УДК 004.056.55

Д. И. Сандруцкий, С. Д. Колдушко, А. И. Калько
Учреждение образования «Барановичский государственный университет», Барановичи

КРИПТОГРАФИЧЕСКИЕ СЕРВИСЫ JAVA

Введение. В настоящее время, когда компьютерные технологии нашли массовое применение, проблематика криптографии включает многочисленные задачи, которые не связаны непосредственно с засекречиванием информации. Современные проблемы криптографии включают разработку систем электронной цифровой подписи и тайного электронного голосования, протоколов электронной жеребьевки и идентификации удаленных пользователей, методов защиты от навязывания ложных сообщений и т. п. Специфика криптографии состоит в том, что она направлена на разработку методов, обеспечивающих стойкость к любым действиям злоумышленника, в то время как на момент разработки криптосистемы невозможно предусмотреть все способы атаки, которые могут быть изобретены в будущем на основе новых достижений теории и технологического прогресса.

Основная часть. Криптография за свое долгое существование прошла четыре основных этапа: наивная криптография, формальная криптография, научная криптография, компьютерная криптография.

Для наивной криптографии характерно использование любых (обычно примитивных) способов запутывания противника относительно содержания шифруемых текстов.

Этап формальной криптографии связан с появлением формализованных и относительно стойких к ручному криптоанализу шифров.

Главная отличительная черта научной криптографии — появление криптосистем со строгим математическим обоснованием криптостойкости, окончательно сформировались разделы математики, являющиеся научной основой криптологии.

Компьютерная криптография обязана своим появлением вычислительным средствам с производительностью, достаточной для реализации криптосистем, обеспечивающих при большой скорости шифрования на несколько порядков более высокую криптостойкость, чем «ручные» и «механические» шифры. Первым классом криптосистем, практическое применение которых стало возможно с появлением мощных и компактных вычислительных средств, стали блочные шифры.

В 1970-е годы был разработан американский стандарт шифрования DES (принят в 1978 году). С появлением DES обогатился криптоанализ, для атак на американский алгоритм было создано несколько новых видов криптоанализа (линейный, дифференциальный и т. д.).

В 1980-90-е годы появились совершенно новые направления криптографии: вероятностное шифрование, квантовая криптография и др. Тогда же были разработаны нефейстеловские шифры (SAFER, RC6 и др.), а в 2006 году после открытого международного конкурса был принят новый национальный стандарт шифрования США-AES [1].

Как известно, все алгоритмы шифрования можно разделить на симметричные и асимметричные. Симметричными являются те, у которых ключ для шифрования и расшифровки — один и тот же. Алгоритмы данной группы обладают относительно высокой скоростью выполнения, но имеют один большой недостаток: ключ необходимо держать в секрете, что проблематично при использовании в больших группах.

Асимметричные алгоритмы же используют разные ключи. Для шифрования данных здесь применяется публичный ключ, имея который, невозможно расшифровать или вычислить приватный ключ. Тем самым решается проблема симметричных шифров, но при этом асимметричные алгоритмы имеют высокую скорость шифрования.

Для того чтобы избавиться от недостатков обоих типов алгоритмов, используется их сочетание. При начале передачи генерируется случайный сеансовый ключ, который шифруется публичным ключом асимметричного алгоритма и далее, при получении, расшифровывается с помощью ключа приватного, после чего все передаваемые данные шифруются с помощью симметричного алгоритма с использованием ранее полученного сеансового ключа. Тем самым решается проблема передачи секретного ключа симметричного шифрования, а также проблема высокого времени выполнения асимметричного шифрования, так как шифрование сеансового ключа не занимает много времени.

К симметричным относятся такие алгоритмы, как DES, его модификация DESede (3DES), AES (Rijndael). К асимметричным относится алгоритм RSA [2].

Помимо алгоритмов шифрования существует понятие хэширования. Хэширование — преобразование массива входных данных произвольной длины в (выходную) битовую строку фиксированной длины, выполняемое определенным алгоритмом [2].

В некоторых ситуациях необходимо сгенерировать сеансовый ключ для создания канала передачи сообщений, зашифрованного симметричным ключом, однако невозможно использовать асимметричный алгоритм по причине возможности утраты приватного ключа. В таких случаях можно использовать протокол Диффи—Хеллмана — криптографический протокол, позволяющий двум или более сторонам получить общий секретный ключ, используя незащищенный от прослушивания канал связи [3].

Реализовать алгоритмы шифрования на языке Java можно с использованием двух подходов: 1) разработка, отладка и тестирование программы по опубликованным криптографическим алгоритмам, т. е. с использованием своего подхода к реализации; 2) разработка программы с использованием криптографических сервисов, предоставляемых платформой Java.

Второй подход является более оптимальным, так как позволяет значительно сократить время разработки программы.

Java Security API предоставляется в виде набора пакетов и классов, используемых для написания приложений. В их состав входят: java.security, java.security.cert, java.security.interfaces, java.security.spec, javax.crypto, javax.crypto.interfaces, javax.crypto.spec. Пакеты java.security и javax.crypto содержат классы, отображающие основные понятия криптографии, такие как шифр, сертификат, генератор случайных чисел и генератор ключей, цифровая подпись, hash-функции и т. д.

Для сравнения производительности предложенных подходов были дополнительно разработаны программные системы с использованием опубликованных криптографических алгоритмов (AES, DES, 3DES, RSA, RSA-AES).

Основное тестирование производительности выполнялось на компьютере с процессором Intel Core(TM) i5-5200U с объемом оперативной памяти 8 Гб. В качестве оперативной системы была выбрана Windows 10 Professional. В качестве объекта для тестирования алгоритмов был выбран текстовый документ, включающий содержание книги “Games of Thrones”.

На первом этапе были протестированы алгоритмы DES, DESede, AES (рисунок 1).

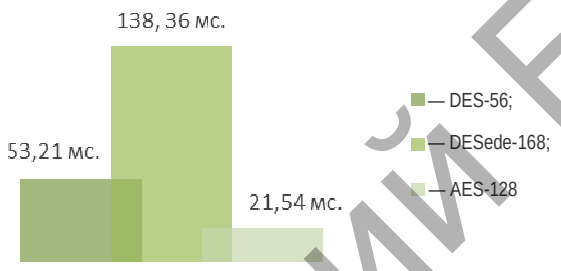

Рисунок 1 — Результаты тестирования симметричных алгоритмов

Как видно, алгоритм DES хотя и имеет 56-битный ключ, уступает по скорости алгоритму AES, который, в свою очередь, имеет 128-битный ключ. Модификация DESede, имеющая ключ, сравнимый с ключом алгоритма AES, показывает очень большое время выполнения. Можно сказать, что из всех трех алгоритмов самым защищенным и быстродействующим является алгоритм AES, так как он имеет модификацию с 256-битным ключом, тем самым превышает показатели DESede. При этом стоит отметить, что AES-256 не отличается от AES-128 по быстродействию, показывает результаты лучше, чем DES-56. Однако стойкость алгоритма AES, а также его специфика позволяют сказать, что, применяя нынешние алгоритмы криптоанализа, AES-128 вскрыть практически невозможно, а если это удастся, то AES-256 тоже будет уязвим. Отсюда следует, что оптимальным из всех симметричных алгоритмов можно назвать AES-128.

На втором этапе протестируем асимметричный алгоритм RSA. При тех же условиях, что и при тестировании симметричных алгоритмов, RSA показал «рекордное» время выполнения (23 секунды), если учесть, что на выполнение цикла шифровки и расшифровки алгоритмом AES-128 ушло 21,5 миллисекунды, что равно 0,0215 секунды. Как видно, алгоритм RSA сильно уступает в скорости алгоритму AES (рисунок 2).

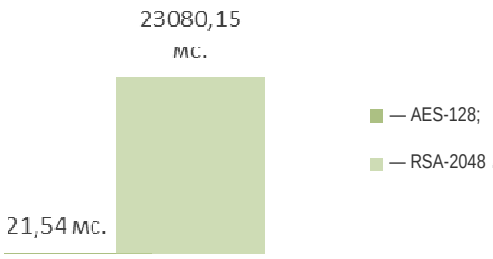

Рисунок 2 — Сравнительная характеристика времени выполнения алгоритмов AES и RSA

На третьем этапе протестируем протокол шифрования с использованием сеансового ключа. Сгенерируем сеансовый ключ, зашифруем AES-алгоритмом текст, далее зашифруем ключ RSA-алгоритмом и повторим все это в обратном порядке, расшифровав сообщение, содержащее зашифрованный текст (рисунок 3).

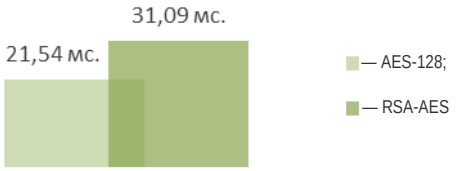

Рисунок 3 — Сравнительная характеристика времени выполнения алгоритма AES и протокола RSA-AES

Отсюда видно, что протокол RSA-AES практически не уступает скорости выполнения алгоритма AES, обеспечивая необходимую безопасность ключей, исключая необходимость передачи ключей для симметричного алгоритма «из рук в руки».

На четвертом этапе исследуем скорость выполнения алгоритмов при загруженном процессоре (рисунок 4).

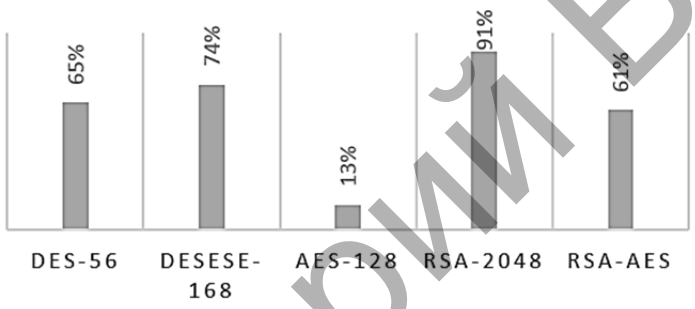

Рисунок 4 — Увеличение времени выполнения алгоритмов при загруженном процессоре

На пятом этапе исследуем, какой процент от времени всего цикла шифрования и расшифровки затрачивается отдельно на шифрование и отдельно на расшифровку.

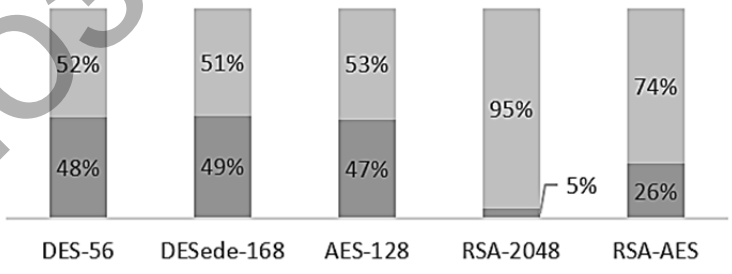

Рисунок 5 — Шифрование и расшифровка от всего времени выполнения

Как видно, в симметричных алгоритмах на шифрование и расшифровку тратится примерно одинаковое время, тогда как в асимметричных алгоритмах большая часть времени затрачивается на расшифровку.

На шестом этапе протестируем алгоритмы AES-128, RSA и протокол RSA-AES на маломощном компьютере, т. е. с процессором Intel Celeron(R) 1007U с объемом оперативной памяти 4 Гб. В качестве операционной системы была выбрана Windows 10 Home (рисунок 6).

Как показано на рисунке, происходит достаточно большое падение скорости работы алгоритма шифрования, при этом сильно выделяется асимметричный алгоритм шифрования RSA, время выполнения которого увеличилось в 725% (3 минуты) в отличие от времени выполнения на стандартном компьютере (23 секунды).

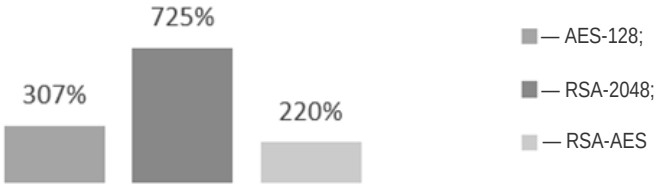

Рисунок 6 — Увеличение времени выполнения шифрования
и расшифровки при падении мощности компьютера

Заключение. Проведя тестирование распространенных алгоритмов шифрования как симметричных, так и асимметричных, мы выделили AES-алгоритм как один из оптимальных алгоритмов, обладающих как высокой скоростью, так и достаточной надежностью. Однако, как оговаривалось ранее, у него есть недостаток, который заключается в проблеме передачи ключа. Поэтому можно утверждать, что протокол RSA-AES является оптимальным, потому что включает в себя алгоритм AES, при этом лишая его главного недостатка — передачи секретного ключа. В результате исследования был разработан программный продукт, использующий стандартные библиотеки шифрования Java, а также проведены тестирования как на стандартном компьютере, так и на маломощном. Это позволяет утверждать, что сделанные выше выводы верны.

Данные исследования могут быть использованы при разработке программных продуктов, в которых необходимо реализовать защиту информации. На основе проведенных исследований будет создан программный продукт на языке программирования Java, предназначенный для передачи сообщений по зашифрованным каналам. Данный программный продукт будет использовать протокол RSA-AES, а также протокол Диффи—Хеллмана.

Список цитируемых источников

1. *Баричев, С. Г.* Основы современной криптографии / С. Г. Баричев, Р. Е. Серов, В. В. Гончаров. — М. : Горячая Линия — Телеком, 2011. — 176 с.
2. *Бернет, С.* Криптография. Официальное руководство RSA Security / С. Бернет, С. Пэйн. — М. : Бином-Пресс, 2009. — 384 с.
3. *Шнейер, Б.* Прикладная криптография / Б. Шнейер. — М. : Триумф, 2002. — 816 с.

УДК 004.657

А. В. Суряпина
Учреждение образования «Барановичский государственный университет», Барановичи

АВТОМАТИЗИРОВАННАЯ СИСТЕМА КОНТРОЛЯ ПРОХОЖДЕНИЯ ПРАКТИКИ СТУДЕНТАМИ В КОМПАНИИ “JAZZTEAM”

Введение. Проблема совершенствования процесса образования постоянно находится в центре внимания общества и государства. Одним из наиболее важных этапов учебного процесса является практика студентов на предприятиях.

Прохождение практики представляет собой планомерную и целенаправленную деятельность студентов по освоению избранной специальности, углубленному закреплению теоретических знаний, профессиональных, творческих и исполнительских навыков на каждом этапе обучения.

Целью практики является обучение студентов практическим навыкам и подготовка их к самостоятельной работе по избранной специальности. Практика должна проводиться в организациях, соответствующих профилю подготовки специалистов.

Одной из таких организаций является компания “JazzTeam” — молодая инновационная компания с офисами в Солигорске и Минске, созданная экспертами с опытом участия в проектах мирового уровня. Она является Agile-компанией, концентрируемой на всех проявлениях технологии и платформы Java (J2SE, J2EE, Android, SOA, OSGI, Automation, Open Source) и оказывающей широкий спектр инновационных услуг на ИТ-рынке.

Уже несколько лет в компании “JazzTeam” проходят практику учащиеся различных учебных заведений, в связи с этим возникла необходимость создания автоматизированной системы контроля прохождения практики студентами в данной организации.