РАЗРАБОТКА МНОГОПОЛЬЗОВАТЕЛЬСКОГО ПРИЛОЖЕНИЯ ДЛЯ БЕЗОПАСНОГО ОБМЕНА СООБЩЕНИЯМИ И ДОКУМЕНТАМИ С ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСЬЮ НА ОСНОВЕ ПРОТОКОЛА ФЕЙГЕ-ФИАТА-ШАМИРА

Н. И. Сенькевич

kosensen@yandex.by;
Научный руководитель — И. С. Войтешенко, кандидат технических наук, доцент

Протоколы нулевого разглашения, несомненно, – будущее конфиденциальности. Данные протоколы позволяют организациям гарантировать безопасные транзакции и операции, аутентификацию, авторизацию и учет, не раскрывая никакой личной информации. В этой статье представлена реализация многопользовательского приложения, позволяющего пользователям безопасно обмениваться информацией при помощи протокола Фейге-Фиата-Шамира и схемы электронной цифровой подписи (ЭЦП) на его основе.

Ключевые слова: протокол нулевого разглашения; ключ; аутентификация; электронная цифровая подпись.

ВВЕДЕНИЕ

Протоколы нулевого разглашения основаны на использовании интерактивных систем проверки. В процессе реализации такой системы проверки один пользователь (доказывающий) может убедить другого (проверяющего) в знании секрета, при этом не раскрывая самого секрета. Таким образом, в процессе выполнения протокола не происходит какой-либо утечки конфиденциальной информации.

Стандартное представление протоколов нулевого разглашения состоит из трех основных элементов между *А* и *Б.* Эти элементы называются *свидетельство, вызов* и *ответ.* На них основываются следующие три типовых шага, из которых состоят многие протоколы нулевого разглашения:

2.1 Генерация доказывающим разового случайного секретного ключа и вычисление по нему разового открытого ключа, который передается проверяющему;

2.2 Генерация проверяющим равновероятного бита запроса и направление его доказывающему;

2.3 Вычисление доказывающим ответа и направление его проверяющему.

Далее подробнее рассмотрим принципы работы разработанного приложения.

ГЕНЕРАЦИЯ МОДУЛЯ

Все вычисления происходят по большому составному модулю n, который является произведением двух больших простых чисел p и q. Значения простых чисел хранятся на сервере и не разглашаются, значение модуля передается всем пользователям при подключении к серверу.

Для безопасности и актуальности программы генерируются такие p и q, что размер, полученного от их произведения числа, составляет 2048 бит. Для генерации больших простых чисел используется алгоритм, основанный на постулате Бертрана, и тест Миллера-Рабина.

АУТЕНТИФИКАЦИЯ

При запуске клиентской части приложения открывается окно, содержащее поле для ввода логина. После ввода логина в соответствующее поле и его подтверждения происходит отправка запроса на сервер о допуске в систему. Если логин является новым и не хранится в списке логинов на стороне сервера, происходит запрос к администратору на добавление нового пользователя. После подтверждения происходит генерация секретного ключа $(s_1, s_2, s_3, \dots, s_k), 1 \le s_i \le n-1, (s_i, n) = 1, i = \overline{1,k}$ и открытых ключей логина $v_i = s_i^2 mod\ n, i = \overline{1,k}$ и подписи $v_i = s_i^{-2} mod\ n, i = \overline{1,k}$. Секретный ключ передается отправителю запроса на подключение и не разглашается. Открытый ключ логина хранится на сервере, а открытый ключ подписи рассылается всем подключенным пользователям.

При повторном подключении пользователя происходит процесс аутентификации на основе протокола Фейге-Фиата-Шамира. Безопасность данного протокола строится на трудно разрешимой задаче нахождения квадратного корня по составному модулю $n = pq$, p и q – большие простые числа, которые не разглашаются.

Протокол включает следующие три шага:

1. Доказывающий (пользователь) случайным образом выбирает в качестве разового секретного ключа число $r, 1 \le r \le n-1$, и вычисляет значение фиксатора $x = r^2 mod\ n$, после чего отправляет его проверяющему (серверу);
2. Проверяющий посылает в ответ доказывающему случайную k-битовую строку $E = (e_1, e_2, e_3, \dots, e_k)$, $e_i = \{0,1\}, i = \overline{1,k}$;
3. Доказывающий вычисляет значение $y = r \prod_{i=1}^{k} s_i^{e_i}\ mod\ n$ и отправляет его проверяющему. Проверяющий считает ответ верным, если выполняется соотношение $y^2 \equiv x \prod_{i=1}^{k} v_i^{e_i}\ mod\ n$.

Вероятность обмана проверяющего в данном протоколе будет равна $\frac{1}{2^k}$. Данный протокол не требует такого количества вычислений, как, например, протокол, использующий криптосистему RSA, и поэтому отлично подходит для маломощных процессоров.

При успешной аутентификации пользователь отправляет имена файлов открытых ключей, хранимых на устройстве и значения хеш-функций от их содержимого на сервер. Полученные данные сравниваются с данными, хранимыми на сервере. Пользователю отправляются все файлы подписи, которых у него нет, и значения хеш-функций которых не совпадает с теми, что хранятся на сервере.

В обоих случаях, после допуска пользователя к системе, окно с полем для ввода логина закрывается и открывается основное окно приложения. В любой момент времени пользователем может быть отправлен запрос на генерацию и рассылку новых ключей.

СОЗДАНИЕ И ПРОВЕРКА ПОДПИСИ

При выборе отправляемого файла необходимо выбрать пользователя, которому будет отправлен исходный файл и подпись к нему. После подтверждения отправки файлов на сервер поступает запрос на предоставление адреса отправляющего принимающему и создание подключения между ними. После подключения к отправителю принимающего последнему выводится оповещение для подтверждения или отказа от отправки. В случае согласия отправителем создается подпись к выбранному файлу. Для этого генерируется случайное число $r, 1 \leq r < n-1$ в качестве разового секретного ключа. Вычисляется значение фиксатора $u = r^2 mod\, n$. К содержимому выбранного файла присоединяется фиксатор и от полученного сообщения вычисляется хеш-функция E. После этого вычисляется вторая часть подписи:

$$s = r \prod_{i=1}^{k} s_i^{e_i} mod\, n \qquad (1)$$

После того, как исходный файл и файл подписи переданы, соединение завершается до следующей отправки файлов, а получатель автоматически проверяет полученную подпись. Проверка подлинности происходит следующим образом:

- Вычисляется значение $w = s^2 \prod_{i=1}^{k} v_i^{e_i} mod\, n$;

- К содержимому выбранного подписанного файла присоединяется значение w и от полученного сообщения вычисляется хеш-функция E'.

Сравниваются значения хеш-функций. Если $E = E'$, подпись признается подлинной.

В зависимости от результата получателю выводится соответствующее сообщение. Данная схема ЭЦП отличается сравнительно малой сложностью процедуры создания и проверки подписи, что позволяет переносить эти процессы даже на маломощные устройства.

Библиографические ссылки

1. *Агиевич, С.В.* Генерация простых / *С.В. Агиевич* // Криптографические методы [Электронный ресурс]. Режим доступа: https://apmi.bsu.by/assets/files/agievich/primes.pdf. – Дата доступа: 11.05.2023.
2. *Молдовян, А.А.*, Протоколы аутентификации с нулевым разглашением секрета: учеб. пособие / *А.А. Молдовян, Д.Н. Молдовян, А.Б.* Левина. СПб.: Университет ИТМО, 2016. 55 с.
3. Простое число [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/Простое_число#Алгоритмы_поиска_и_распознавания_простых_чисел. – Дата доступа: 10.05.2023.
4. Тест Миллера-Рабина [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Тест_МиллераРабина. Дата доступа: 10.05.2023.
5. *Menezes, A.J.* Handbook of Applied Cryptography / *A.J. Menezes,* P.C. van Oorschot, S.A. Vanstone. Florida: CRC Press, 1997. 780 p.