Полностью гомоморфное шифрование, основанное на кодах Рида-Маллера

Доклад

Доледенок Илья Вадимович

Науч. рук.:
Чижов Иван Владимирович,
доцент кафедры ИБ ВМК МГУ,
канд. физ.-мат. наук

МГУ имени М.В.Ломоносова
факультет вычислительной математики и кибернетики
кафедра информационной безопасности

22 января 2026 г.

Гомоморфное шифрование

Гомоморфное шифрование отличается от обычного тем, что кроме алгоритмов формирования ключей *KeyGen*, шифрования *Encrypt*, расшифрования *Decrypt*, у него еще есть алгоритм *Eval* и некоторое допустимое множество операций F. Для каждой операции $f \in F$, и шифртекстов $c_1, \ldots, c_n$, где $c_i = \mathrm{Encrypt}(m_i)$ выполняется: $\mathrm{Decrypt}(c) = f(m_1, \ldots, m_n)$, где $c = \mathrm{Eval}(f, c_1, \ldots, c_n)$.

Гомоморфное шифрование

Гомоморфное шифрование разделяется на 3 типа:

- Частично гомоморфное, если поддерживается единственная операция с бесконечным числом применений к данным
- Ограниченно гомоморфное, если поддерживается произвольное количество операций с конечным числом применений
- Полностью гомоморфное, если поддерживается произвольное количество операций с бесконечным числом применений

Определения

- Расстояние Хемминга $|v|$ для вектора $v \in \mathbb{F}_2^n$ это количество ненулевых элементов в v
- Линейный $[n, k, d]$ код $\mathcal{C}$ — это k-размерное линейное подпространство $\mathbb{F}_2^n$ с минимальным расстоянием Хемминга равным d
- Код Рида-Маллера $RM(r, m)$ — это код порядка r и длины 2^m, определяемый множеством всех векторов-значений булевых функций $f(v_1, \ldots, v_m)$, задаваемых многочленом Жегалкина степени не более r.
- Расширенный код Рида-Маллера — это множество матриц $\{W_1, W_2, W_3, \ldots\}$, где $W_i = m_i \odot G_{rm}$, $W_i \in (\mathbb{F}_2^n)^k$, $m_i \in \mathbb{F}_2^k$, $G_{rm} \in (\mathbb{F}_2^n)^k$ - порождающая матрица кода $RM(r, m)$
- Функция перестановки σ_S — это функция, переставляющая элементы матрицы $V \in (\mathbb{F}_2^n)^k$ в соответствии с перестановочным ключом $S \in (\mathbb{F}_{(x,y)}^n)^k$, являющимся матрицей перетасованных пар индексов (x, y).
  То есть, если у нас есть матрица $V \in (\mathbb{F}_2^n)^k$, то $W = \sigma_S(V)$ такова, что $W[i, j] = V[i', j']$, где $[i', j'] = S[i, j]$.

Описание схемы

KeyGen: $(r, m) \to K$

- Вычисляем $k = 1 + C_m^1 + C_m^2 + \cdots + C_m^r$
  $n = 2^m$
  $d = 2^{m-r}$
- Выбираем $S_1 \subset 0, 1, \ldots, n-1$ так, чтобы $\frac{d}{2} < |S_1| < d$
- Выбираем ключ перестановки S_2 для функции σ_{S_2}
- Выдаем секретный ключ $K = (S_1, S_2)$

Encrypt: $(K, m) \to C, m \in \mathbb{F}_2^k$

- Рандомно генерируем матрицу ошибок
  $E_{S_1} = (e_1, e_2, \ldots, e_k) : e_i \in \mathbb{F}_2^n, supp(e_i) \subseteq S_1$
- Вычисляем $C = \sigma_{S_2}(m \odot G_{rm} + E_{S_1})$
- Выдаем шифртекст C

Описание схемы

Decrypt: $(K, C) \rightarrow m$

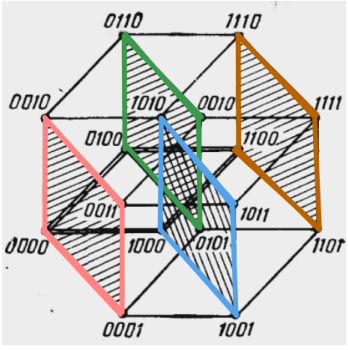
- Применяем обратную перестановку σ'_{S_1} к $C : W = \sigma'_{S_1}(C) = m \odot G_{rm} + E_{S_1} = (w_1, w_2, \ldots, w_k)$
- $w = w_1 + w_2 + \cdots + w_k$
- Мажоритарное декодирование вектора w с известными местами ошибок: $m = Decode(S_1, w)$

Метод Decode. Пример для RM(2, 4)

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
v_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
v_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
v_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
v_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
v_1v_2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
v_1v_3	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
v_1v_4	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
v_2v_3	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
v_2v_4	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1
v_3v_4	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
$v_1v_2v_3$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
$v_1v_2v_4$	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
$v_1v_3v_4$	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
$v_2v_3v_4$	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
$v_1v_2v_3v_4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	Кр	Кр	Кр	Кр	З	З	З	З	С	С	С	С	К	К	К	К

$w = (y_0, y_1, \ldots, y_{15})$

$m = (x_0, x_1, x_2, x_3, x_4, x_{12}, x_{13}, x_{14}, x_{23}, x_{24}, x_{34})$



Метод Decode. Алгоритм

- Идем с конца сообщения m к началу. Пробегаемся по s от r до 0.
- Перебираем все различные комбинации индексов $i_1, \ldots, i_s$ и вычисляем $x_{i_1 \ldots i_s}$:
  - Находим все подкубы B^s в гиперкубе B^m такие, что в вершинах каждого из этих подкубов координаты с индексами $i_1, \ldots, i_s$ меняются, а все остальные не изменяются.
  - Перебираем эти кубы, пока не найдем тот куб B', у которого нет пересечения между множеством чисел в его вершинах с множеством S_1
  - Тогда $x_{i_1 \ldots i_s} = \sum\limits_{j \in B'} y_j$
- $w = w + \sum\limits_{i_1, \ldots, i_s} G_{rm}[v_{i_1} \cdots v_{i_s}]$, где $G_{rm}[v_{i_1} \cdots v_{i_s}]$ это строка, соответствующая одночлену $v_{i_1} \cdots v_{i_s}$

Eval

Поддерживается две операции. Пусть есть два сообщения m_1 и m_2, $c_i = Encrypt(m_i, K), i = 1, 2$:

- Поэлементное сложение.
  $m_3 = m_1 + m_2$, а $c_3 = Encode(m_3, K) = c_1 + c_2$
- Поэлементное умножение.
  $m_3 = m_1 \odot m_2$, а $c_3 = Encode(m_3, K)$ такова, что $c_3[i, j] = c_1[i, j] \cdot c_2[i, j]$

Время работы на практике

Характеристики системы: процессор 11th Gen Intel(R) Core(TM) i5-11300H @ 3.10GHz, 16 ГБ оперативной памяти, система Ubuntu 24.04.2 LTS. Программа написана на python

(r, m)	Шифрование	Расшифровка	Сумма	Произведение
$(1, 4)$	0.0000551919	0.0001657619	0.0000007483	0.0000006263
$(2, 5)$	0.0002487523	0.0006183957	0.0000010014	0.0000008491
$(1, 8)$	0.0010219483	0.0013443941	0.0000011781	0.0000013130
$(3, 8)$	0.0101977236	0.0126601022	0.0000027009	0.0000030117
$(1, 12)$	0.0241815319	0.0249023124	0.0000043755	0.0000050490
$(2, 12)$	0.1385504727	0.1482715894	0.0000287139	0.0000258685
$(1, 15)$	0.2254706886	0.2424323083	0.0000560185	0.0000570237
$(1, 18)$	2.1394342389	2.4840883483	0.0008174370	0.0007413267

Список литературы

- Ratnakumari Challa, VijayaKumari Gunta. *A Modified Symmetric Key Fully Homomorphic Encryption Scheme Based on Reed-Muller Code.* In: Bagdad Science Journal 18.2, 899–906 (2021).
- Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. *Теория кодов, исправляющих ошибки.* Пер. с англ. — М.: Связь, 1979. — 744 с.