СЕКЦИОННОЕ ЗАСЕДАНИЕ «ВОПРОСЫ СТАНДАРТИЗАЦИИ В ОБЛАСТИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ»

КРИПТОГРАФИЧЕСКИЕ СТАНДАРТЫ РЕСПУБЛИКИ БЕЛАРУСЬ И ИХ РАЗВИТИЕ

С.В. АГИЕВИЧ, Д.В. ЕРМОЛИЦКИЙ, С.В. КУТУЗОВ, Д.В. НЕСТЕРОВ, О.В. СОЛОВЕЙ, Ю.С. ХАРИН
Белорусский государственный университет
НИИ прикладных проблем математики и информатики
Оперативно-аналитический центр при Президенте Республики Беларусь

Существующая система криптографических стандартов

В Республике Беларусь с 1999 года ведутся работы по разработке технических нормативных правовых актов (ныне государственных стандартов и предварительных стандартов, ранее руководящих документов Национального банка), устанавливающих требования к компонентам криптографической «инфраструктуры» (см. [1]). В настоящее время в республике действует более 20 государственных стандартов в данной области (+ 2 предварительных стандарта и проект руководящего документа). Данные стандарты определяют требования к алгоритмам криптографических преобразований и протоколам (алгоритмам шифрования и режимам его использования — ГОСТ 28147–89, СТБ 34.101.31–2011, процедурам выработки и проверки электронной цифровой подписи — СТБ 1176.2–99, СТБ 34.101.45–2013, функции хэширования — СТБ 1176.1–99, СТБ 34.101.31–2011, протоколу TLS — СТБ 34.101.65–2014), управления криптографическими ключами (алгоритмы генерации псевдослучайных чисел — СТБ 34.101.47–2012, формат запроса на издания сертификата открытого ключа — СТБ 34.101.17–2012, формат сертификата открытого ключа и списка отозванных сертификатов — СТБ 34.101.19–2012, алгоритм разделения секрета — СТБ 34.101.60–2014, протоколы формирования общего ключа на основе эллиптических кривых — СТБ 34.101.66–2014). Также определены требования безопасности к программным, программно-аппаратным и техническим средствам криптографической защиты информации — СТБ 34.101.27–2011 и СТБ П 34.101.43–2009. Установлены требования к формату карточки открытого ключа и политике применения сертификатов удостоверяющего центра.

Развитие системы криптографических стандартов

В 2016 году в Республике Беларусь планируется расширить действующую систему стандартов в области криптографической защиты информации: ввести в действие СТБ 34.101.77 «Информационная технология и безопасность. Алгоритмы хэширования» и дополнить СТБ 34.101.47 «Информационная технология и безопасность. Алгоритмы генерации псевдослучайных чисел». В первом стандарте будет определено семейство алгоритмов хэширования, во втором будут дополнительно введены алгоритмы генерации одноразовых паролей. Опишем стандартизируемые алгоритмы.

Алгоритмы хэширования

В информационных системах республики для управления электронными цифровыми подписями (ЭЦП) используется СТБ 34.101.45 «Информационные технологии и безопасность. Алгоритмы электронной цифровой подписи на основе эллиптических кривых». Этот стандарт определяет три уровня стойкости: $l = 128$, $l = 192$ и $l = 256$. Для того чтобы поддержать уровень l, требуется использовать функцию хэширования с 2l-битовыми значениями. СТБ 34.101.31 определяет 256-битовую хэш-функцию belt-hash, но 384- и 512-битовые функции хэширования пока в нашей стране не стандартизованы. Отметим, что ЭЦП на высоких уровнях стойкости востребованы уже сейчас. Например, сверхнадежные подписи нужны для построения долговременных архивных хранилищ.

Для поддержки ЭЦП на уровнях $l = 192$ и $l = 256$ было решено разработать семейство алгоритмов хэширования. Это семейство получило название Bash (Hash с заменой H на B).

Алгоритмы Bash строятся по схеме sponge (губка), предложенной Г. Бертони, Дж. Дэменом, М. Питерсом и Ж. Ван Ашем в [2]. Базовым композиционным элементом Bash является шаговая функция Bash-f. Эта функция обновляет хэш-состояние по очередному блоку обрабатываемых данных. Функция строится с помощью логических операций ($\neg$, $\wedge$, $\vee$) над 64 разрядными словами, циклических сдвигов этих слов и поразрядных и исключающих сложений ($\oplus$). Таким образом, функция относится к классу LRX (Logical+Rotation+Xor).

Выбранная платформа sponge + LRX была с успехом применена в алгоритмах хэширования Keccak (SHA-3), которые были стандартизированы в США в 2015 году в FIPS 202. Считая семейство Keccak творческой удачей и с уважением относясь к многочисленным криптографическим находкам его разработчиков, мы, тем не менее, решили продолжить традицию национальной криптографии и разработать собственное семейство. По нашим оценкам, производительность Bash близка к производительности Keccak. Текущие гарантии стойкости, выраженные в оценках снизу для числа активных S-блоков, для Bash и Keccak также сопоставимы.

Алгоритмы Bash работают не только на уровнях $l = 192$ и $l = 256$, но и на других уровнях из множества $\{16, 32, 48, \ldots, 256\}$. Алгоритм уровня l обрабатывает данные блоками из 1536–4l битов и возвращает хэш-значение длины $2l$. Принципы построения алгоритмов описаны в [3].

Алгоритмы генерации одноразовых паролей

В массовых информационных системах для проверки подлинности клиентов перед серверами чаще всего используется парольная аутентификация. Пароль — это классический секрет, с обработкой которого может справиться обычный человек. Пароль достаточно легко запомнить, для его хранения не нужны дополнительные устройства, пароль легко ввести практически в любой программно-аппаратной среде. Но парольная аутентификация обладает и существенными недостатками. Во-первых, пользователь не способен запомнить сильный высокоэнтропийный секрет. Пароль — это всегда слабый секрет, и у противника появляется возможности угадать его. Во-вторых, парольная аутентификация теряет надежность в агрессивной среде, т.е. там, где противник может перехватывать вводимые пользователем данные.

Аутентификация может быть усилена, если пользователь применяет специальные устройства, которые способны хранить сильные секреты и проводить с ними криптографические вычисления. С помощью этих устройств можно организовать надежный криптографический протокол аутентификации. Но такой протокол, как правило, будет иметь высокую вычислительную сложность. Поэтому устройство должно обладать доста-

точно серьезными вычислительными ресурсами, что повышает его стоимость и во многих случаях делает его применение экономически неоправданным. Кроме этого, для выполнения сложного протокола устройство должно быть плотно сопряжено с информационной системой, в которой пользователь работает, а это оказывается не всегда возможным.

Выходом является упрощение криптографических вычислений и выдача их результата в виде короткого пароля с малым сроком действия. Такой пароль принято называть одноразовым (one-time password, OTP). Даже если одноразовый пароль будет скомпрометирован, компрометация затронет только конкретный сеанс пользователя или даже определенный промежуток этого сеанса.

Средства формирования одноразовых паролей получили широкое распространение в связи с массовой аутентификацией в сети Интернет. Эти средства могут быть аппаратными (пример — токен SecureID компании RSA) или программными (приложение Authenticator компании Google). Унификация средств требует стандартизации алгоритмов формирования одноразовых паролей, правил формирования их входных данных, правил кодирования паролей, правил аутентификации.

Основные шаги по стандартизации сделаны международным комитетом OATH (Open AuTHentication). Под эгидой этого комитета разработаны 3 основных стандарта: RFC 4226, RFC 6238 и RFC 6287. В стандартах определены алгоритмы HOTP, TOTP и OCRA. Алгоритмы строят пароль по общему для клиента и сервера ключу и дополнительным уникальным данным: монотонному счетчику, отметке времени, случайному запросу. Алгоритмы по сути определяют 3 типа систем аутентификации: на основе событий, на основе времени и на основе запросов.

В системах первого типа пароль меняется при возникновении определенных событий. Как правило, таким событием для клиента является любая попытка аутентификации, а для сервера — успешная попытка.

В системах на основе событий противник может предъявлять серверу пароли до тех пор, пока целевое событие не произошло и, таким образом, возникают предпосылки для перебора паролей. Для защиты от перебора сервер должен предусмотреть блокировку ввода пароля на некоторое время или до возникновения определенного события, например, разблокировки.

События для клиента и сервера могут различаться, их пароли могут разойтись, и поэтому система аутентификации нуждается в дополнительных механизмах синхронизации. В системах на основе алгоритма HOTP клиент и сервер хранят счетчики попыток аутентификации. Синхронизация состоит в том, что сервер проверяет пароль не только на текущем значении счетчика, но и на d будущих значениях. Фактически сервер проверяет будущие пароли, которые могут оказаться текущими относительно клиента. Если i-й будущий пароль оказался подходящим, то сервер восстанавливает синхронизацию, увеличивая счетчик не на 1, а на $i + 1$. Синхронизация будет полностью потеряна, если клиент выполнил более d попыток аутентификации, о которых не известно серверу.

В системах на основе времени одноразовый пароль обновляется каждые t секунд (на практике $t = 30$ или 60). Устройство должно быть снаряжено достаточно точным таймером, что может привести к значительному его удорожанию. С другой стороны, системы на основе времени имеют ряд преимуществ относительно систем на основе событий. Во-первых, возможности противника по перебору паролей ограничены — он должен успеть подобрать пароль в течение t секунд. Во-вторых, опасность рассинхронизации не столь велика, как в предыдущем случае. Для синхронизации клиенту и серверу достаточно согласовать показания времени.

В системах на основе запросов одноразовый пароль строится на основании случайного запроса сервера. Фактически одноразовый пароль превращается в ответ протокола «запрос — ответ». Таймер не нужен, но дополнительно к интерфейсу вывода пароля должен быть предусмотрен интерфейс ввода запроса. Проблемы с синхронизацией отсутствуют, но противник может перебирать пароли относительно текущего запроса сколь угодно долго. Серверу следует ограничивать число попыток ввода пароля, иначе система перестанет быть стойкой.

Итоги представленного сравнительного анализа обобщены в таблице 1.

Таблица 1

Схема генерации OTP	Аппаратные требования	Синхронизация	Надежность
на основе событий	интерфейс вывода (дисплей)	динамически по правильным паролям, возможна потеря	средняя
на основе времени	интерфейс вывода, таймер	согласования времени	высокая
на основе запросов	интерфейс вывода, интерфейс ввода (миниклавиатура)	не нужна	высокая

В СТБ 34.101.47 стандартизируются все три схемы. Сохранена максимальная совместимость с базовыми RFC. В качестве базового алгоритма генерации псевдослучайных чисел предлагается использовать hmac[belt-hash] (он определен в этом же стандарте).

ЛИТЕРАТУРА

1. Криптографические стандарты Республики Беларусь. Электронный ресурс по адресу: http://apmi.bsu.by/resources/std. Последнее обращение: 2016.04.21.
2. *Bertoni G., Daemen J., Peeters M., Van Assche G.* Cryptographic sponge functions. Version 0.1, 2011. Avail. at: http://sponge.noekeon.org/CSF-0.1.pdf.
3. *Agievich S., Marchuk V., Maslau A., Semenov V.* Bash-f: another LRX sponge function. Submitted to the 5th Workshop on Current Trends in Cryptology, 2016.

ПЕРСПЕКТИВЫ РАЗВИТИЯ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В ОБЛАСТИ ЗАЩИТЫ ИНФОРМАЦИИ

Ю.К. ЯЗОВ, С.В. СОЛОВЬЕВ, Е.В. КОЛЕСНИКОВА
ФАУ «ГНИИИ ПТЗИ ФСТЭК РОССИИ»

В настоящее время общие информационные ресурсы Республики Беларусь и Российской Федерации включают в себя более 20 крупных информационных систем, обеспечивающих деятельность Союзного государства и государств-участников Договора о создании Союзного государства в таких сферах, как военно-техническое сотрудничество и оборона, таможенный, пограничный и экспортный контроль, взаимодействие правоохранительных органов и органов юстиции, а также других сферах деятельности. В сфере совместной