УДК 004.056.55 **Технические науки**

Хорошев Матвей Артемович, *студент ФДП и СПО*
ФГБОУ ВО «Мордовский государственный университет им. Н. П. Огарёва»,
г. Саранск
Бурова Анастасия Дмитриевна, *студентка ФДП и СПО*
ФГБОУ ВО «Мордовский государственный университет им. Н. П. Огарёва»,
г. Саранск
Помелова Дарья Витальевна, *студентка ФДП и СПО*
ФГБОУ ВО «Мордовский государственный университет им. Н. П. Огарёва»,
г. Саранск
Старушенкова Екатерина Евгеньевна, *преподаватель ФДП и СПО ФГБОУ ВО «Мордовский государственный университет им. Н. П. Огарёва», г. Саранск*

МЕТОДЫ МОДИФИКАЦИИ RSA ШИФРОВАНИЯ ДЛЯ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ SCADA

Аннотация: В данной статье рассматриваются методы модификации асимметричных шифров для обеспечения безопасности систем SCADA. Приводится конкретный пример улучшения шифра RSA и его работы.

Ключевые слова: SCADA, шифрование, RSA, ассиметричные шифры, ключ, безопасность, АИС.

Annotation: This article discusses methods for modifying asymmetric ciphers to secure SCADA systems. A concrete example of RSA cipher improvement and its work is given.

Key words: SCADA, encryption, RSA, asymmetric ciphers, key, security, AIS.

Системы управления и сбора данных SCADA используются для

управления критически важными системами в различных отраслях, таких как энергетика, промышленность и транспорт. Однако, из-за их важности, SCADA системы становятся объектом атак со стороны злоумышленников, что может привести к серьезным последствиям, включая прекращение работы критически важных систем. Согласно исследованию ТАСС, доля утечек, связанных с коммерческой тайной, за 2022 год удвоилась в сравнении с 2021 годом и составила 9,1% [1].

Одним из методов защиты SCADA систем является шифрование данных. Шифрование является важной мерой безопасности, которая может защитить конфиденциальные данные от несанкционированного доступа и фальсификации. Существуют различные типы шифрования, включая симметричное и асимметричное шифрование. Симметричное шифрование использует один и тот же ключ для шифрования и расшифровки данных, в то время как асимметричное шифрование использует разные ключи для каждой операции.

Одним из примеров шифрования, который может быть использован в контексте технологий АИС и SCADA, является Advanced Encryption Standard (AES). Это широко используемый алгоритм симметричного шифрования, который считается надежным и эффективным. Он работает путем разделения данных на блоки и применения серии операций подстановки и перестановки к каждому блоку с использованием секретного ключа. Зашифрованные данные могут быть расшифрованы только с помощью того же ключа.

Другим примером шифрования является алгоритм RSA (Rivest-Shamir-Adleman), который является широко используемым асимметричным алгоритмом шифрования. RSA использует открытый ключ для шифрования и закрытый ключ для расшифровки. Открытым ключом можно свободно обмениваться, в то время как закрытый ключ должен храниться в секрете. RSA обычно используется для защиты цифровых сертификатов и цифровых подписей.

Асимметричные шифры, такие как RSA, широко используются в SCADA системах для защиты конфиденциальности передаваемых данных. Однако, из-за своей относительной слабости, атаки на системы, использующие RSA

шифрование, могут привести к нарушению конфиденциальности и целостности передаваемых данных.

Методы модификации RSA шифрования:

– Использование дополнительных ключей:

Один из методов модификации RSA-шифрования для обеспечения безопасности SCADA-систем состоит в использовании дополнительных ключей. Это означает, что помимо основного ключа шифрования и дешифрования, используется дополнительный ключ, который используется для проверки подлинности сообщений. Дополнительный ключ может быть создан с помощью алгоритма цифровой подписи, такого как ECDSA или DSA [4].

– Использование многоуровневых схем шифрования:

Еще один метод модификации RSA-шифрования для обеспечения безопасности SCADA-систем — использование многоуровневых схем шифрования. Это означает, что данные шифруются несколько раз, используя различные алгоритмы шифрования. Например, можно использовать комбинацию RSA и AES (Advanced Encryption Standard) для создания более безопасной схемы шифрования.

– Использование квантовых алгоритмов:

Еще один метод модификации RSA-шифрования для обеспечения безопасности SCADA-систем - использование квантовых алгоритмов шифрования. Квантовые алгоритмы шифрования используют квантовые свойства для шифрования данных. Например, квантовые алгоритмы шифрования, такие как QKD (Quantum Key Distribution), могут создавать ключи, которые могут быть считаны только отправителем и получателем.

– Использование хеш-функций:

Другой метод модификации RSA-шифрования для обеспечения безопасности SCADA-систем - использование хеш-функций. Хеш-функции используются для создания фиксированной длины кода из произвольного ввода данных. Это позволяет создавать улучшенную безопасность, так как хеш-

функции могут использоваться для проверки целостности данных, а также для создания цифровых подписей. В этом случае хеш-функция может использоваться для создания дайджеста (контрольной суммы) для сообщения, а затем дайджест будет подписан RSA-ключом отправителя. Получатель сможет использовать открытый ключ отправителя для проверки подписи и убедиться в целостности сообщения [2].

Примеры использования модифицированных методов RSA-шифрования в автоматизированных информационных системах SCADA:

– Использование квантовых алгоритмов для создания безопасного канала связи между серверами системы SCADA и терминалами управления.

– Использование многоуровневых схем шифрования, таких как комбинация RSA и AES, для шифрования данных, передаваемых между серверами и терминалами управления.

– Использование дополнительных ключей, созданных с помощью алгоритма цифровой подписи, для проверки подлинности сообщений, передаваемых между серверами и терминалами управления.

Также для увеличения безопасности RSA можно модифицировать, используя увеличенную длину ключа и добавление дополнительных слоев шифрования. Например, RSA с ключом длиной 4096 бит обычно считается более безопасным, чем RSA с ключом длиной 2048 бит. Кроме того, можно использовать многократное шифрование (например, двойное или тройное шифрование), чтобы усложнить процесс дешифрования.

Рассмотрим пример работы модифицированного RSA шифра с увеличенной длиной ключа и двойным шифрованием. Пусть Пользователь 1 (далее П1) хочет отправить сообщение Пользователю 2 (далее П2). П1 использует открытый ключ П2, чтобы зашифровать сообщение дважды: сначала он шифрует сообщение своим собственным открытым ключом, а затем шифрует результат с открытым ключом П2. После этого П1 отправляет зашифрованное сообщение П2. П2 сначала расшифровывает сообщение своим секретным ключом, затем расшифровывает полученный результат открытым ключом П1.

Таким образом, дополнительный уровень шифрования и увеличенная длина ключа делают процесс взлома шифра более сложным для злоумышленников [5].

Однако, использование более длинных ключей приводит к увеличению времени, необходимого для шифрования и расшифровки данных. Поэтому мы предлагаем использовать метод, основанный на добавлении случайных значений к шифруемому сообщению перед шифрованием. Этот метод называется Optimal Asymmetric Encryption Padding (OAEP) и позволяет устранить некоторые из возможных атак на RSA-шифрование.

Шаги модифицированного RSA-шифрования с использованием OAEP:

– Сервер выбирает RSA-ключи, создавая пару приватного и открытого ключей. Открытый ключ будет использоваться для шифрования сообщений, а приватный ключ для их расшифровки.

– Сервер генерирует случайную строку и добавляет ее к шифруемому сообщению.

– Сервер использует хеш-функцию для вычисления хеш-значения добавленной строки.

– Сервер использует публичный ключ для шифрования строки с добавленным хеш-значением.

– Сервер шифрует шифруемое сообщение и полученную ранее зашифрованную строку вместе с помощью открытого ключа.

– Сервер отправляет полученный шифр на терминал управления.

– Терминал управления использует свой приватный ключ для расшифровки зашифрованного сообщения.

– Терминал управления извлекает зашифрованную строку и расшифровывает ее с помощью публичного ключа.

– Терминал управления использует хеш-функцию для вычисления хеш-значения расшифрованной строки.

– Терминал управления сравнивает вычисленное хеш-значение с полученным от сервера. Если они совпадают, терминал управления извлекает и применяет шифруемое сообщение [4].

Таким образом, модифицированный RSA-шифр с использованием OAEP позволяет улучшить безопасность АИС SCADA за счет добавления случайных значений к шифруемому сообщению перед шифрованием. Это позволяет устранить некоторые возможные атаки на RSA-шифрование, например, атаку посредника или атаку на выбранный шифротекст. Предлагаем ознакомиться с конкретным примером:

Допустим, мы хотим отправить сообщение "Hello, world!" от сервера к терминалу управления в системе SCADA. Мы используем модифицированный RSA-шифр с использованием OAEP.

– Сервер выбирает RSA-ключи и создает пару приватного и открытого ключей. Открытый ключ будет использоваться для шифрования сообщений, а приватный ключ для их расшифровки.

– Сервер генерирует случайную строку "abc123" и добавляет ее к шифруемому сообщению "Hello, world!".

– Сервер использует хеш-функцию SHA-256 для вычисления хеш-значения добавленной строки: "hash(abc123) = 9c6c53e3b3c6f3cb98dafe88f51aee7c2b735db6f9d9e0198c3147e562492f64".

– Сервер использует открытый ключ для шифрования строки с добавленным хеш-значением: "encrypt(abc123 || hash(abc123)) = c89c8830d701b0bb8d19a76cde39e86b4c18243a8b317a832c22adbb1b9e1c74".

– Сервер шифрует шифруемое сообщение и полученную ранее зашифрованную строку вместе с помощью открытого ключа: "encrypt('Hello, world!' || c89c8830d701b0bb8d19a76cde39e86b4c18243a8b317a832c22adbb1b9e1c74)".

– Сервер отправляет полученный шифр на терминал управления.

– Терминал управления использует свой приватный ключ для расшифровки зашифрованного сообщения.

– Терминал управления извлекает зашифрованную строку и расшифровывает ее с помощью публичного ключа:

"decrypt(c89c8830d701b0bb8d19a76cde39e86b4c18243a8b317a832c22adbb1b9e1c74) = abc123 || hash(abc123)".

– Терминал управления использует хеш-функцию SHA-256 для вычисления хеш-значения расшифрованной строки: "hash(abc123) = 9c6c53e3b3c6f3cb98dafe88f51aee7c2b735db6f9d9e0198c3147e562492f64".

– Терминал управления сравнивает вычисленное хеш-значение с полученным от сервера. Если они совпадают, терминал управления извлекает и применяет шифруемое сообщение "Hello, world!".

Этот пример демонстрирует, как модифицированный RSA-шифр с использованием OAEP может использоваться для безопасной передачи сообщений в автоматизированных системах управления и сбора данных.

Однако, стоит отметить, что при использовании RSA-шифрования в SCADA системах, необходимо учитывать ограничения ресурсов на стороне терминалов управления, поэтому могут потребоваться дополнительные оптимизации, такие как сжатие сообщений или использование более эффективных алгоритмов шифрования [1].

Преимущества метода добавления случайных значений к шифруемому сообщению:

– Увеличение стойкости к атакам:

При использовании этого метода, для каждого сообщения создается уникальное значение, которое затрудняет проведение атак типа словарного перебора и других методов анализа структуры сообщений. Это снижает вероятность успешной атаки на зашифрованные данные.

– Защита от известных шаблонов данных:

Если злоумышленник имеет доступ к источнику шифруемых данных и имеет знание о шаблонах этих данных, то добавление случайных значений перед шифрованием может усложнить анализ структуры этих данных и сделать невозможным использование известных шаблонов для проведения атак.

– Дополнительная безопасность передачи данных:

При передаче данных по незащищенному каналу, добавление случайных значений перед шифрованием усложняет проведение атак типа "человек посередине", когда злоумышленник перехватывает и изменяет данные внутри канала передачи.

– Необходимость взломать два шифра:

Криптоаналитику необходимо взломать как шифр добавленных случайных значений, так и основной шифр, что усложняет проведение атак и повышает уровень безопасности.

Однако, важно отметить, что добавление случайных значений к сообщению не является универсальным решением для обеспечения безопасности передаваемых данных. В зависимости от конкретного случая и выбранного алгоритма шифрования, могут существовать другие методы и механизмы, которые лучше подходят для достижения требуемого уровня безопасности.

Библиографический список:

1. ТАСС: исследование: число утекших личных данных в России в 4,5 раза превысило ее население: сайт. – URL: https://tass.ru/ekonomika/17539541 (дата обращения: 15.04.2023). – Режим доступа: свободный. – Текст : электронный.

2. Шиповалов А. В. Модификация алгоритма RSA для повышения уровня безопасности системы защиты информации. Информационно-управляющие системы, / А. В. Шиповалов, С. А. Шиповалова // № 6, 2019 г.

3. Модификация алгоритма RSA для повышения эффективности шифрования больших объемов данных Ю.А. Гурин, Д.Ю. Панфилова // Современные технологии, экономика и образование, № 4, 2019 г.

4. Брынзак А.С. Модифицированный алгоритм RSA для обеспечения безопасности данных в сетях передачи информации. – Научно-технический вестник информационных технологий, механики и оптики, том 17, № 3, 2017 г.

5. "Алгоритм RSA с измененным паддингом OAEP и его применение для защиты данных" А.А. Макаренко, А.А. Петров // Труды международной конференции по интеллектуальным системам IS'12, 2012 г.