 1, \perp)$.
- Add a ciphertext check during decryption.
  - Not clear how to achieve this.

LGM is not IND-CCA1 secure

No concrete HE scheme is IND-CCA1 secure

Full version: ePrint 2021/658