УДК 004.41

ЛОКАЛИЗАЦИЯ НЕКОТОРЫХ ЧАСТЕЙ КОММУНИКАЦИОННОГО ПРОТОКОЛА WIREGUARD

А.Г. Бокун
Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», Минск, Беларусь

Аннотация. В статье рассмотрена одна из составляющих механизма защиты информации в коммуникационном протоколе WireGuard. Во введении обозначена актуальность применения и локализации современных протоколов безопасной передачи данных с открытой спецификацией. В основной части была дана краткая характеристика частям механизма обеспечения аутентифицированного шифрования в Wireguard. Кроме этого было рассмотрено альтернативное решение на основе государственного стандарта Республики Беларусь СТБ 34.101.31. В заключении на основе проведенной оценки криптографических примитивов WireGuard и шифров из СТБ были сделаны выводы о целесообразности локализации именно этого протокола.
Ключевые слова: аутентифицированное шифрование; блочный шифр; функция генерации имитовставки; протокол сетевого уровня; криптографические примитивы; шифротекст.

LOCALIZATION OF SOME PARTS OF THE WIREGUARD COMMUNICATION PROTOCOL

A.G. Bokun
Educational Institution "Belarusian State University of Informatics and Radioelectronics", Minsk, Belarus

Abstract. The article considers one of the components of the information protection mechanism in the WireGuard communication protocol. In the introduction the relevance of application and localization of modern protocols of secure data transmission with an open specification is outlined. In the main part a brief characterization of parts of the mechanism of providing authenticated encryption in Wireguard was given. In addition, an alternative solution based on the state standard of the Republic of Belarus STB 34.101.31 was considered. In conclusion, based on the evaluation of cryptographic primitives of WireGuard and ciphers from STB, conclusions were made about the expediency of localization of this protocol.
Keywords: authenticated encryption; block cipher; message authentication code; network layer protocol; cryptographic primitives; ciphertext.

Введение

Современные тренды в области защиты средств связи делают ставку на легкие и производительные протоколы. Манифестацией этого тренда стал выпущенный в 2015 г. протокол сетевого уровня TCP/IP WireGuard. Сильными сторонами этого протокола являются скорость, применение современных криптографических моделей и компактная кодовая база. Исходя из этого считаем рациональным рассмотреть варианты адаптации данного протокола под стандарты Республики Беларусь и его дальнейшую эксплуатацию.

Основная часть

Одним из ключевых аспектов криптографической защиты информации в протоколе WireGuard является алгоритмическая система аутентифицированного шифрования с дополнительными данными (AEAD). В спецификации протокола указано, что в качестве AEAD используется стек ChaCha20-Poly1305. Он состоит из двух алгоритмов – поточного шифра ChaCha20 и кода аутентификации сообщений Poly1305. Актуальным стандартам, регламентирующим работу данного стека, является RFC 8439.

Существует несколько вариантов реализации алгоритма ChaCha: на 8 раундов, на 12 раундов и на 20 раундов. В документе RFC 8439 описывается реализация на 20 раундов, соответственно, WireGuard также использует именно этот вариант.

Суть механизма шифрования заключается в циклическом вызове блок-функции ChaCha20 с одинаковым ключом и вектором инициализации, при этом последовательно увеличиваются параметры счетчика блоков. Затем ChaCha20 сериализует полученное состояние, записывая числа в порядке little-endian, создавая блок потока ключей. Поток ключей образуется через конкатенацию релевантных значений из последовательных блоков. Далее алгоритм выполняет операцию исключающего ИЛИ (XOR) над потоком ключей и исходным текстом. В качестве более оптимизированной альтернативы данному шагу: операцию XOR можно выполнять с каждым блоком потока ключей по отдельности, выбирая для этого соответствующий блок исходного текста.

Стоит помнить, что для исходного текста нет никаких требований по части размера (то есть ему необязательно быть целым кратным 512 битам). Но некоторые конкретные протоколы могут требовать, чтобы открытый и шифрованный текст имели определенную длину.

В спецификации также указано, что если после последнего блока остался лишний ключевой поток, то он отбрасывается. Входными данными для шифра ChaCha20 являются:

– 256-битный ключ;
– 32-битный начальный счетчик, обычно это ноль или единица;
– 96-битный вектор инициализации;
– исходный текст произвольной длины.

В результате работы получается зашифрованное сообщение, той же длины, что и исходный.

Расшифровка выполняется аналогичным образом. Блок-функция ChaCha20 используется для расширения ключа в поток ключей. Далее проводится операция исключающего ИЛИ с зашифрованным текстом, что позволяет получить исходный текст.

Poly1305 является одноразовым аутентификатором. Этот примитив принимает на вход исходные данные и 32-байтный одноразовый ключ, отдает же 16-байтную подпись. Эта подпись может быть использована для проверки подлинности и целостности сообщения.

Poly1305 – это полиномиально-оценочная функция генерации имитовставки. Такие функции представляют каждое сообщение как одномерный многочлен над конечным полем, а затем оценивает этот многочлен на ключе. Полиномиально-оценочные имитовставки сочетают в себе несколько привлекательных в контексте производительности особенностей: не ресурсоемкую генерация коротких ключей и быструю аутентификацию сообщений.

Функция выработки имитовставки Poly1305 представлена в следующей формуле: $Poly1305_r(m, AES_k(n))$, где m – сообщение, k – ключ AES, r – дополнительный ключ, n – синхропосылка. Из формулы создания имитовставки можно понять, что AES здесь используется только для шифрования синхропосылки и получения 128-битной уникальной строки. При этом в стандарте Poly1305 никак не закреплено использование именно AES. Это одновременно делает алгоритм крайне безопасным (его безопасность напрямую зависит от безопасности AES или другого используемого шифра) и гибким, так как AES можно будет заменить любым другим шифром того же класса. Например, в схеме AEAD AES будет заменен шифрованием ChaCha20.

Независимо от способа генерации ключ делится на две части, называемые *r* и *s*. Пара (*r*, *s*) должна быть уникальной и непредсказуемой для каждого обращения (именно поэтому изначально она была получена путем шифрования синхропосылки), в то время как r быть константным. Таким образом, входные данные для функции генерации имитовставки *Poly*1305 будут следующими:

– 256-битный одноразовый ключ;

– некоторое сообщение произвольной длины.

На выходе же будет 128-битный код аутентификации.

AEAD (Authenticated Encryption with Additional Data) является конструкцией, которая представляет собой единый стек из ChaCha20 и Poly1305. AEAD необходим для реализации схемы аутентифицированного шифрования (AE). Существует несколько подходов к реализации AE:

– Шифрование до MAC (EtM) – этот подход предполагает шифрование исходного текста и дальнейшее получение имитовставки через функцию генерации.

– Шифрование и MAC (E&M) – здесь шифрование и получение имитовставки происходят независимо на одном и том же входном тексте.

– MAC до шифрования (MtE) – в данном сценарии первоначально происходит получение имитовставки, после чего имитовставка и входной текст шифруются вместе.

В стеке ChaCha20-Poly1305 реализован первый вариант – шифрование до получения имитовставки. Сначала ChaCha20 шифрует исходный текст, а потом, уже на шифре, получается имитовставка. Общая схема работы ChaCha20-Poly1305 представлена на рис. 1.

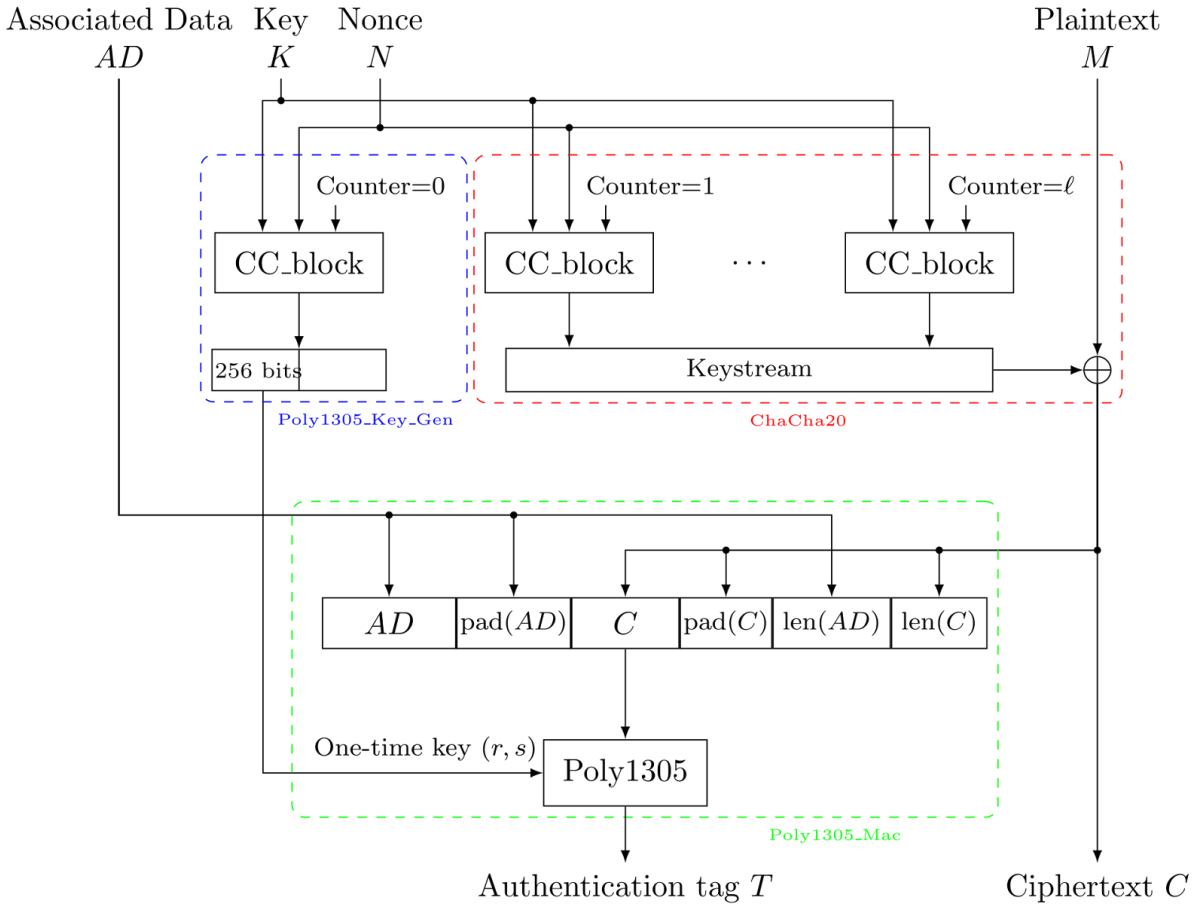

Рис. 1. Общая схема работы AEAD ChaCha20-Poly1305
Fig. 1. General operating scheme of AEAD ChaCha20-Poly1305

Расшифровка сообщения выполняется обратным ходом ChaCha20, при этом Poly1305 все еще применяется на зашифрованном сообщении так как необходимо проверить целостность сообщения.

Имея представление о принципах работы этой части системы шифрования Wireguard можно подобрать алгоритм, который сможет заменить AEAD, определенный в RFC 8439, на подходящий стек криптографических примитивов, соответствующих СТБ.

Наиболее подходящим для задачи интеграции AEAD шифром, который стандартизирован в Республике Беларусь, является алгоритм BelT. BelT – блочный шифр, который описан в СТБ 34.101.31. AEAD был представлен во второй (belt-dwp) и третьей (belt-che) версиях стандарта, от 2011-го и 2020-го годов соответственно. Оба алгоритма (belt-dwp и belt-che) являются актуальными схемами аутентифицированного шифрования и присутствуют в последних редакциях стандарта.

Для использования в локализованной версии Wireguard предлагается вариант belt-dwp, так как режим DWP у AEAD является аналогом режима GCM, но более защищенным, режим же GCM в свою очередь активно используется в традиционных VPN-протоколах вроде OpenVPN. В спецификации belt-dwp указаны следующие входные данные:

– сообщение произвольной длины;
– дополнительные (ассоциированные данные);
– ключ (256 байт);
– синхропосылка (128 байт).

Размерность входных данные соответствуют (за исключением размера синхропосылки) спецификации AEAD ChaCha20-Poly1305, что значительно упростит процесс интеграции.

Заключение

Wireguard является крайне производительным и удобным для расширения протоколом, что делает его локализацию крайне полезной и важной задачей. В статье было продемонстрировано, что государственные стандарты Республики Беларусь могут предоставить криптографические примитивы, которые способны стать аналогами нестандартизированных решений и обеспечить верифицированную безопасность коммуникации. Кроме этого соответствие алгоритмов из СТБ внешним интерфейсам родных шифров Wireguar делает интеграцию довольно удобной, а работы в этой области перспективными.

Список использованных источников / References

1. Bernstein, D.J. (2008) ChaCha, a Variant of Salsa20. Journal of Software Engineering and Applications. 16 (12), 24-30.

2. Bernstein, D.J. (2005) The Poly1305-AES Message-Authentication Code. Lecture Notes in Computer Science. 3557, 32-49.

3. Y. Nir, A. Langley (2018) RFC 8439: ChaCha20 and Poly1305 for IETF Protocols. USA, RFC Editor.

Сведения об авторе

Бокун А.Г., ассистсент кафедры информатики, учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», a.bokun@bsuir.by.

Information about the author

Bokun A.G, Assistant at the Department of Informatics, Educational Institution "Belarusian State University of Informatics and Radioelectronics", a.bokun@bsuir.by.