# Метод обеспечения безопасности данных на серверах ЦОД

**Студент 5 курса ОмГТУ Сагайдак Д.А. научный руководитель д.т.н, профессор Файзуллин Рашит Тагирович**

# Что такое ЦОД?

- ЦОД — специализированное здание для размещения (хостинга) серверного и коммуникационного оборудования и подключения к каналам сети Интернет.

- Дата-центр исполняет функции обработки, хранения и распространения информации, как правило, в интересах корпоративных клиентов. ЦОД ориентирован в первую очередь, на решение бизнес-задач, путем предоставления услуг в виде информационных сервисов. Консолидация вычислительных ресурсов и средств хранения данных в ЦОД позволяет сократить совокупную стоимость владения IT-инфраструктурой за счет возможности эффективного использования технических средств, например, перераспределения нагрузок для оптимального решения бизнес-задач, а также за счет сокращения расходов на администрирование.

# Предлагаемый метод

Представляется возможным предложить решение вопроса конфиденциальности хранимой информации с помощью **схемы разделения секрета**, когда клиент центра обработки данных передает на хранение данные, но не информацию, как таковую. Простейшим примером такого рода может служить передача в ЦОД набора бит, зашифрованного однообразным ключом, длина которого больше или равна длине набора передаваемых бит.

В качестве основного алгоритма можно рассмотреть процедуру, её суть заключается в том, что на начальном этапе имеется объем информации, который необходимо хранить на серверах ЦОД, для этого объема формируется функция F, которая является набором определённых действий. Функция F может принимать следующие значения:

- осуществление инверсии файла;
- осуществление разбиения и перестановки (запоминается место разбиения и пропорции, в которых осуществляется разбиение).
- осуществление выборки некоторой части информации (запоминается место, откуда осуществлялась выбора и часть информации, которую извлекли);

Эти действия могут выполняться как по отдельности, так и совместно.

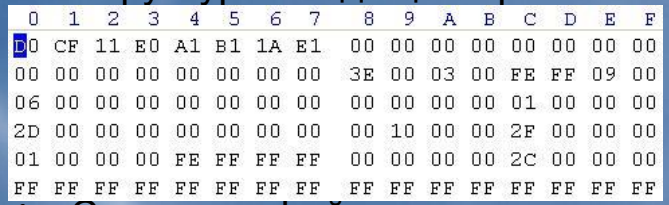
После того, как сформирована функция F, осуществляется сохранение преобразованного файла на сервер ЦОД, а функция F хранится у пользователя.

При обращении к той или иной части информации, которая хранится на сервере ЦОД, осуществляется восстановление этой информации при помощи функции $F^{-1}$.

# Осуществление инверсии

Для быстроты осуществления инверсии входящий файл представляется в виде HEX (шестнадцатеричного кода).

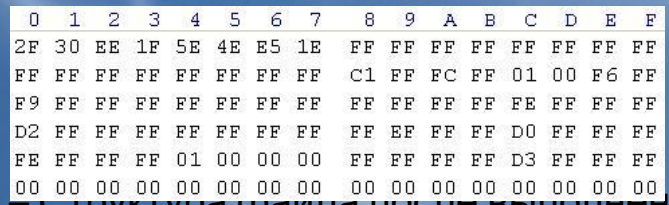
На рисунке 1 приведена часть структуры входящего файла в виде HEX.



| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D0 | CF | 11 | E0 | A1 | B1 | 1A | E1 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 3E | 00 | 03 | 00 | FE | FF | 09 | 00 |
| 06 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 |
| 2D | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 10 | 00 | 00 | 2F | 00 | 00 | 00 |
| 01 | 00 | 00 | 00 | FE | FF | FF | FF | 00 | 00 | 00 | 00 | 2C | 00 | 00 | 00 |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |

Рисунок 1 – Структура файла до выполнения инверсии.

Процедура инверсии осуществляется с помощью функции логического отрицания (функция NOT). На рисунке 2 приведена часть структуры входящего файла в виде HEX после выполнения логической функции NOT.



| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2F | 30 | EE | 1F | 5E | 4E | E5 | 1E | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | C1 | FF | FC | FF | 01 | 00 | F6 | FF |
| F9 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FE | FF | FF | FF |
| D2 | FF | FF | FF | FF | FF | FF | FF | FF | EF | FF | FF | D0 | FF | FF | FF |
| FE | FF | FF | FF | 01 | 00 | 00 | 00 | FF | FF | FF | FF | D3 | FF | FF | FF |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Рисунок 2 – Структура файла после выполнения инверсии.

Вследствие выполнения данной процедуры, формируются файл с выполненной инверсией, который помещается на сервер ЦОД, и ключ, который хранится у пользователя, указывающий на то, что для восстановления файла необходимо осуществить инверсию.

# Осуществление разбиения и перестановки частей файла

Разбиение файла может осуществляться на две части различного размера (размер указывает пользователь), после разбиения файла, части меняются местами.

На рисунке 3 изображена процедура разбиения файла.

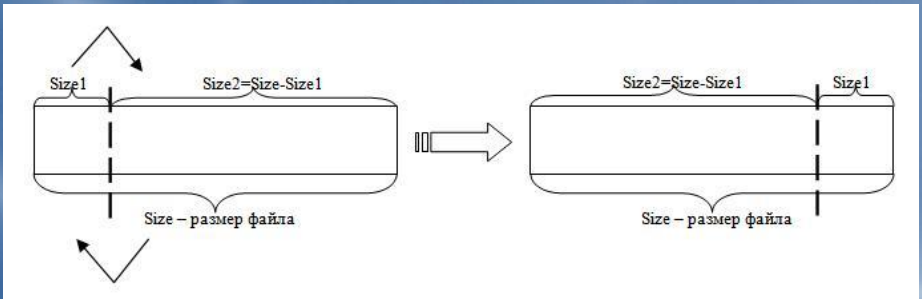



Рисунок 3 – Разбиение и перестановка частей файла местами.

Вследствие выполнения данной процедуры формируются файл с произведённым разбиением и перестановкой, который помещается на сервер ЦОД, и ключ, хранящийся у пользователя и указывающий на то, что необходимо произвести разбиение файла в определённом размере и перестановку частей.

# Осуществление изъятия определённых частей из файла

Для осуществления изъятия определённых частей из файла, пользователь указывает количество частей, которые хочет изъять, также задаёт размер каждой части, т.е. с какого и до какого байта осуществлять выемку.

Например, осуществим выборку двух частей, одна – с k1 до k2, вторая – с k3 до k4. На рисунке 4 изображена выемка частей из файла.

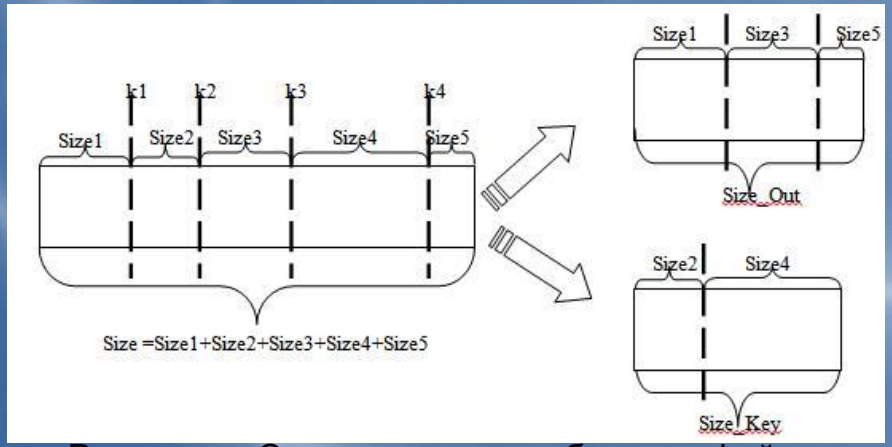



Рисунок 4 – Осуществление выборки из файла.

Вследствие выполнения данной процедуры формируется файл размером «Size_Out» - файл, который помещается на хранение на сервер ЦОД, также формируется файл размером «Size_Key» - изъятые части из входящего файла, и файл key.dat, в котором указывается, что произведена выборка определённого числа частей и размерности частей.

# Оценка эффективности и скорости

Для оценки скорости преобразования будут использоваться файлы различной длины, персональные компьютеры различной комплектации, а также различная последовательность предложенных выше действий.

В таблице приведены результаты преобразования файлов, оценка осуществлялась на двух компьютерах. Частей для изъятия во всех входящих файлах: N=2, размер первой части: с 3 до 10000, второй части

| Размер файла (Байт) | Инверсия (мс) | Инверсия, разбиение (мс) | Инверсия, изъятие (мс) | Инверсия, разбиение, изъятие (мс) | Разбиение (мс) | Разбиение, изъятие (мс) | Изъятие (мс) |
|---|---|---|---|---|---|---|---|
| **Технические характеристики компьютера №1: Процессор: DualCore Intel Pentium T2370 1,73 ГГц. Объем оперативной памяти: 2ГГб.** | | | | | | | |
| **972322** | 6093 | 6438 | 6499 | 6797 | 251 | 469 | **281** |
| **1737772** | 11016 | 11374 | 11313 | 11704 | 281 | 656 | **374** |
| **4950771** | 31889 | 32561 | 32422 | 33030 | 656 | 1359 | **735** |
| **11266885** | 71124 | 72029 | 72311 | 73700 | 1358 | 2813 | **1500** |
| **67039267** | 415561 | 428376 | 426795 | 432982 | 8562 | 16327 | **7750** |
| **Технические характеристики персонального компьютера №2: Процессор: DualCore Intel Core 2 Duo E6550 1,87 ГГц. Объем оперативной памяти: 2ГГб.** | | | | | | | |
| **972322** | 4438 | 4656 | 4640 | 4733 | 171 | 344 | **203** |
| **1737772** | 7953 | 8016 | 8014 | 8264 | 186 | 390 | **233** |
| **4950771** | 21125 | 21374 | 21514 | 21748 | 312 | 687 | **360** |
| **11266885** | 44891 | 46641 | 45593 | 46077 | 594 | 1185 | **624** |
| **67039267** | **258565** | **269778** | **266961** | **278141** | **4515** | **12186** | **4703** |

# Оценка эффективности и скорости

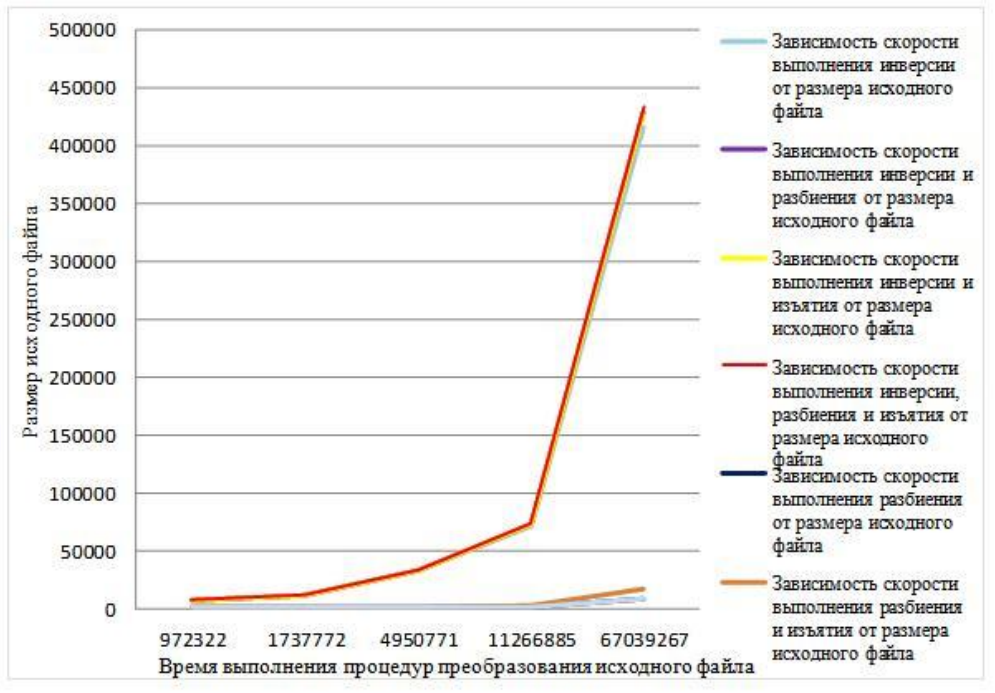



Рисунок 5 – Зависимость скорости выполнения последовательности операций от размера исходного файла на компьютере №1.

На выполнение операции инверсии требуется больше времени, чем на операции разбиения, изъятия, или их вместе взятых. Это объясняется тем, что осуществляется инверсия каждого байта входящего файла (в один байт помещается две шестнадцатеричные цифры). Как показывает численный эксперимент эффективно можно работать с файлами размером меньше 20 МБ.