Однако, всё не так однозначно, когда речь заходит о передаче файлов между компьютерами в сети: не у каждого офисного работника хватит навыков открыть общий доступ к нужной папке для определенного набора пользователей. Что уж говорить о случаях, когда по каким-то причинам устройство скрыто для остальных устройств сети из-за некорректных параметров сетевого адаптера в разделе настроек операционной системы. Наверняка, кто-то скажет, что для этих целей можно использовать внешние накопители. Однако, такое предложение перестаёт быть решением, когда размер файла достаточно велик, либо файл нужно передать большому количеству устройств в максимально короткие сроки. Кроме того, не всегда безопасно использовать внешние накопители, ведь неисправное устройство может повредить и сам компьютер, что приведёт к его неработоспособности.

В связи с перечисленными проблемами возникла идея создания приложения, которое позволило бы делиться файлами в локальной сети быстро и просто. Кроме того, одной из ключевых особенностей такого приложения должна быть простота и удобство использования.

Приложение должно предоставлять возможность отправки и приёма файлов в локальной сети, возможность отменить загрузку или отправку файлов, а также возможность восстановить процесс загрузки или передачи после сбоя в сети.

Для разработки был выбран следующий набор технологий:C#,WindowsPresentationFoundation,SQLite, Akka.net[1].

ЛИТЕРАТУРА

1. Akka.Net[Электронный ресурс]–Режим доступа: https://getakka.net/ - Дата доступа: 25.03.2020.

УДК 557.114:616-006

Студ. К.Д. Цыбулько
Науч. рук. зав. кафедрой Н.В. Пацей
(кафедра программной инженерии, БГТУ)

ОЦЕНКА СЛОЖНОСТИ ВЗЛОМА СКРЭМБЛЕРА FLASHNAND ПАМЯТИ

Возникновение ошибок во флэш-памяти во многом зависит от данных, хранящихся в ячейках, и от работы различных приложений [1]. Для уменьшения зависимости частоты ошибок от данных и предотвращения записи данных вредоносными приложениями, контроллер скремблирует (scrambling) данные перед записью. Основная концепция скремблирования заключается в генерации псевдослучайной последова-

тельности нулей и единиц, для того чтобы минимизировать любые ошибки, которые зависят от данных. Реализация скремблирования выполняется с использованием линейного сдвигового регистра с обратной связью (Linear Feedback Shift Register, LFSR)[2].

Для создания системы защиты от вредоносных программ необходимо воссоздать путь взлома скрэмблера. Наиболее оптимальным является использование алгоритма Берлекемпа –Месси.

На вход алгоритма подается битовая последовательность $X_n = \{x_0, x_1, \dots, x_{n-1}\}$ длины n. Алгоритм выполняет n итераций, на каждой итерации, определяя минимальный многочлен и линейную сложность подпоследовательности из первых битов последовательности. На выходе алгоритма – минимальный многочлен всей последовательности

$$F(\lambda) = 1 + a_1\lambda + a_2\lambda^2 + a_3\lambda^3 + \cdots + a_L\lambda^L$$

значение ее сложности $L(X_n), 0 \le L(X_n) \le n$.

Алгоритм Берлекемпа–Месси:

1.Задать $F(\lambda) = 1; \quad L = 0; \quad m = -1; G(\lambda) = 1; \quad N = 0.$

2. Пока $N < n$, выполняются следующие шаги:

- Вычисление разности между двумя последовательными состояниями генератора $d = \left(x_N + \sum_{i=1}^{L} a_i\, x_{N-i}\right) mod 2;$
  - Если $d = 1$, то
    - $T(\lambda) = F(\lambda); \quad T(\lambda) = F(\lambda) + G(\lambda) \cdot \lambda^{N-m};$
    - Если$L \le N/2$, то $L = N + 1 - L; \quad m = N; \quad G(\lambda) = T(\lambda)$
  - $N = N + 1.$

Сложность алгоритма $O(n^2)$, гдеn – длина последовательности.

Для увеличения надежности скремблера возможно использование полиномов большей степени, использование генератора Геффе.

ЛИТЕРАТУРА

1.YuCai, SaugataGhose, ErichF. Haratsch, YixinLuo, OnurMutlu. Error Characterization, Mitigation and Recovery in Flash-Memory-Based Solid-State Drives // Proceedings of the IEEE, September 2017.

2.J.P. van Zandwijk. A mathematical approach to NAND falsf-memory descrambling and decoding //Digital Investigation. – 12. – 2015.