

Особенности реализации и применения ассиметричных шифров

Вихляев Дмитрий Романович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье приводится описание реализаций и способов применения ассиметричных методов шифрования. Приведены сравнительные характеристики распространённых алгоритмов. В результате исследования будут разобраны методики генерации ключей и формул, а также способы оценки надёжности шифров.

Ключевые слова: ассиметричное шифрование, безопасность, хэш-функции

Features of implementation and application of asymmetric ciphers

Vikhlyayev Dmitry Romanovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article describes implementations and ways to use asymmetric encryption methods. Comparative characteristics of common algorithms are given. As a result of the research, methods for generating keys and formulas, as well as ways to assess the reliability of ciphers, will be analyzed.

Keywords: asymmetric encryption, security, hash functions

1 Введение

1.1 Актуальность

Асимметрические методы шифрования представляют собой класс криптографических алгоритмов, в которых используются два различных, но математически связанных ключа – публичный и приватный. Публичный ключ может свободно распространяться и использоваться для шифрования сообщений или проверки цифровых подписей, тогда как приватный ключ должен храниться в секрете и используется для дешифрования или создания подписи. Основным свойством таких методов является невозможность вычислить приватный ключ за разумное время, обладая только публичным. Необходимость существования различных алгоритмов обусловлена разнообразием требований к безопасности, производительности, совместимости и устойчивости. Одни алгоритмы более подходят для цифровых подписей, другие – для шифрования или обмена ключами. Разные применения предъявляют разные требования к размерам ключей, скорости, латентности и объёму трафика. Кроме того, наличие альтернатив повышает

криптографическую надёжность всей системы, позволяя выбирать наиболее подходящее решение в зависимости от контекста использования и угроз.

1.2 Обзор исследований

Ю.С.Иванов Т.И.Кирюшко программно реализовали метод асимметричного шифрования [1]. С.В.Мельников исследовал семантическую диссипацию информации в асимметричном шифровании [2]. П.И.Тутубалин провёл оценку криптографической стойкости асимметричных алгоритмов шифрования RSA и ГОСТ р 34.10-2001 [3]. Б.Э.Ибраимов провёл исследование асимметричного шифрования [4]. В.С.Сляднев исследовал асимметричные системы шифрования в криптографии [5].

1.3 Цель исследования

Цель исследования – привести примеры различных алгоритмов асимметричного шифрования для определения использования наиболее эффективного в конкретной задаче информационной безопасности.

2 Материалы и методы

В описываемых материалах приведены формулы шифрования, расшифровки и генерации ключей асимметричных алгоритмов. Приведены методы оценки и основные способы применения каждого алгоритма.

3 Результаты и обсуждения

Главной особенностью асимметричной криптографии является её способность обеспечивать безопасную передачу данных без предварительного обмена секретами, а также реализация цифровых подписей, которые подтверждают подлинность отправителя и неизменность содержимого сообщения. Эти функции критичны в системах электронной коммерции, цифровой идентификации, протоколах защищённой связи и цифровых валютах.

Существуют различные математические основы, на которых строятся асимметричные алгоритмы, наиболее распространённые приведены в таблице 1. Наиболее традиционные из них – RSA и алгоритмы на основе дискретного логарифма (DSA, ElGamal, Diffie-Hellman), которые зависят от вычислительной сложности факторизации и логарифмирования в конечных полях. Современные постквантовые алгоритмы, такие как Kyber, Dilithium и McEliece, опираются на более сложные задачи, устойчивые к квантовым атакам, включая задачи на решётках, кодах с исправлением ошибок, хэш-функциях и многочленах.

Различия в реализации затрагивают размеры ключей, вычислительные ресурсы и назначение. Например, алгоритмы на эллиптических кривых (ECDSA, ECDH) обеспечивают сопоставимую безопасность при значительно меньших размерах ключей и более высокой производительности по сравнению с RSA. В то же время, такие алгоритмы требуют более сложной реализации и более строгой защиты от побочных каналов. Постквантовые

методы часто имеют существенно большие размеры публичных ключей или подписей, но предлагают устойчивость к квантовым вычислениям, что делает их приоритетными для долгосрочной защиты информации.

Методы оценки эффективности и безопасности включают математический анализ стойкости задач, устойчивость к известным классическим и квантовым атакам, устойчивость к побочным каналам и соответствие международным стандартам. Также важно учитывать производительность в различных средах, таких как встроенные устройства, облачные сервисы или распределённые системы.

PQ-стойкость (пост квантовая стойкость) – это свойство криптографических алгоритмов сохранять безопасность в условиях появления квантовых компьютеров, которые способны эффективно решать некоторые задачи, на которых основаны классические алгоритмы, например, факторизацию (RSA) или дискретный логарифм (DSA, ElGamal, Diffie-Hellman).

Квантовые алгоритмы, такие как алгоритм Шора, могут полиномиально решать задачи факторизации и вычисления дискретных логарифмов, что делает уязвимыми традиционные асимметричные схемы. Пост квантовые (PQ) алгоритмы базируются на других математических проблемах, для которых не известно квантовых алгоритмов с эффективным решением. К таким проблемам относятся задачи на решётках (Lattice-based cryptography), например, Learning With Errors (LWE), Ring-LWE (Kyber, Dilithium). Коды с исправлением ошибок (Code-based cryptography), например, McEliece. Многочленное шифрование (Multivariate cryptography). Хэш-деревья (Hash-based signatures), например, SPHINCS+.

PQ-стойкие алгоритмы считаются способными противостоять атакам даже с использованием мощных квантовых компьютеров, поэтому активно разрабатываются и стандартизируются (например, NIST PQC) (табл.1).

Шифр RSA – это один из первых и наиболее широко используемых алгоритмов с открытым ключом. Он основан на сложности разложения большого числа на простые множители. Применяется для шифрования и цифровой подписи.

Генерация ключей:

1. Выбираются два простых числа p и q .
2. Вычисляется $n = p \times q$.
3. Вычисляется $\varphi(n) = (p - 1)(q - 1)$.
4. Выбирается e : $1 < e < \varphi(n)$, взаимно просто с $\varphi(n)$.
5. $d \equiv e^{-1} \bmod \varphi(n)$.

Формулы вычислений:

- шифрование: $c = m^e \bmod n$;
- дешифрование: $m = c^d \bmod n$;
- подпись: $s = H(m)^d \bmod n$;
- проверка подписи: $H(m) = s^e \bmod n$;

RSA обеспечивает безопасность за счёт сложности факторизации n , но требует больших ключей для высокой защиты.

Таблица 1. Асимметричные криптографические алгоритмы

№	Название	Основа	Назначение	Размер ключа (бит)	PQ-стойкость
1	RSA / PCA	Факторизация	Шифрование, подпись	1024–4096	Нет
2	DH / Диффи–Хеллман	Дискретный логарифм	Обмен ключами	2048–3072	Нет
3	DSA/ Подпись DSA	Дискретный логарифм	Подпись	1024–3072	Нет
4	ElGamal/ Эль-Гамаль	Дискретный логарифм	Шифрование, подпись	2048–4096	Нет
5	ECC/ Эллиптические кривые	Эллиптическая арифметика	Универсальный	224–521	Нет
6	ECDSA / Подпись ECC	Эллиптическая арифметика	Подпись	224–521	Нет
7	Kyber/ Кайбер	Решётки (MLWE)	Обмен ключами	800–1536	Да
8	Dilithium/ Дилитиум	Решётки (CRYSTALS)	Подпись	2048–3168	Да
9	NTRU	Решётки (кольца)	Шифрование	1125–2048	Да
10	McEliece/ МакЭлиса	Коды (Горра)	Шифрование	260К–1МБ	Да
11	SPHINCS+	Хэш-деревья	Подпись	16К–40К	Да

Diffie-Hellman – это протокол обмена ключами, позволяющий двум сторонам безопасно согласовать общий секретный ключ по незащищённому каналу связи. Основан на сложности вычисления дискретного логарифма в конечных группах.

Параметры: выбирается большое простое число p и генератор g порядка p .

Процесс: сторона A выбирает приватный секрет a и вычисляет публичное значение $A = g^a \bmod p$. Аналогично сторона B выбирает секрет b и вычисляет $B = g^b \bmod p$. Они обмениваются A и B .

Общий секрет: сторона A вычисляет $s = B^a \bmod p$, сторона B — $s = A^b \bmod p$. Благодаря свойствам степеней по модулю p , оба получают одинаковое значение s .

Diffie-Hellman не обеспечивает аутентификацию, но широко используется для безопасного обмена ключами в протоколах TLS, SSH.

DSA (Digital Signature Algorithm) – это стандартный алгоритм цифровой подписи, основанный на сложности задачи дискретного логарифма в конечных группах. Используется для проверки подлинности и целостности сообщений.

Параметры: выбирается большое простое число p , делитель q простого порядка (обычно 160 или 256 бит), генератор g порядка q в группе по модулю p .

Генерация ключей: приватный ключ x – случайное число из $[1, q-1]$, публичный ключ $y = g^x \bmod p$.

Подпись: для сообщения m выбирается случайное число k из $[1, q-1]$, вычисляются $r = (g^k \bmod p) \bmod q$ и $s = k^{-1} (H(m) + x \cdot r) \bmod q$, где $H(m)$ – хэш сообщения. Подпись – пара (r, s) .

Проверка подписи: вычисляются $w = s^{-1} \bmod q$, $u_1 = H(m) \cdot w \bmod q$, $u_2 = r \cdot w \bmod q$, затем $v = ((g^{u_1} \cdot y^{u_2}) \bmod p) \bmod q$. Подпись считается действительной, если $v = r$.

DSA широко используется в различных протоколах и стандартах цифровой подписи, включая федеральные стандарты США.

ElGamal – это асимметричный криптографический алгоритм, основанный на сложности вычисления дискретного логарифма в конечных группах. Используется для шифрования и цифровых подписей.

Параметры: выбирается большая простая группа G с генератором g порядка p , где p – большое простое число.

Генерация ключей: приватный ключ x – случайное число из $[1, p-1]$, публичный ключ $y = g^x \bmod p$.

Шифрование: для сообщения m выбирается случайное число k из $[1, p-1]$, вычисляются $c_1 = g^k \bmod p$ и $c_2 = m \cdot y^k \bmod p$. Шифротекст — пара (c_1, c_2) .

Дешифрование: вычисляется $m = c_2 \cdot (c_1^x)^{-1} \bmod p$.

ElGamal обеспечивает семантическую безопасность и используется в различных криптосистемах, однако размер шифр текста в два раза больше размера сообщения.

ECDSA – это алгоритм цифровой подписи, основанный на эллиптических кривых (Elliptic Curve Digital Signature Algorithm). Он обеспечивает безопасность на основе сложности задачи дискретного логарифма на эллиптической кривой, что позволяет использовать более короткие ключи по сравнению с RSA при том же уровне защиты.

Основные параметры: выбирается эллиптическая кривая над конечным полем, точка генератор G с большим порядком n , приватный ключ d — случайное число из $[1, n-1]$, публичный ключ $Q = d \cdot G$.

Подпись: для сообщения m выбирается случайное число k из $[1, n-1]$, вычисляются точки $R = k \cdot G$ с координатой x_R , и значения $r = x_R \bmod n$ и $s = k^{-1} (H(m) + d \cdot r) \bmod n$, где $H(m)$ — хэш сообщения. Подпись — пара (r, s) .

Проверка подписи: вычисляются $u_1 = s^{-1} \cdot H(m) \bmod n$ и $u_2 = s^{-1} \cdot r \bmod n$, затем точка $X = u_1 \cdot G + u_2 \cdot Q$. Подпись считается действительной, если $r = x_X \bmod n$, где x_X – координата X .

ECDSA широко используется в крипто валютных системах и протоколах безопасности благодаря эффективности и компактности ключей.

Kyber – это постквантовый алгоритм обмена ключами, основанный на проблеме обучения с ошибками в модулях решёток (Module Learning With Errors, MLWE). Он обеспечивает устойчивость к атакам квантовых компьютеров и высокую эффективность.

Параметры включают публичную матрицу A , секретный вектор s и вектор ошибок e , все в кольце многочленов с модулем q .

Генерация ключей: секретный ключ – случайный вектор s и ошибки e ; публичный ключ — $t = A \cdot s + e$.

Шифрование: для случайного сообщения m выбирается случайный вектор r и ошибка e' ; вычисляется ciphertext $c = (c_1, c_2)$, где $c_1 = A \cdot r + e'$, $c_2 = t \cdot r + e'' + \text{Encode}(m)$.

Дешифрование: с помощью секретного ключа s вычисляется $m' \approx c_2 - s \cdot c_1$, затем декодируется исходное сообщение m .

Kyber обеспечивает компактные ключи, быстрые операции и является одним из стандартов пост квантовой криптографии.

Dilithium – постквантовый алгоритм цифровой подписи, основанный на решётках и задачах поиска коротких векторов в модулях решёток (MLWE – модульная задача обучения с ошибками). Предназначен для обеспечения высокой безопасности и эффективности.

Параметры включают матрицы и векторы в кольце многочленов, с модулем q .

Генерация ключей: секретный ключ – набор коротких векторов s ; публичный ключ — вычисляется как $t = A \cdot s + e$, где A – публичная матрица, e – вектор ошибок.

Подпись: вычисляется с помощью случайного вектора u и значения $z = u + s \cdot c$, где c – хэш вызова, а также проверяется, что подпись удовлетворяет ограничениям по норме.

Проверка: публичным ключом и подписью проверяется корректность уравнения $Az = t \cdot c + u$ с учётом норм.

Dilithium отличается компактными ключами и подписями, хорошей скоростью и высоким уровнем безопасности против квантовых атак.

NTRU – это асимметричный криптографический алгоритм, основанный на теории решёток и операциях с многочленами в кольце вычетов. Используется для шифрования и обмена ключами, отличается высокой скоростью и устойчивостью к квантовым атакам.

Параметры: выбираются два многочлена f и g с небольшими коэффициентами, модуль q (обычно большое число) и модуль p (малое число, например 3).

Генерация ключей:

- Приватный ключ – многочлен f и его обратный по модулю q и p .
- Публичный ключ – $h = p * g * f^{-1} \bmod q$.

Шифрование:

Для сообщения m выбирается случайный многочлен r с небольшими коэффициентами, вычисляется $e = r * h + m \bmod q$.

Дешифрование:

Используя приватный ключ f , вычисляется $a = f * e \bmod q$, затем применяется округление и восстановление исходного сообщения m .

NTRU обеспечивает быструю работу благодаря использованию операций с многочленами и является одним из перспективных постквантовых алгоритмов.

McEliece – это асимметричный криптографический алгоритм, основанный на теории кодирования, использующий исправляющие ошибки коды (обычно коды Гоппы) для обеспечения безопасности. Основная идея — использовать затруднённую задачу декодирования случайных линейных кодов с ошибками.

Параметры: публичный ключ – зашумлённая генераторная матрица G , приватный ключ – структура кода и секретные параметры.

Шифрование: шифротекст $c = m \cdot G + e$, где m — вектор сообщения, G — публичная генераторная матрица, e – вектор ошибки с ограниченным весом.

Дешифрование: с помощью знания структуры кода и секретного ключа можно исправить ошибки в c и восстановить исходное сообщение m .

McEliece обладает высокой скоростью шифрования и устойчивостью к атакам квантовых компьютеров, однако публичный ключ довольно большой по размеру (несколько сотен килобайт).

SPHINCS+ – это пост квантовый алгоритм цифровой подписи, основанный не на арифметике (как RSA или ECC), а исключительно на криптографических хэш-функциях. Он устойчив к атакам квантовых компьютеров, потому что не зависит от математических проблем, таких как факторизация или дискретный логарифм. Статически безопасная (stateless) схема цифровой подписи, использующая массив хэш-деревьев (в основном Merkle-деревья) и одноразовые подписи (OTS — one-time signatures), такие как FORS и WOTS+.

WOTS+ – Winternitz One-Time Signature Plus используется для создания подписей над данными, используя последовательность хэшей. Приватный ключ: набор случайных битовых строк x_i . Публичный ключ: $y_i = H^{a_i}(x_i)$, где хэш применяется a_i раз к каждому элементу. Подпись: $sig_i = H^{a_i}(x_i)$ для всех i , где a_i зависит от сообщения.

FORS – Forest of Random Subsets подписывает сообщение с помощью множества одноуровневых деревьев. Представляет хэш сообщения в виде набора индексов. Для каждого индекса подписывается лист дерева и добавляется путь к корню (аутентификационный путь).

Merkle tree – объединяет подписи (FORS и WOTS+) в одно дерево, корень которого является частью открытого ключа. Корень дерева (root) – часть публичного ключа. Каждая подпись сопровождается Merkle-доказательством, то есть набором хэшей, необходимым для подтверждения пути от подписи к корню.

SPHINCS+ использует комбинацию одноразовых подписей и хэш-деревьев. Чтобы подписать сообщение, оно сначала кодируется в индексы, которые используются в одноразовых подписях (FORS), а потом эти подписи

защищаются ещё одним одноразовым уровнем (WOTS+), и вся структура собирается в дерево. Подпись включает хэши по дереву, подписи FORS и WOTS+, чтобы можно было всё проверить только по открытому корню дерева и хэш-функции.

В результате исследования были рассмотрены асимметричные методы криптографии, проведено сравнение их реализаций. Развитие и разнообразие алгоритмов обусловлено необходимостью адаптации к новым вычислительным возможностям, особенно в условиях угроз со стороны квантовых технологий.

Библиографический список

1. Иванов Ю.С., Кирюшко Т.И. Программная реализация метода асимметричного шифрования // Вестник научного общества студентов, аспирантов и молодых ученых. 2014. № 1. С. 145-154.
2. Мельников С.В. Семантическая диссипация информации при асимметричном шифровании // Техника и технология. 2012. № 3. С. 42-44.
3. Тутубалин П.И. Оценка криптографической стойкости асимметричных алгоритмов шифрования RSA и ГОСТ р 34.10-2001 // В сборнике: Татищевские чтения: актуальные проблемы науки и практики. Материалы XIV Международной научно-практической конференции. В 4-х томах. 2017. С. 45-56.
4. Ибраимов Б.Э. Асимметричное шифрование // В сборнике: Передовые инновационные разработки. Перспективы и опыт использования, проблемы внедрения в производство. Сборник научных статей по итогам Пятой международной научной конференции. 2019. С. 186-188.
5. Сляднев В.С. Исследование асимметричных систем шифрования в криптографии // Интернаука. 2024. № 18-4 (335). С. 14-16.