Дерандомизация в контексте «луковичной» архитектуры криптографической защиты облачной базы данных

А. В. Трепачева, Ф.Б. Буртыка

Южный Федеральный Университет

3 декабря 2015 г.

Гомоморфное шифрование

Гомоморфная схема шифрования (ГСШ) *– это криптосистема, позволяющая проводить некоторые вычисления над данными в зашифрованном виде с последующей возможностью извлечения результата вычислений над соответствующими открытыми текстами с помощью секретного ключа.*

Существует два основных типа ГСШ:

- *Полностью гомоморфные схемы шифрования (ПГСШ)* – криптосистемы, позволяющие эффективно проводить вычисление любой функции гомоморфно.
- *Частично гомоморфные схемы шифрования (ЧГСШ)* – криптосистемы, разрешающие эффективное вычисление некоторых функций (но не всех возможных) гомоморфно.

Использование гомоморфного шифрования на практике

Известные системы для вычислений над зашифрованными данными СУБД:

- CryptDB [1]
- Проект «защищенная облачная БД» исследователей из НГУ [2].
- Работы исследователей из Индии и Китая

прочие:

- MyLar – система защиты web-приложений
- PrivStats и VPriv – системы защиты конфиденциальности (от недоверенного сервера) места нахождения пользователя в мобильных системах.

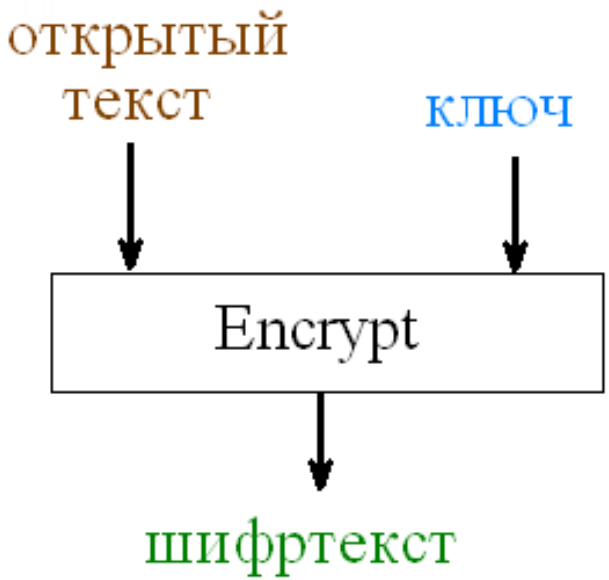
[1] Popa, R. A., Redfield, C., Zeldovich, N., Balakrishnan, H. CryptDB: Processing queries on an encrypted database //Communications of the ACM. – 2012. – Т. 55. – №. 9. – С. 103-111.
[2]Shatilov, K., Boiko, V., Krendelev, S., Anisutina, D., Sumaneev, A. Solution for secure private data storage in a cloud //Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on. – IEEE, 2014. – С. 885-889.

Алгоритмы полностью гомоморфного шифрования (ПГШ)

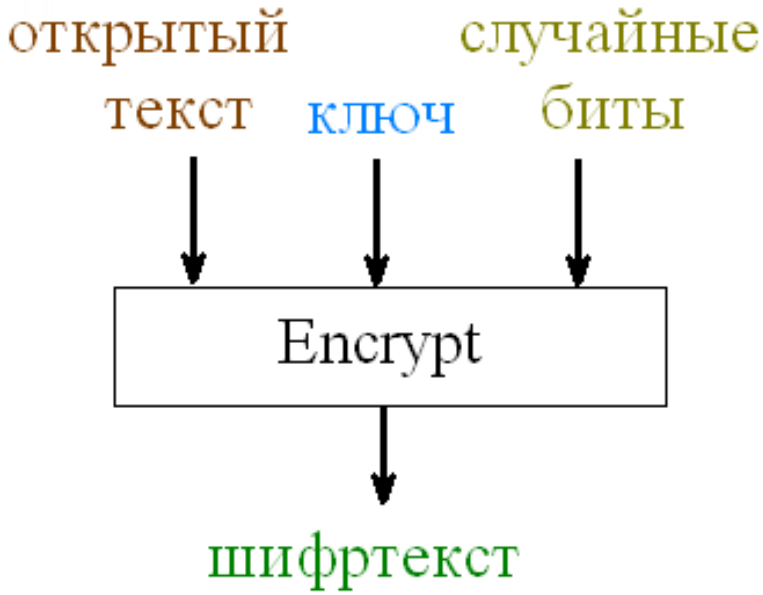
Полностью гомоморфная криптосистема (криптосхема) – это четверка алгоритмов

- Алгоритм KeyGen генерирует ключи (секретный и, возможно, открытый)
- Алгоритм Encrypt зашифровывает открытые тексты с помощью ключа
- Алгоритм Decrypt расшифровывает шифртексты с помощью ключа
- Алгоритм Evaluate производит вычисления над открытыми текстами

Вероятностное и детерминированное шифрование



Детерминированное шифрование



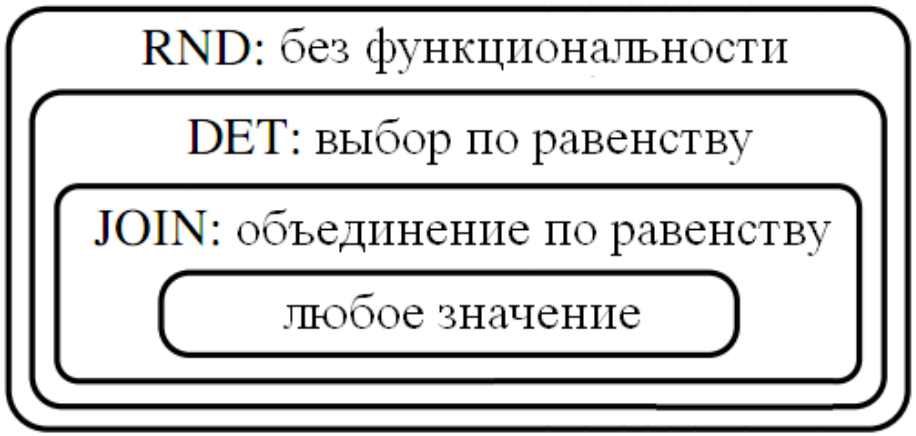
Вероятностное шифрование

Вероятностное и детерминированное шифрование

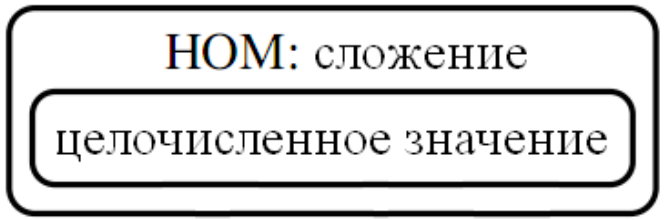
Гомоморфное шифрование стараются сделать вероятностным, поскольку

- из-за гомоморфных свойств нейтральный по операции элемент определен единственным образом (таким образом, не может быть нетривиально зашифрован)
- в случае детерминированного шифрования невозможно обеспечить семантическую криптостойкость (неразличимость двух шифртекстов), поскольку по условиям протокола проверки криптоаналитик может запустить процедуру Encrypt для различаемых открытых текстов и сравнить её результат с имеющимся шифртекстом.

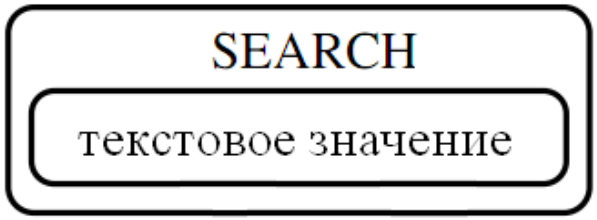
CryptDB: Луковичная архитектура



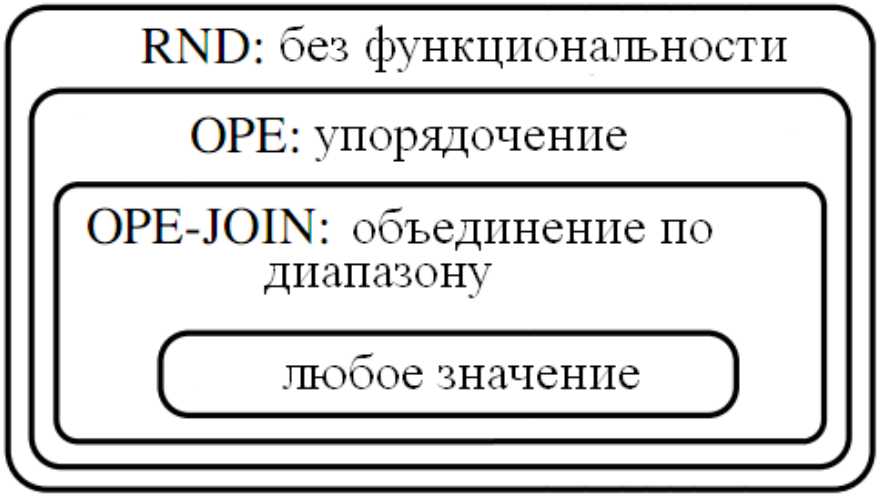
Луковица "Равенство"



Луковица "Сложение"



Луковица "Поиск"



Луковица "Упорядочение"

Анализ защищенности

Одному атрибуту БД может соответствовать несколько столбцов, поскольку одному атрибуту может соответствовать несколько «луковиц» – режимов обработки – и поэтому они должны быть продублированы в нескольких столбцах, зашифрованных с использованием разных криптосистем, поддерживающих необходимую функциональность

Анализ защищенности

Наблюдение: Сочетания луковиц сложение и сравнение на равенство ведет к преобразованию гомоморфного шифра в детерминированный!

Дерандомизация

Дерандомизация

Вероятностный алгоритм (принимает на вход кроме данных еще и случайные биты) → Детерминированный алгоритм (не использует случайные биты)

Дерандомизация

Вероятностная криптосистема → Детермированная криптосистема

Дерандомизация

Вероятностной криптосистеме можно поставить в соответствие некоторое количество детерминированных криптосистем, каждая из которых получается зафиксированием набора случайных битов, принимаемых на вход алгоритмом Encrypt. (если на вход в вероятностной криптосистеме подавалось λ случайных битов, то будет 2^{λ} детерминированнных криптосистем)

Дерандомизация

Пусть $\mathcal{E} = \{KeyGen, Enc, Dec\}$ – вероятностная криптосистема, $c_1, ..., c_n$ – шифртексты, произведенные алгоритмом Enc криптосистемы $\mathcal{E}$, т.е. $c_i = Enc(m_i, \mathbf{k}, r_i)$, где r_i – случайные элементы, $\mathbf{k}$ – ключ зашифрования.

Если существует *полиномиальный* алгоритм $\mathcal{A}$, который преобразует каждое c_i в c_i' (где c_i' – шифртексты криптосистемы $\mathcal{E}' = \{KeyGen, Enc', Dec\}$, произвольного детерминированного варианта криптосистемы $\mathcal{E}$, т.е. $c_i' = Enc'(m_i, \mathbf{k})$), то будем говорить, что криптосистема $\mathcal{E}$ является нестойкой к дерандомизации

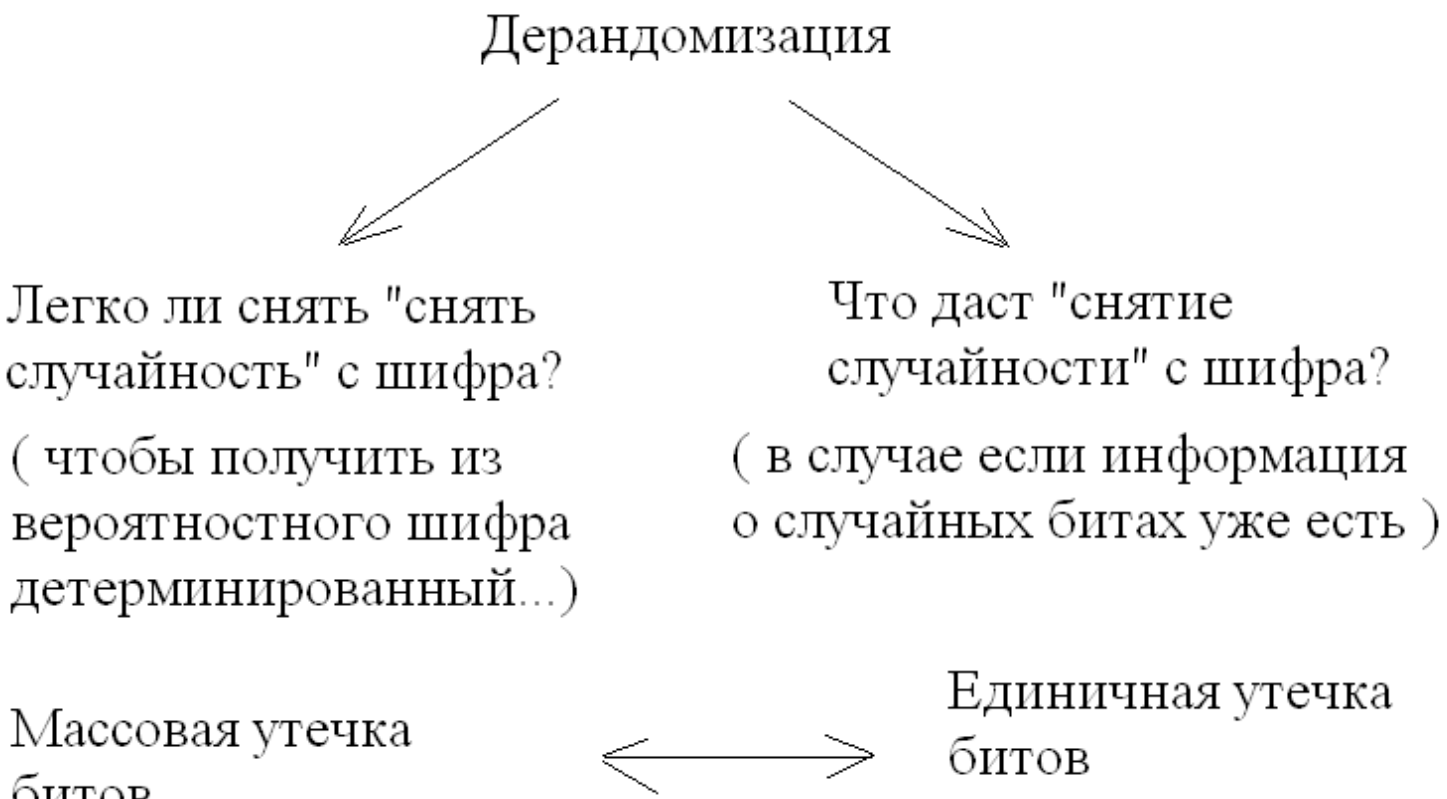
Частичная дерандомизация

«Заморозим» только часть случайных битов, подававшихся на вход алгоритма Enc криптосистемы $\mathcal{E}$, получим другую вероятностную криптосистему $\mathcal{E}''$ в которой «меньше случайного».
Соответственно, если можно полиномиальным алгоритмом «уменьшить случайность», то будем говорить о нестойкости к частичной дерандомизации.

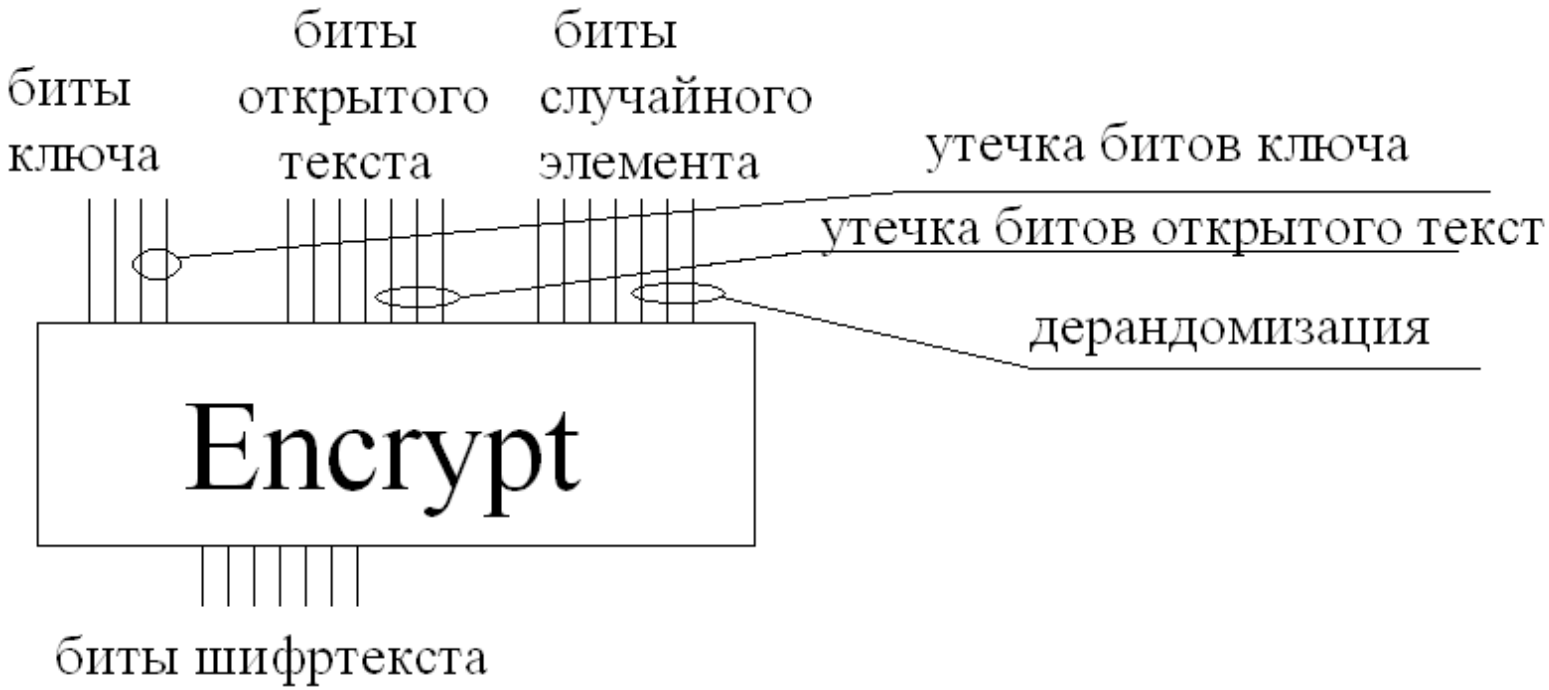
Пример (нестойкой к) дерандомизации гомоморфной криптосистемы

Пусть открытый текст m – элемент кольца, ему ставится в соответствие полином (элемент случайности), который затем переводится в полином-шифртекст некоторым гомоморфизмом полиномиальных колец (этот гомоморфизм является секретным ключом).
Известно, что алгоритм декомпозиции системы полиномов является полиномиально эффективным, следовательно такой шифр будет нестойким к дерандомизации.

Дерандомизация



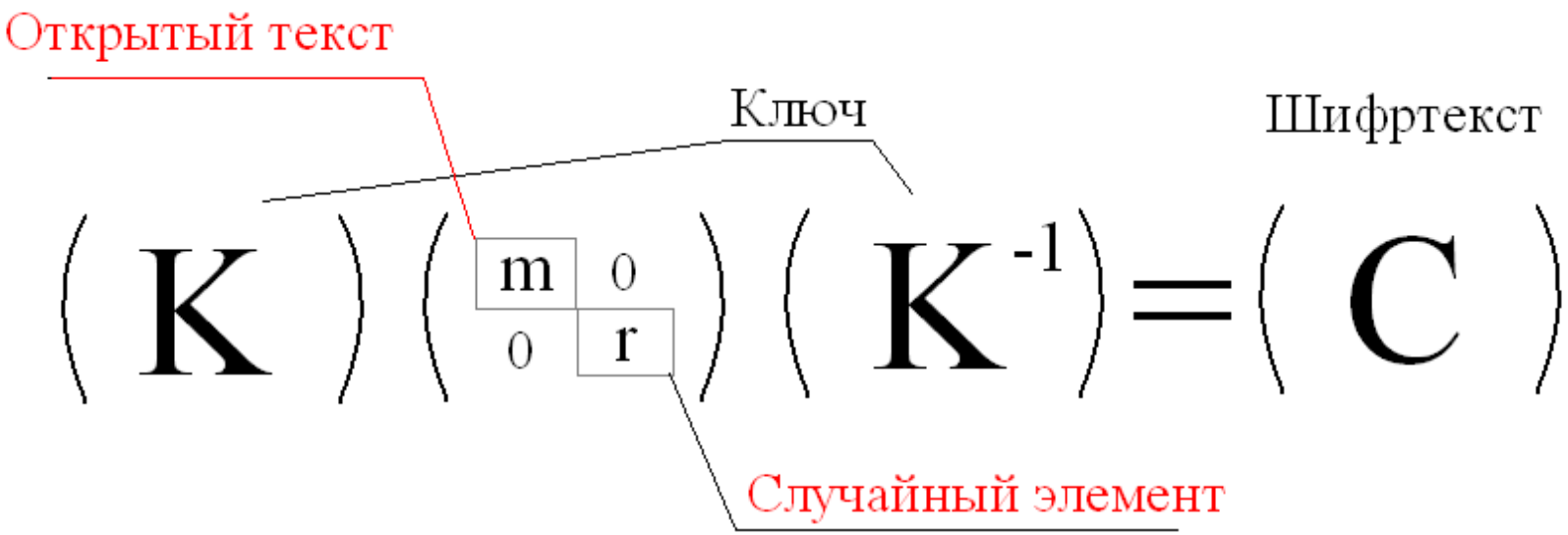
Варианты утечек информации в вероятностной криптосистеме



Второе определение дерандомизации

Дерандомизация – это массовая (для всех известных криптоаналитику шифртекстов открываются случайные биты с фиксированных позиций, которые использовались при зашифровании) утечка случайных битов.
А криптосистема нестойкая к дерандомизации – которая взламывается при наличии такой утечки битов.

Криптосистема MORE



Открытый текст и случайный элемент входят при шифровании симметрично (оба – собственные числа) $\Longrightarrow$ Для данной криптосистемы дерандомизация эквивалентна утечке битов данных

Криптоанализ: система сравнений

$$\mathbf{K}^{-1} \cdot \mathbf{D}_i \cdot \mathbf{K} = \mathbf{C}_i \qquad (1)$$

$$\mathbf{K} = \begin{pmatrix} k_1 & k_2 \\ k_3 & k_4 \end{pmatrix}, \mathbf{D}_i = \begin{pmatrix} m_i & 0 \\ 0 & r_i \end{pmatrix}, \mathbf{C}_i = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix}$$

Криптоанализ: система сравнений

$$\begin{cases} a_1x_1 + b_1x_2 - m_1x_1 = 0 \\ c_1x_1 + d_1x_2 - m_1x_2 = 0 \\ a_2x_1 + b_2x_2 - m_2x_1 = 0 \\ c_2x_1 + d_2x_2 - m_2x_2 = 0 \\ \dots \\ a_Nx_1 + b_Nx_2 - m_Nx_1 = 0 \\ c_Nx_1 + d_Nx_2 - m_Nx_2 = 0 \\ x_1x_2 - x_1 - x_2 + 1 = 0 \end{cases}$$

где (x_1, x_2) – искомый с.в. а m_i – с.ч.,

Криптоанализ: система сравнений

Рассмотрим такой простой пример (все вычисления происходят в $\mathbb{Z}_{2147483647}$):
пусть у криптоаналитика есть четыре матрицы шифртекста

$$\mathbf{C}_0 = \begin{pmatrix} 1820657886 & 832737699 \\ 910898858 & 1756518807 \end{pmatrix} \mathbf{C}_1 = \begin{pmatrix} 335409971 & 940241364 \\ 927842574 & 196446533 \end{pmatrix}$$

$$\mathbf{C}_2 = \begin{pmatrix} 1075192797 & 1902511881 \\ 728461503 & 1636291009 \end{pmatrix} \mathbf{C}_3 = \begin{pmatrix} 1117471643 & 1285026887 \\ 598405443 & 221889721 \end{pmatrix}$$

Криптоанализ: система сравнений

$$\begin{cases} 1820657886x_1 + 832737699x_2 - m_0x_1 = 0 \\ 910898858x_1 + 1756518807x_2 - m_0x_2 = 0 \\ 335409971x_1 + 940241364x_2 - m_1x_1 = 0 \\ 927842574x_1 + 196446533x_2 - m_1x_2 = 0 \\ 1075192797x_1 + 1902511881x_2 - m_2x_1 = 0 \\ 728461503x_1 + 1636291009x_2 - m_2x_2 = 0 \\ 1117471643x_1 + 1285026887x_2 - m_3x_1 = 0 \\ 598405443x_1 + 221889721x_2 - m_3x_2 = 0 \\ x_1x_2 - x_1 - x_2 + 1 = 0 \end{cases}$$

Крипоанализ: приведенная система сравнений

Базис Грёбнера этой системы даст систему уравнений

$$\begin{cases} m_3^2 + 808122283 m_3 - 618545452 = 0 \\ m_2 + 1071359822 m_3 + 263024109 = 0 \\ m_1 - 595218519 m_3 + 255496385 = 0 \\ m_0 - 169180587 m_3 + 2760400 = 0 \end{cases}$$

Крипоанализ: добавление в систему неравенств

Решение системы симплекс-методом

$$m_i < m_j \Rightarrow m_i - m_j < 2147483647$$

$$\begin{cases} m_3^2 + 808122283 m_3 - 618545452 = 0 \\ m_2 + 1071359822 m_3 + 263024109 = 0 \\ m_1 - 595218519 m_3 + 255496385 = 0 \\ m_0 - 169180587 m_3 + 2760400 = 0 \\ m_2 - m_1 < 2147483647 \\ m_1 - m_0 < 2147483647 \\ m_0 - m_3 < 2147483647 \end{cases} \Rightarrow \begin{cases} m_0 = 913328479 \\ m_1 = 454173540 \\ m_2 = 425765409 \\ m_3 = 1602984932 \end{cases}$$

Выводы и заключение

- Обнаружен критерий безопасности использования гомоморфной криптосистемы в луковичной архитектуре системы защиты СУБД.
- Установлена небезопасность использования криптосистем MORE и PORE в этой ситуации.

Спасибо за внимание!