Российская Академия Наук
Институт системного программирования

IP Authentication Header

Протокол аутентифицирующего заголовка на уровне IP

RFC 4302 “ IP Authentication Header ”
Перевод на русский язык.

Москва 2007 г.

Список исполнителей

Исполнитель	Контактная информация	Примечания
Шнитман Виктор Зиновьевич, д.т.н., профессор	vzs@ispras.ru	Перевод.

История документа

№ версии	Дата	Примечания
1.0	01.12.2007	Перевод опубликован.

Аннотация.

В данном документе представлен перевод на русский язык RFC 4302, регламентирующего формат и правила обработки аутентифицирующей информации при защищенном обмене данными в рамках архитектуры IPsec.

Перевод выполнен в рамках проекта по гранту Российского фонда фундаментальных исследований № 07-07-00243 «Верификация функций безопасности протокола нового поколения IPsec v2».

Network Working Group
Request for Comments: 4302
Obsoletes: 2402
Category: Standards Track

S. Kent
BBN Technologies

December 2005

IP Authentication Header

Протокол аутентифицирующего заголовка на уровне IP

Статус данного меморандума

Данный документ определяет для сообщества Internet протокол, находящийся в процессе стандартизации, и требует обсуждения, а также предложений по его улучшению. За информацией, касающейся состояния стандартизации и статуса данного протокола, обращайтесь, пожалуйста, к текущей версии документа "Официальные стандарты протоколов Internet" (STD 1). Распространение данного меморандума не ограничено.

Замечание относительно авторских прав

Аннотация

В данном документе описывается обновленная версия протокола аутентифицирующего заголовка IP (AH - Authentication Header), которая разработана с целью предоставления сервисов аутентификации в IPv4 и IPv6. Данный документ отменяет действие документа RFC 2402 (Ноябрь 1998).

Содержание

1. Введение

В данном документе предполагается, что читатель знаком с терминами и концепциями, описанными в документе "Security Architecture for the Internet Protocol" [Ken-Arch], который далее называется документом по архитектуре безопасности. В частности читатель должен быть знаком с определениями сервисов безопасности, которые предлагаются протоколом инкапсулирующей защиты данных (ESP - Encapsulating Security Payload) [Ken-ESP] и протоколом аутентифицирующего заголовка (AH - Authentication Header), концепцией контекста безопасности (Security Association), способами использования ESP совместно с AH и различными дополнительными возможностями управления ключами, доступными для ESP и AH.

Когда в данном документе появляются ключевые слова MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY и OPTIONAL, они должны интерпретироваться так, как описано в документе RFC 2119 [Bra97].

Аутентифицирующий заголовок (AH) используется для обеспечения сервисов целостности в режиме без установления соединения и аутентификации источника данных IP-дейтаграмм (далее называется просто "целостностью"), а также для защиты от повторного воспроизведения. Этот последний необязательный сервис может выбираться получателем в процессе установления контекста безопасности. (По умолчанию протокол требует от отправителя инкрементирования порядкового номера, используемого для противодействия повторному воспроизведению, но этот сервис действует, только если получатель контролирует порядковые номера). Однако чтобы использовать функцию расширенных порядковых номеров (Extended Sequence Number) в интероперабельной форме, AH предъявляет к протоколам управления контекстами безопасности (SA) требование возможности согласования этой новой функции (см. ниже п. 2.5.1).

Протокол AH обеспечивает аутентификацию максимально возможного количества полей IP-заголовка, а также данных протокола следующего уровня. Однако некоторые поля IP-заголовка могут по пути меняться, и значения этих полей в момент поступления пакета к получателю не могут быть предсказаны отправителем. Значения таких полей не могут защищаться с помощью AH. Таким образом, защита, предоставляемая IP-заголовку протоколом AH, является частичной (см. приложение A).

Протокол AH может применяться в одиночку, в комбинации с протоколом ESP [Ken-ESP], или во вложенном виде (см. документ по архитектуре безопасности [Ken-Arch]). Сервисы безопасности могут предоставляться между парой хостов, осуществляющих обмен информацией, между парой общающихся между собой защитных шлюзов, или между защитным шлюзом и хостом. Для предоставления тех же самых сервисов предотвращения повторного воспроизведения и аналогичных сервисов целостности может использоваться протокол ESP, но, кроме того, он предоставляет также сервис конфиденциальности (шифрования). Основное отличие между целостностью, обеспечиваемой протоколом ESP и целостностью, обеспечиваемой протоколом AH, заключается в степени покрытия. Более точно, ESP не защищает никакие поля IP-заголовка, если эти поля не инкапсулированы в ESP (например, посредством использования туннельного режима). Более подробную информацию о том, как использовать AH и ESP в различной сетевой среде, см. в документе по архитектуре безопасности [Ken-Arch].

В разд. 7 представлен краткий обзор отличий данного документа от документа RFC 2402 [RFC2402].

2. Формат аутентифицирующего заголовка

Заголовок протокола (IPv4, IPv6) или заголовок расширения IPv6, непосредственно предшествующий заголовку AH, должен (SHALL) содержать значение 51 в полях Protocol (IPv4) или Next Header (IPv6, заголовка расширения) [DH98].

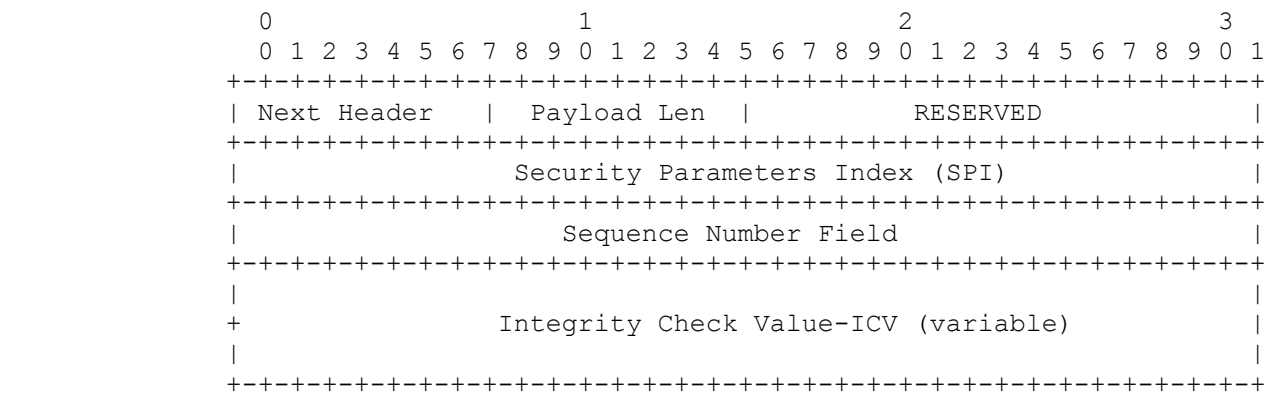


Рис. 1. Формат AH

Следующая таблица относится к полям, составляющим AH (показанных на рис. 1), а также к другим полям, включенным в вычисление целостности, и иллюстрирует, какие поля покрываются ICV и что передается.

	Количество байтов	Обязательность [1]	Что покрывает целостность	Что передается
IP Header	переменное	M	[2]	открытый текст
Next Header	1	M	Да	открытый текст
Payload Len	1	M	Да	открытый текст
RESERVED	2	M	Да	открытый текст
SPI	4	M	Да	открытый текст
Seq# (младшие 32 бита)	4	M	Да	открытый текст
ICV	переменное	M	Да[3]	открытый текст
IP datagram [4]	переменное	M	Да	открытый текст
Seq# (старшие 32 бита)	4	если ESN	Да	не передается
ICV Padding	переменное	если необходимо	Да	не передается

[1] - M = mandatory (обязательное)
[2] - Подробнее о том, какие поля IP-заголовка покрываются, см. в п. 3.3.3 "Вычисление значения для контроля целостности"
[3] – Обнуляется перед вычислением ICV (здесь помещается ICV, полученный в результате вычисления)
[4] - Если туннельный режим -> IP-дейтаграмма
Если транспортный режим -> следующий заголовок и данные

В следующих подразделах определяются поля, составляющие формат AH. Все описанные здесь поля являются обязательными; т.е. они всегда присутствуют в формате AH и включаются в вычисление значения для контроля целостности (ICV - Integrity Check Value) (см. подразд. 2.6 и п. 3.3.3).

Примечание: Все криптографические алгоритмы, используемые в IPsec, предполагают, что их входные данные поступают в каноническом сетевом порядке байтов (см. приложение к RFC 791 [RFC791]) и формируют свои результирующие данные в каноническом сетевом порядке байтов. Пакеты IP также передаются в сетевом порядке байтов.

AH не содержит номера версии, поэтому, если возникает беспокойство относительно обратной совместимости, оно должно (MUST) закрываться посредством механизма сигнализации между двумя партнерами IPsec, чтобы гарантировать совместимость версий AH, например, с помощью протокола IKE [IKEv2] или посредством отдельного механизма конфигурирования.

2.1. Поле Next Header

Поле Next Header (следующий заголовок) представляет собой 8-битовое поле, которое определяет тип блока данных, следующего за аутентифицирующим

заголовком. Значение этого поля выбирается из множества номеров протоколов IP, которое определено на web-странице регистрационного органа Internet (IANA - Internet Assigned Numbers Authority). Например, значение 4 указывает на IPv4, значение 41 – на IPv6, а значение 6 – на TCP.

2.2. Поле Payload Length (длина блока данных)

Это 8-битовое поле определяет длину AH в 32-битовых словах (четырехбайтовых элементов данных), минус "2". Таким образом, например, если алгоритм целостности формирует 96-битовое аутентифицирующее значение, то это поле длины будет содержать "4" (три 32-битовых слова фиксированных полей плюс три 32-битовых слова для ICV, минус 2). Для IPv6 общая длина заголовка должна быть кратна 8-октетным элементам данных. (Заметим, что хотя IPv6 [DH98] описывает AH как заголовок расширения, его длина измеряется в 32-битовых словах, а не в 64-битовых словах, которые используются другими заголовками расширения IPv6). Комментарии относительно заполнения этих полей см. в подразд. 2.6 "Значение для контроля целостности (ICV)" и п.п. 3.3.3.2.1 "Заполнение ICV".

2.3. Поле Reserved

Это 16-битовое поле зарезервировано для будущего использования. Оно должно (MUST) устанавливаться в "ноль" отправителем и должно (SHOULD) игнорироваться получателем. (Заметим, что это значение включается в вычисление ICV, но во всем остальном игнорируется получателем.

2.4. Поле Security Parameters Index (SPI)

Поле SPI (индекс параметров безопасности) представляет собой произвольное 32-битовое значение, которое используется получателем для идентификации SA, к которому привязывается входящий пакет. Для индивидуального (unicast) контекста безопасности SPI может использоваться для определения SA сам по себе, или он может использоваться вместе с типом протокола IPsec (в данном случае с AH). Поскольку для индивидуальных SA значение SPI создается получателем, вопрос о том, является ли это одно значение достаточным для идентификации SA, или оно должно использоваться вместе со значением протокола IPsec, является локальным вопросом. Поле SPI является обязательным, и этот описанный выше механизм отображения входного трафика на индивидуальные SA должен (MUST) поддерживаться всеми реализациями AH.

Если реализация IPsec поддерживает многоадресную рассылку (multicast), то для отображения на SA входящих IPsec-дейтаграмм она должна (MUST) поддерживать групповые SA с помощью описанного ниже алгоритма. Реализации, поддерживающие только индивидуальный трафик, не нуждаются в реализации этого алгоритма демультиплексирования.

Во многих архитектурах защищенной групповой рассылки, например, в [RFC3740], SPI группового контекста безопасности в одностороннем порядке устанавливает контроллер группы/сервер ключей. Это присваивание SPI не согласуется и не координируется с подсистемами управления ключами (например, с IKE), которые размещаются в отдельных оконечных системах, составляющих группу. Следовательно возможно, что групповой контекст безопасности и индивидуальный контекст безопасности будут одновременно использовать один и тот же SPI. Реализация IPsec, допускающая многоадресную рассылку, должна (MUST) правильно демультиплексировать входящий трафик даже в случае коллизий SPI.

Каждый элемент базы данных контекстов безопасности (SAD - Security Association Database) [Ken-Arch] должен указывать, используются ли при поиске SA совместно со SPI IP-адрес места назначения, или IP-адреса места назначения и источника. Для групповых SA поиск SA осуществляется без использования поля протокола. В случае групповых контекстов безопасности для поиска SA поле протокола не используется. Для каждого входящего пакета, защищенного IPsec, реализация должна осуществлять поиск в SAD так, чтобы он находил элемент, который соответствует "самому длинному" идентификатору SA. В этом контексте,

если два или большее число элементов SAD совпадают по значению SPI, то элемент, который при сравнении адресов дает также совпадение по адресу места назначения, или по адресу места назначения и адресу источника, и представляет собой "самое длинное" соответствие. Это предполагает следующее логическое упорядочивание поиска в SAD:

1. Выполнить поиск в SAD на совпадение по полям {SPI, destination address, source address}. Если какой-то элемент SAD дает совпадение, то обработать входящий пакет AH в соответствии с этим совпадающим элементом SAD. В противном случае перейти к шагу 2.

2. Выполнить поиск в SAD на совпадение по полям {SPI, destination address}. Если какой-то элемент SAD дает совпадение, то обработать входящий пакет AH в соответствии с этим совпадающим элементом SAD. В противном случае перейти к шагу 3.

3. Выполнить поиск в SAD на совпадение только по полю {SPI}, если получатель для AH и ESP выбрал поддержку только одного пространства SPI, или в противном случае поиск на совпадение по полям {SPI, protocol}. Если какой-то элемент SAD дает совпадение, то обработать входящий пакет AH в соответствии с этим совпадающим элементом SAD. В противном случае отбросить пакет и записать в журнал подлежащее аудиту событие.

На деле для ускорения этого поиска реализация может (MAY) выбрать любой метод, хотя ее видимое извне поведение должно (MUST) быть функционально эквивалентным поиску в SAD в указанном выше порядке. Например, программная реализация может индексировать хэш-таблицу с помощью SPI. Элементы SAD в каждом связанном списке областей памяти хэш-таблицы хранятся отсортированными таким образом, что в этом связанном списке первыми оказываются элементы SAD с самыми длинными идентификаторами SA. Те элементы SAD, которые имеют самые короткие идентификаторы SA, отсортированы так, что они находятся в конце этого связанного списка. Аппаратная реализация может в действительности осуществлять поиск самого длинного совпадения с помощью общедоступной адресуемой по содержимому памяти с тремя адресными элементами (TCAM - Ternary Content-Addressable Memory).

Указание того, требуется ли сравнение с адресом источника и адресом места назначения для отображения входящего трафика на контексты безопасности, должно (MUST) устанавливаться либо как побочный эффект ручного конфигурирования SA, либо посредством согласования с помощью протокола управления SA, например, IKE или домена интерпретации групп (GDOI - Group Domain of Interpretation) [RFC3547]. Обычно группы, применяющие многоадресную рассылку для конкретных источников (SSM - Source-Specific Multicast) [HC03], используют трехэлементный кортеж идентификатора SA, состоящий из SPI, группового адреса места назначения и адреса источника. Контекст безопасности многоадресной рассылки для произвольных источников (Any-Source Multicast) в качестве идентификатора SA требует только SPI и групповой адрес места назначения.

Множество значений SPI в диапазоне от 1 до 255 резервируется регистрационным органом IANA (Internet Assigned Numbers Authority) для будущего использования: зарезервированное значение SPI обычно не будет присваиваться IANA, если только использование присвоенного значения SPI не определено в каком-то RFC. Нулевое значение SPI (0) резервируется для локального использования конкретной реализацией и не должно (MUST NOT) посылаться по линии связи. (Например, реализация управления ключами может использовать нулевое значение SPI в смысле "Контекст безопасности не существует") в течение периода времени, когда реализация IPsec запросила, чтобы ее объект управления ключами установил новый SA, но этот SA еще не установлен).

2.5. Поле Sequence Number (порядковый номер)

Это 32-битовое поле без знакового разряда содержит значение счетчика, которое увеличивается на единицу после посылки каждого пакета, т.е. содержит порядковый номер пакета для каждого SA. Для индивидуального (unicast) SA или для группового SA с одним отправителем, отправитель должен (MUST) инкрементировать это поле для каждого передаваемого пакета. Разделение одного контекста безопасности между несколькими отправителями разрешено, хотя обычно не рекомендуется. В контексте нескольких отправителей AH не предоставляет никаких средств для синхронизации счетчиков пакетов между несколькими отправителями или для осмысленного управления счетчиком пакетов и окном получателя. Таким образом, для SA с несколькими отправителями свойства AH по противодействию повторному воспроизведению не доступны (см. п. 3.3.2 и п. 3.4.3).

Это поле является обязательным и должно (MUST) всегда присутствовать, даже если получатель для конкретного SA решил не использовать сервис противодействия повторному воспроизведению. Обработка поля Sequence Number оставлена на усмотрение получателя, но все реализации AH должны (MUST) быть способны выполнять обработку, описанную в п. 3.3.2 "Формирование порядкового номера" и п. 3.4.3 "Проверка порядкового номера". Таким образом, отправитель должен (MUST) передавать это поле, но получатель не обязан им руководствоваться.

Счетчик отправителя и счетчик получателя инициализируются нулем при установлении SA. (Первый пакет, посланный с помощью данного SA, будет иметь порядковый номер 1; дополнительные детали порождения порядковых номеров см. в п. 3.3.2). Если дается разрешение на противодействие повторному воспроизведению (по умолчанию), передаваемый порядковый номер не должен никогда зацикливаться. Таким образом, счетчик отправителя и счетчик получателя должны (MUST) сбрасываться (путем установления нового SA и, соответственно, нового ключа) до передачи по SA пакета с номером 2^32.

2.5.1. Расширенные (64-битовые) порядковые номера

Для поддержки высокоскоростных реализаций IPsec в качестве расширения текущего 32-битового поля порядкового номера должны (SHOULD) реализовываться расширенные порядковые номера (ESN - Extended Sequence Number). Использование ESN должно (MUST) быть согласовано протоколом управления SA. Заметим, что в IKEv2 такое согласование является неявным; если согласование 32-битовых порядковых номеров не производится явно, то по умолчанию используются ESN. (Опция ESN применима как к групповым, так и к индивидуальным SA).

Средство ESN позволяет использовать для SA 64-битовые порядковые номера. Старшие 32 бита являются частью счетчика порядковых номеров, поддерживаются как передатчиком, так и приемником, и включаются в вычисление ICV, но не передаются.

2.6. Поле Integrity Check Value (ICV)

Это поле переменной длины, которое содержит значение для контроля целостности данного пакета. Это поле по длине должно быть целым кратным 32 битам (для IPv4 или IPv6). Детали обработки ICV описываются в п. 3.3.3 "Вычисление значения для контроля целостности" и п. 3.4.4 "Проверка значения для контроля целостности". При необходимости это поле может включать явное заполнение, чтобы гарантировать, что длина заголовка AH, будет целым кратным 32 битам (IPv4) или 64 битам (IPv6). Все реализации должны (MUST) поддерживать такое заполнение и должны (MUST) вставлять заполнение только такой длины, которая достаточна для удовлетворения требований выравнивания IPv4/IPv6. Детали вычисления необходимой длины заполнения даны ниже в пп. 3.3.3.2 "Заполнение". Спецификация алгоритма целостности должна (MUST) определять длину ICV, правила сравнения и шаги обработки для подтверждения правильности.

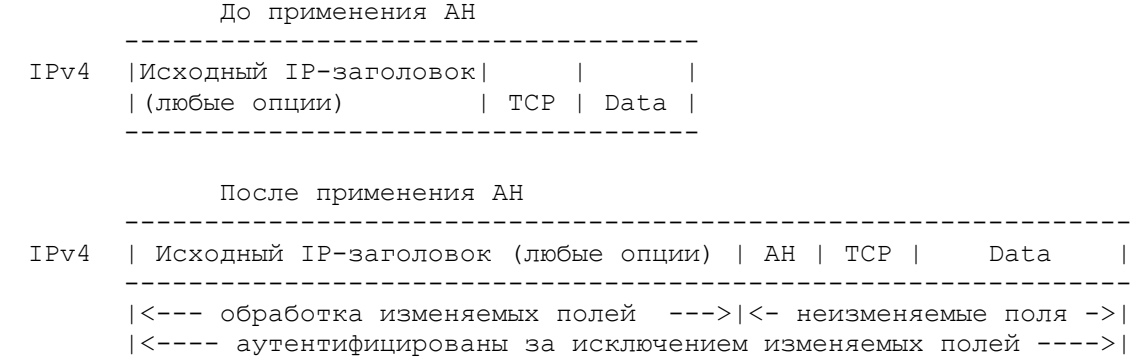
3. Обработка аутентифицирующего заголовка

3.1. Местоположение аутентифицирующего заголовка

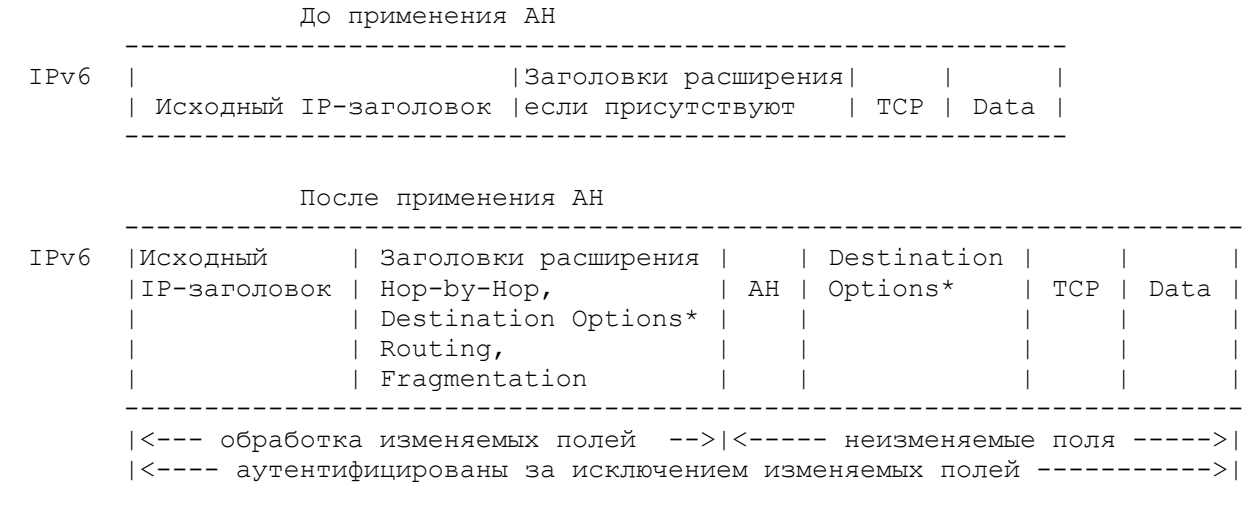
AH может применяться в двух известных режимах: в транспортном режиме или в туннельном режиме. (Описание того, когда должен использоваться каждый из этих режимов см. в документе по архитектуре безопасности).

3.1.1. Транспортный режим

В транспортном режиме AH вставляется после IP-заголовка и перед протоколом следующего уровня (например, TCP, UDP, ICMP и т.д.), или перед любыми другими заголовками IPsec, которые уже были вставлены до этого. В контексте IPv4 это требует помещения AH после IP-заголовка (и любых опций, которые он содержит), но до протокола следующего уровня. (Заметим, что термин "транспортный" режим не должен неправильно истолковываться как ограничение его использования протоколами TCP и UDP). Следующая диаграмма иллюстрирует выбор позиции AH транспортного режима для типового пакета IPv4 по принципу "до и после" его применения.



В контексте IPv6 AH рассматривается как сквозной блок данных и, таким образом, должен появляться после заголовков расширения Hop-by-Hop Extension Header, Routing Extension Header и Fragmentation Extension Header. Заголовок (заголовки) расширения опций места назначения Destination Options Extension Header может появляться до или после, а также как до, так и после заголовка AH в зависимости от требуемой семантики. Следующая диаграмма иллюстрирует выбор местоположения AH транспортного режима для типового пакета IPv6.



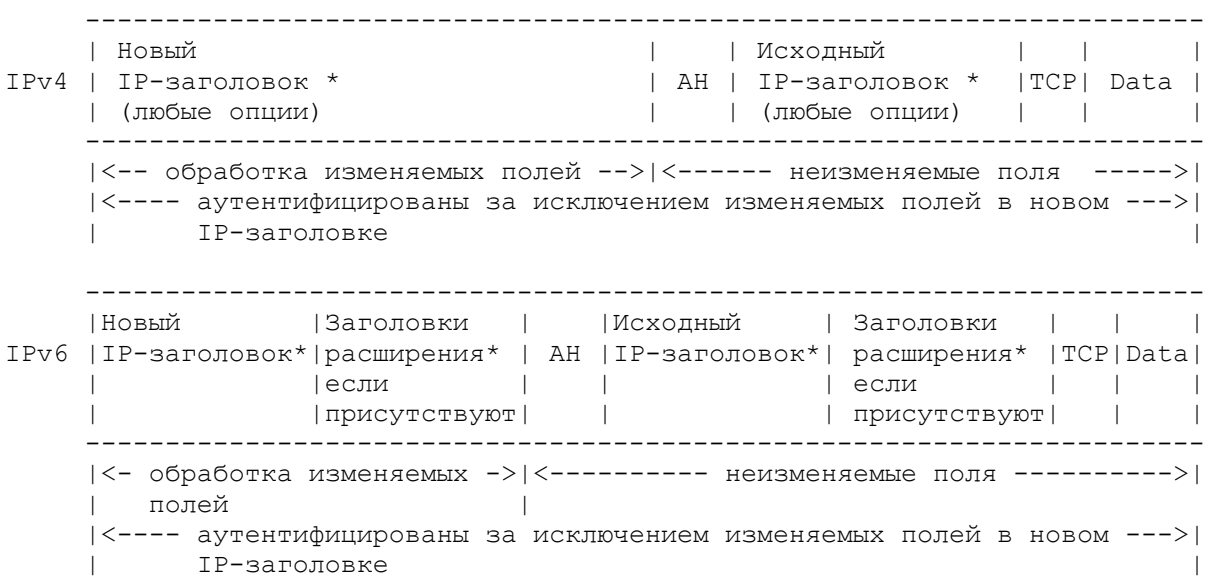
* = Если присутствует, может находиться до AH, после AH, или и до и после AH

Заголовки ESP и AH могут комбинироваться множеством способов. В документе по архитектуре безопасности описаны комбинации контекстов безопасности, которые должны поддерживаться реализациями.

Заметим, что для обеспечения соответствия данной спецификации, а также прозрачной поддержки IPsec в реализациях типа "вставки в стек" или "вставки в линию", которые определены в документе по архитектуре безопасности, в транспортном режиме входящие и исходящие IP-фрагменты могут потребовать от реализации IPsec выполнения дополнительного реассемблирования или дополнительной фрагментации. Если используются несколько интерфейсов, то при выполнении таких операций в этих реализациях требуется особая тщательность.

3.2. Туннельный режим

В туннельном режиме "внутренний" IP-заголовок доставляет конечные IP-адреса источника и места назначения, в то время как "внешний" IP-заголовок содержит адреса "партнеров" IPsec, например, адреса защитных шлюзов. Допускается смешивание внутренней и внешней версий IP, т.е. работа IPv6 поверх IPv4 и IPv4 поверх IPv6. В туннельном режиме AH защищает весь внутренний IP-пакет, включая весь внутренний IP-заголовок. Месторасположение AH в туннельном режиме по отношению к внешнему IP-заголовку то же самое, что и для AH в транспортном режиме. Следующая диаграмма иллюстрирует выбор местоположения AH туннельного режима для типовых пакетов IPv4 и IPv6.



* = если присутствует; построение внешнего IP-заголовка/заголовков расширения и модификация внутреннего IP-заголовка/заголовков расширения обсуждаются в документе по архитектуре безопасности.

3.2. Алгоритмы целостности

Алгоритм целостности, применяемый для вычисления ICV, определяется контекстом безопасности. Для коммуникаций точка-точка подходящими алгоритмами целостности являются управляемые ключом коды аутентификации сообщений (MAC - Message Authentication Code), основанные на симметричных алгоритмах шифрования (например, AES [AES]) или односторонние хэш-функции (например, MD5, SHA-1, SHA-256 и т.д.). Для группового обмена информацией разработано множество криптографических стратегий, и в этой области продолжаются исследования.

3.3. Обработка исходящих пакетов

В транспортном режиме отправитель, как описано выше, вставляет заголовок AH после IP-заголовка и перед заголовком протокола следующего уровня. В туннельном режиме внешний и внутренний IP-заголовки/ заголовки расширения

могут быть взаимосвязаны различными способами. Построение внешнего заголовка/ заголовков расширения в процессе инкапсуляции описано в документе по архитектуре безопасности.

3.3.1. Поиск контекста безопасности

AH применяется к исходящему пакету только после того как реализация IPsec определит, что этот пакет связан с некоторым SA, который требует обработки AH. Процесс определения того, какая обработка IPsec должна применяться к исходящему трафику (если таковая вообще должна иметь место), описывается в документе по архитектуре безопасности.

3.3.2. Формирование порядкового номера

Когда создается SA, счетчик отправителя инициализируется нулем. Отправитель инкрементирует счетчик порядковых номеров (или ESN) для данного SA и вставляет младшие 32 бита полученного значения в поле Sequence Number. Таким образом, первый посланный пакет, использующий данный SA, будет содержать порядковый номер, равный единице.

Если дается разрешение на противодействие повторному воспроизведению (которое устанавливается по умолчанию), отправитель осуществляет проверку, чтобы гарантировать, что перед вставкой нового значения в поле Sequence Number счетчик не зациклился. Другими словами, отправитель не должен (MUST NOT) посылать пакет по SA, если это приведет к зацикливанию порядковых номеров. Попытка передачи пакета, который приведет к переполнению порядкового номера, является событием, которое подлежит аудиту. Элемент журнала аудита для такого события должен (SHOULD) включать значение SPI, текущие дату и время, адрес источника, адрес места назначения и (в IPv6) открытый текст идентификатора потока (Flow ID).

Отправитель предполагает, что разрешение на противодействие повторному воспроизведению дается по умолчанию, если только иное не сообщено получателем (см. п. 3.4.3) или, если SA был сконфигурирован посредством ручного управления ключами. Таким образом, обычное поведение реализации AH требует от отправителя установления нового SA, когда порядковый номер (или ESN) зацикливается, или упреждения зацикливания этого значения.

Если противодействие повторному воспроизведению запрещается (как указано выше), отправителю не надо наблюдать или обнулять этот счетчик, например, в случае ручного управления ключами (см. разд. 5). Однако отправитель все еще инкрементирует этот счетчик и, когда он достигает максимального значения, счетчик обнуляется. (Такое поведение рекомендуется для групповых SA, имеющих нескольких отправителей, если только между отправителем и получателем не согласуются механизмы противодействия повторному воспроизведению, выходящие за рамки настоящего стандарта).

Если выбраны расширенные порядковые номера (ESN) (см. приложение B), то в поле Sequence Number передаются только 32 младших бита порядкового номера, хотя как отправитель, так и получатель поддерживают полные 64-битовые счетчики. Однако старшие 32 бита включаются в вычисление ICV.

Примечание: Если получатель принимает решение не разрешать противодействие повторному воспроизведению для некоторого SA, то этот получатель не должен (SHOULD NOT) согласовывать ESN в протоколе управления SA. Использование ESN создает необходимость для получателя управлять окном противодействия повторному воспроизведению (чтобы определять корректное значение для старших битов ESN, которые используются при вычислении ICV), что, вообще говоря, противоречит понятию запрещения противодействия повторному воспроизведению для SA.

3.3.3. Вычисление значения для контроля целостности

В вычисление значения для контроля целостности (ICV) в AH включаются следующие поля:

- Поля IP-заголовка или заголовка расширения перед заголовком AH, которые либо остаются постоянными в пути, либо значение которых в момент поступления в оконечную точку AH SA можно предсказать
- Заголовок AH (поля Next Header, Payload Len, Reserved, SPI, Sequence Number (младшие 32 бита), поле ICV (которое устанавливается в ноль для этого вычисления) и заключительные байты заполнения (если таковые имеются))
- Предполагается что все, что следует за AH, остается в пути неизменным
- Старшие биты ESN (если они используются) и любое неявное заполнение, требуемое алгоритмом целостности

3.3.3.1. Обработка изменяющихся полей

Если в пути поле может быть модифицировано, то значение этого поля с целью вычисления ICV устанавливается в ноль. Если поле может измениться, но его значение в получателе (IPsec) является предсказуемым, то именно это значение с целью вычисления ICV вставляется в это поле. Само поле Integrity Check Value при подготовке к этому вычислению также устанавливается в ноль. Заметим, что благодаря замене каждого значения поля нулем, а не опусканию этого поля, при вычислении ISV сохраняется выравнивание. Кроме того, подход заполнения нулями гарантирует, что длина полей, которые обрабатываются таким образом, не может меняться по пути, даже если их содержимое покрывается ICV неявно.

В случае разработки нового заголовка расширения или новой опции IPv4, они будут определены в своем собственном документе RFC, который должен (SHOULD) включать (в разделе Анализ безопасности) указания по его (ее) обработке при вычислении AH ICV. Если реализация IP (v4 или v6) обнаруживает заголовок расширения, который она не распознает, она отбросит пакет и пошлет сообщение ICMP. IPsec никогда не увидит этот пакет. Если реализация IPsec обнаруживает опцию IPv4, которую она не распознает, она должна обнулить всю опцию, используя второй байт опции в качестве длины. Опции IPv6 (в заголовках расширения Destination Options Extension Header или Hop-by-Hop Extension Header) содержат флаг, указывающий на возможность изменения, который определяет соответствующую обработку такой опции.

3.3.3.1.1. Вычисление ICV для IPv4

3.3.3.1.1.1. Основные поля заголовка

Основные поля заголовка IPv4 классифицируются следующим образом:

Неизменяемые поля

Version
Internet Header Length
Total Length
Identification
Protocol (Это должно быть значение для AH).
Source Address
Destination Address (без нестрогой или без строгой маршрутизации от источника)

Изменяемые, но предсказуемые поля

Destination Address (с нестрогой или со строгой маршрутизацией от источника)

Изменяемые поля (обнуляются перед вычислением ICV)

Differentiated Services Code Point (DSCP)
(6 бит, см. RFC 2474 [NBBB98])
Explicit Congestion Notification (ECN)
(2 бита, см. RFC 3168 [RFB01])
Flags
Fragment Offset
Time to Live (TTL)
Header Checksum

DSCP – маршрутизаторы могут переделать поле DS так, как это необходимо для обеспечения желаемого локального или сквозного сервиса, так что его значение в момент получения не может быть предсказано отправителем.

ECN – Это поле будет меняться, если маршрутизатор, находящийся на пути, испытывает перегрузку, и, таким образом, его значение в момент получения не может быть предсказано отправителем.

Flags – Это поле исключается, поскольку промежуточный маршрутизатор может установить бит DF, даже если источник его не установил.

Fragment Offset – Поскольку AH применяется только к нефрагментированным IP-пакетам, поле Offset всегда должно быть нулевым, и, таким образом, оно исключается (хотя его значение предсказуемо).

TTL – Это поле меняется по пути маршрутизаторами в результате нормального хода обработки, и, таким образом, его значение в получателе отправителю предсказать невозможно.

Header Checksum – Это поле будет меняться, если изменится любое из указанных других полей, и, таким образом, его значение в получателе отправителю предсказать невозможно.

3.3.3.1.1.2. Опции

В IPv4 (в отличие от IPv6) отсутствует механизм пометки опций, которые изменяются по пути. Поэтому опции IPv4 подробно перечислены в приложении A и классифицированы по категориям: неизменяемые, изменяемые, но предсказуемые, и изменяемые. В IPv4 вся опция рассматривается как единый элемент; таким образом, если опция классифицирована как изменяемая, то вся эта опция для целей вычисления ICV обнуляется, хотя поля типа и длины в большинстве опций по пути не меняются.

3.3.3.1.2. Вычисление ICV для IPv6

3.3.3.1.2.1. Основные поля заголовка

Основные поля заголовка IPv6 классифицируются следующим образом:

Неизменяемые поля

Version
Payload Length
Next Header
Source Address
Destination Address (без заголовка расширения Routing Extension Header)

Изменяемые, но предсказуемые поля

Destination Address (с заголовком расширения Routing Extension Header)

Изменяемые поля (обнуляются до вычисления ICV)

DSCP (6 бит, см. RFC2474 [NBBB98])
ECN (2 бита, см. RFC3168 [RFB01])
Flow Label (*)
Hop Limit

(*) Метка потока, описанная в AHv1, была изменяемой, а в RFC 2460 [DH98] была потенциально изменяемой. Чтобы сохранить совместимость с существующими реализациями AH, в AHv2 метка потока не включена в вычисление ICV.

3.3.3.1.2.2. Заголовки расширения, содержащие опции

Опции IPv6 в заголовках расширения Hop-by-Hop Extension Header и Destination Extension Header содержат бит, который указывает, может ли опция по пути (непредсказуемо) измениться. Для любой опции, содержимое которой может измениться по пути, все поле "Option Data" при вычислении или при проверке ICV должно рассматриваться как заполненное нулевыми октетами. Поля Option Type и Opt Data Len включаются в вычисление ICV. Все опции, для которых этот бит указывает неизменность, включаются в вычисление ICV. Более подробную информацию см. в спецификации IPv6 [DH98].

3.3.3.1.2.3. Заголовки расширения, не содержащие опций

Заголовки расширения IPv6, которые не содержат опций, подробно перечислены в приложении A и классифицированы по категориям: неизменяемые, изменяемые, но предсказуемые, и изменяемые.

3.3.3.2. Заполнение и расширенные порядковые номера

3.3.3.2.1. Заполнение ICV

Как упоминалось в подразд. 2.6, поле ICV может включать явное заполнение, если необходимо гарантировать, чтобы заголовок AH был кратным 32 битам (IPv4) или 64 битам (IPv6). Если требуется заполнение, то его длина определяется двумя факторами:

- длиной ICV
- версией IP-протокола (v4 или v6)

Например, если выходные данные выбранного алгоритма имеют длину 96 бит, то для IPv4 или IPv6 никакого заполнения не требуется. Однако если при использовании другого алгоритма формируется ICV другой длины, то в зависимости от этой длины и версии IP-протокола заполнение может потребоваться. Содержимое поля заполнителя выбирается произвольно отправителем. (Сам заполнитель может быть произвольным, но для обеспечения безопасности не обязательно должен быть случайным). Эти байты заполнения включаются в вычисление ICV, учитываются в длине блока данных (Payload Length) и передаются в конце поля ICV, чтобы получатель мог выполнить вычисление ICV. Включение заполнения, превышающего по размеру минимальное количество, необходимое для удовлетворения требований выравнивания IPv4/IPv6, запрещается.

3.3.3.2.2. Неявное заполнение пакета и ESN

Если для некоторого SA выбрана опция ESN, то старшие 32 бита ESN должны быть включены в вычисление ICV. С целью вычисления ICV эти биты присоединяются неявно непосредственно после конца блока данных и перед любым неявным заполнителем пакета.

Для некоторых алгоритмов целостности строка битов, над которой выполняется вычисление ICV, должна быть кратна размеру блока, определяемому этим алгоритмом. Если длина IP-пакета (включая AH и 32 старших бита ESN, в случае, когда ESN разрешается) не соответствует требованиям алгоритма к размеру блока, то до вычисления ICV к концу пакета должно (MUST) быть добавлено неявное заполнение. Октеты заполнения должны (MUST) иметь нулевое значение. Размер блока (а, следовательно, длина заполнения) определяется спецификацией алгоритма. Это заполнение с пакетом не передается. Для определения необходимости описанного выше неявного заполнения, требуется (MUST) учитывать документ, определяющий алгоритм целостности. Если документ не дает ответ на этот вопрос, то по умолчанию предполагается, что неявное заполнение требуется (поскольку необходимо соответствие между длиной пакета и размером блока в алгоритме). Если байты заполнения необходимы, но алгоритм не определяет содержимое заполнителя, то октеты заполнения должны (MUST) иметь нулевое значение.

3.3.4. Фрагментация

Если необходимо, IP-фрагментация происходит после обработки AH в рамках реализации IPsec. Таким образом, AH транспортного режима применяется только к целым IP-дейтаграммам, а не к IP-фрагментам. Пакет IPv4, к которому применен AH, сам по себе может быть фрагментирован по пути маршрутизаторами, и такие фрагменты должны быть реассемблированы до обработки AH в получателе. (Это не применимо к IPv6, в котором отсутствует фрагментация, инициируемая маршрутизаторами). В туннельном режиме AH применяется к IP-пакету, блок данных которого может быть фрагментированным пакетом. Например, шлюз безопасности или реализация IPsec типа "вставки в стек" или "вставки в линию" (подробности см. в документе по архитектуре безопасности) может применить к таким фрагментам AH туннельного режима.

Примечание: Для транспортного режима – Как упоминалось в конце пункта 3.1.1, реализации типа "вставки в стек" и "вставки в линию" могут сначала реассемблировать пакет, фрагментированный локальным IP-уровнем, а затем применить IPsec и после этого фрагментировать результирующий пакет.

Примечание: Для IPv6 – Для реализаций типа "вставки в стек" и "вставки в линию" необходимо будет проверить все заголовки расширения, чтобы определить, не имеется ли заголовок фрагментации и, следовательно, такой пакет необходимо реассемблировать до обработки IPsec.

Фрагментация независимо от того, выполняется ли она реализацией IPsec или маршрутизаторами, находящимися на пути между партнерами IPsec, существенно снижает производительность. Более того, требование к получателю AH принимать фрагменты для реассемблирования создает угрозы доступности (отказа в обслуживании). Таким образом, реализация IPsec может (MAY) принять решение не поддерживать фрагментацию и может пометить передаваемые пакеты битом DF, чтобы способствовать определению Path MTU (PMTU). В любом случае, реализация AH должна (MUST) поддерживать формирование сообщений ICMP PMTU (или эквивалентную внутреннюю сигнализацию для естественных реализаций в хостах для минимизации вероятности фрагментации. Детали поддержки, требуемой для управления MTU, содержатся в документе по архитектуре безопасности.

3.4. Обработка входящих пакетов

Если имеется более одного заголовка IPsec/заголовка расширения, то обработка каждого из них игнорирует (не обнуляет, не использует) любые заголовки IPsec, применяемые впоследствии к обрабатываемому заголовку.

3.4.1. Реассемблирование

Если необходимо, реассемблирование выполняется до обработки AH. Если пакет, предлагаемый AH для обработки оказывается IP-фрагментом, например, поле OFFSET имеет ненулевое значение, или установлен флаг MORE FRAGMENTS, то получатель должен (MUST) отбросить пакет; это событие подлежит аудиту.

Элемент журнала аудита для такого события должен (SHOULD) включать значение SPI, дату и время, Source Address, Destination Address и Flow ID (в IPv6).

Примечание: Текущая спецификация IPv4 не требует ни обнуления поля OFFSET, ни сброса флага MORE FRAGMENTS при реассемблировании пакетов. Чтобы реассемблированный пакет был обработан IPsec (в отличие от отбрасываемого как бесспорный фрагмент), код IP должен делать эти две вещи после того, как он реассемблирует пакет.

3.4.2. Поиск контекста безопасности

После получения пакета, содержащего аутентифицирующий заголовок IP, получатель определяет соответствующий (однонаправленный) SA посредством поиска в SAD. Как описано в подразд. 2.4, для индивидуальных (unicast) SA этот поиск базируется на SPI, или на SPI и поле протокола. Если реализация поддерживает групповой (multicast) трафик, то, как описано в подразд. 2.4, при поиске используется также адрес места назначения (в дополнение к SPI) и, кроме того, может быть использован адрес отправителя. (Этот процесс описывается более подробно в документе по архитектуре безопасности). Элемент SAD для этого SA указывает также, будет ли контролироваться поле Sequence Number и применяются ли для этого SA 32- или 64-битовые порядковые номера. Кроме того, элемент SAD для этого SA определяет алгоритм (алгоритмы), используемый для вычисления ICV, и указывает ключ, требуемый для подтверждения правильности ICV.

Если для этого пакета ни одного годного контекста безопасности не существует, то получатель должен (MUST) отбросить пакет; это событие подлежит аудиту. Элемент журнала аудита для такого события должен (SHOULD) включать значение SPI, дату и время, Source Address, Destination Address и Flow ID (в IPv6).

(Заметим, что трафик управления SA, например, пакеты IKE, не требует обработки на основе SPI, т.е. этот трафик можно демультиплексировать отдельно на основе полей Next Protocol и Port).

3.4.3. Проверка порядкового номера

Все реализации AH должны (MUST) поддерживать сервис противодействия повторному воспроизведению, хотя его использование может разрешаться или не разрешаться получателем для каждого SA. Противодействие повторному воспроизведению применимо как к индивидуальным (unicast), так и к групповым (multicast) SA. Однако данный стандарт не специфицирует никаких механизмов для противодействия повторному воспроизведению для (индивидуальных или групповых) SA с несколькими отправителями. Как отмечено ниже, при отсутствии согласования (или при ручном конфигурировании) механизма противодействия повторному воспроизведению для такого SA рекомендуется, чтобы контроль порядковых номеров для такого SA был отключен в отправителе и получателе (путем согласования или ручного конфигурирования).

Если получатель не разрешает противодействие повторному воспроизведению для некоторого SA, то никакого входного контроля порядковых номеров не выполняется. Однако с точки зрения отправителя по умолчанию предполагается, что противодействие повторному воспроизведению разрешено в получателе. Чтобы отправителю избежать выполнения ненужного мониторинга порядковых номеров и начальной установки SA (см. п. 3.3.2 "Формирование порядковых номеров"), в случае применения протокола установления SA, например, IKE, получатель во время установления SA должен (SHOULD) уведомить отправителя о том, что он не будет обеспечивать защиту противодействия повторному воспроизведению.

Если получатель для данного SA разрешил сервис противодействия повторному воспроизведению, то во время установления контекста безопасности счетчик полученных пакетов этого SA должен (MUST) быть инициализирован нулем. Для каждого принятого пакета получатель должен (MUST) проверить, что этот пакет содержит порядковый номер, который не дублирует порядковый номер любого другого пакета, принятого в течение времени жизни этого SA. Для ускорения

отбрасывания дублированных пакетов это должен (SHOULD) быть первый контроль АН, применяемый к пакету после его сопоставления с SA.

Дубликаты отбрасываются с помощью скользящего приемного окна. Каким образом реализуется это окно, является локальным вопросом, но в последующем тексте описана функциональность, которую реализации должны демонстрировать.

"Правый" край окна представляет самое большое утвержденное значение порядкового номера, полученного по данному SA. Пакеты, содержащие порядковые номера, меньшие, чем "левый" край окна, отбрасываются. Пакеты, попадающие в окно, сравниваются со списком пакетов, принятых в рамках этого окна.

Если для некоторого SA выбрана опция ESN, то в явном виде передаются только младшие 32 бита порядкового номера, но получатель при сравнении принятого порядкового номера с окном приема использует полный порядковый номер, вычисленный с помощью старших 32 бит для указанного SA (из своего локального счетчика). При создании полного порядкового номера в случае, когда младшие 32 бита, передаваемые в пакете, оказываются меньше по значению, чем младшие 32 бита счетчика порядковых номеров получателя, получатель предполагает, что старшие 32 бита были инкрементированы, осуществив переход в новое подпространство порядковых номеров. (Этот алгоритм согласует разрывы в получении пакетов для одного SA размером до 2**32-1 пакет. Если возникает разрыв большего размера, то, как описано в приложении B, для повторной синхронизации счетчика порядковых номеров получателя могут (MAY) применяться дополнительные эвристические проверки).

Если принятый пакет попадает в рамки окна и не является дубликатом, или если пакет оказывается справа от окна, то получатель переходит к проверке ICV. Если проверка ICV не проходит, то получатель должен (MUST) отбросить принятую дейтаграмму как неправильную. Это событие подлежит аудиту. Элемент журнала аудита должен (SHOULD) включать значение SPI, дату и время, Source Address, Destination Address, Sequence Number и Flow ID (в IPv6). Окно приема корректируется, только если проверка ICV прошла успешно.

Минимальный размер окна должен (MUST) поддерживаться равным 32 пакетам, но размер окна в 64 пакета является предпочтительным и должен (SHOULD) применяться по умолчанию. Получатель может (MAY) выбрать другой размер окна (больший минимального). (Получатель не уведомляет отправителя о размере окна). Для высокоскоростной среды, независимо от проблем обеспечения, размер окна приема должен увеличиваться. Значения для минимальных и рекомендованных размеров окна приема для сверхскоростных устройств (например, работающих со скоростью несколько гигабит в секунду) в данном стандарте не специфицируются.

3.4.4. Проверка значения для контроля целостности

Получатель вычисляет ICV от соответствующих полей пакета, используя специфицированный алгоритм целостности, и проверяет, что он совпадает с ICV, включенным в поле ICV пакета. Детали вычисления представлены ниже.

Если вычисленное и принятое значения ICV совпадают, то дейтаграмма является действительной и принимается. Если проверка не прошла, то получатель должен (MUST) отбросить принятую IP-дейтаграмму как неправильную. Это событие подлежит аудиту. Элемент журнала аудита должен (SHOULD) включать значение SPI, дату и время получения, Source Address, Destination Address и Flow ID (в IPv6).

Примечание по реализации:

> Реализации могут использовать любую последовательность шагов, которые приведут к тому же результату, что и следующая последовательность шагов. Начать с сохранения значения ISV и с замены его нулем (но без какого-либо заполнения поля ICV). Обнулить все другие поля, которые по пути могли быть модифицированы. (Подробное обсуждение перечня полей, которые обнуляются

перед выполнением вычисления ICV, см. в пп. 3.3.3.1 "Обработка изменяемых полей").

Если для данного SA выбрана опция ESN, то присоединить 32 старших бита ESN после конца пакета. Проверить общую длину пакета (как описано выше) и, если требуется неявное заполнение, основанное на требованиях алгоритма целостности, то присоединить к концу пакета необходимое количество нулевых байтов (после ESN, если он имеется). Выполнить вычисление ICV и сравнить результат с сохраненным значением, используя правила сравнения, определенные спецификацией алгоритма. (Например, если для вычисления ICV используются цифровая подпись и однонаправленная хэш-функция, то процесс сопоставления является более сложным).

4. Аудит

Не все системы, реализующие AH, будут осуществлять аудит. Однако если AH встраивается в систему, которая поддерживает аудит, то реализация AH также должна (MUST) поддерживать аудит и должна (MUST) позволять системному администратору разрешать или не разрешать проводить аудит AH. Степень детализации аудита в основном является локальным вопросом. Однако в данной спецификации определено несколько подлежащих аудиту событий, и для каждого из этих событий определен минимальный набор информации, которая должна (SHOULD) быть включена в журнал аудита. Для каждого из этих событий в журнал аудита может (MAY) быть включена также дополнительная информация, и в элементах журнала аудита могут (MAY) находить отражение дополнительные события, не указанные в данной спецификации. К получателю не предъявляется требование передачи подразумеваемому отправителю какого-либо сообщения в ответ на обнаружение подлежащего аудиту события из-за возможности вызвать таким действием отказ в обслуживании.

5. Требования к соответствию

Реализации, которые заявляют о соответствии или о согласованности с настоящей спецификацией, должны (MUST) полностью выполнять синтаксис и обработку AH, описанную здесь для индивидуального (unicast) трафика и должны (MUST) соблюдать все требования документа по архитектуре безопасности [Ken-Arch]. Кроме того, если реализация заявляет о поддержке группового (multicast) трафика, она должна (MUST) соблюдать дополнительные требования, определенные для поддержки такого трафика. Если ключ, используемый для вычисления ICV, раздается вручную, то корректное обеспечение сервиса противодействия повторному воспроизведению потребует правильной поддержки состояния счетчика в отправителе до тех пор, пока ключ не будет заменен, и если приближается переполнение счетчика, то автоматическое восстановление вероятно осуществляться не будет. Таким образом, реализация, соответствующая настоящей спецификации, не должна (SHOULD NOT) обеспечивать такой сервис для SA, которые управляются вручную.

Чтобы упростить обновление требований к алгоритмам независимо от самого протокола, обязательные для использования в AH алгоритмы, описываются в отельном документе RFC [Eas04]. Помимо обязательных для AH алгоритмов, могут (MAY) поддерживаться дополнительные алгоритмы.

6. Анализ безопасности

Безопасность является центральным вопросом разработки данного протокола, и эти соображения безопасности пронизывают насквозь всю спецификацию. Дополнительные аспекты использования протокола IPsec, связанные с безопасностью, обсуждаются в документе по архитектуре безопасности.

7. Отличия от RFC документа 2402

Данный документ имеет следующие отличия от документа RFC 2402 [RFC2402]:

- SPI – модифицирован для определения унифицированного алгоритма поиска в SAD индивидуальных и групповых контекстов безопасности, покрывающего более широкий диапазон технологий многоадресной рассылки. Для индивидуальных контекстов безопасности SPI может использоваться для выбора SA сам по себе, или, по выбору получателя, в комбинации с протоколом. Для групповых контекстов безопасности для выбора SA SPI комбинируется с адресом места назначения и, по выбору, с адресом источника.
- Расширенные порядковые номера – добавлена новая опция 64-битового порядкового номера для передачи информации со сверхвысокими скоростями. Разъяснены требования к обработке отправителем и получателем для групповых SA и SA с несколькими отправителями.
- Ссылки на обязательные алгоритмы перенесены в отдельный документ [Eas04].

8. Благодарности

Автор хотел бы поблагодарить Рэна Аткинсона (Ran Atkinson), который играл главную роль в начальной деятельности по IPsec и был автором первой серии стандартов IPsec: RFC 1825-1827. Карен Сео (Karen Seo) заслуживает специальной благодарности за помощь в редактировании данной и предыдущей версий этой спецификации. Автор хотел бы также поблагодарить членов рабочих групп IPsec и MSEC, которые внесли свой вклад в разработку данной спецификации протокола.

9. Ссылки

9.1. Нормативные ссылки

[Bra97] Bradner, S., "Key words for use in RFCs to Indicate Requirement Level", BCP 14, RFC 2119, March 1997.

[DH98] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[Eas04] 3rd Eastlake, D., "Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 4305, December 2005.

[Ken-Arch] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC1108] Kent, S., "U.S. Department of Defense Security Options for the Internet Protocol", RFC 1108, November 1991.

9.2. Информативные ссылки

[AES] Advanced Encryption Standard (AES), Federal Information Processing Standard 197, National Institutes of Standards and Technology, November 26, 2001.

[HC03] Holbrook, H. and B. Cain, "Source Specific Multicast for IP", Work in Progress, November 3, 2002.

[IKEv2] Kaufman, C., Ed., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[Ken-ESP] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[NBBB98] Nichols, K., Blake, S., Baker, F., and D. Black,

"Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFB01] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[RFC1063] Mogul, J., Kent, C., Partridge, C., and K. McCloghrie, "IP MTU discovery options", RFC 1063, July 1988.

[RFC1122] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[RFC1385] Wang, Z., "EIP: The Extended Internet Protocol", RFC 1385, November 1992.

[RFC1393] Malkin, G., "Traceroute Using an IP Option", RFC 1393, January 1993.

[RFC1770] Graff, C., "IPv4 Option for Sender Directed Multi-Destination Delivery", RFC 1770, March 1995.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, February 1997.

[RFC2402] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[RFC3547] Baugher, M., Weis, B., Hardjono, T., and H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003.

[RFC3740] Hardjono, T. and B. Weis, "The Multicast Group Security Architecture", RFC 3740, March 2004.

Приложение A. Изменчивость опций/ заголовков расширения IP

A1. Опции IPv4

В данной таблице показана классификация опций IPv4 в отношении их "изменчивости". Там, где даются две ссылки, вторая ссылка отменяет первую. Эта таблица частично базируется на информации из документа RFC 1700 "ASSIGNED NUMBERS" (Октябрь 1994).

Copy	Класс	Номер опции	Имя	Ссылка
Неизменяемые – включены в вычисление ICV				
0	0	0	End of Options List	[RFC791]
0	0	1	No Operation	[RFC791]
1	0	2	Security	[RFC1108] (устарел, но используется)
1	0	5	Extended Security	[RFC1108] (устарел, но используется)
1	0	6	Commercial Security	
1	0	20	Router Alert	[RFC2113]
1	0	21	Sender Directed Multi-Destination Delivery	[RFC1770]
Изменяемые – обнуляются				
1	0	3	Loose Source Route	[RFC791]
0	2	4	Time Stamp	[RFC791]
0	0	7	Record Route	[RFC791]
1	0	9	Strict Source Route	[RFC791]
0	2	18	Traceroute	[RFC1393]
Экспериментальные, замененные – обнуляются				
1	0	8	Stream ID	[RFC791, RFC1122 (Треб. к хостам)]
0	0	11	MTU Probe	[RFC1063, RFC1191 (PMTU)]
0	0	12	MTU Reply	[RFC1063, RFC1191 (PMTU)]
1	0	17	Extended Internet Protocol	[RFC1385, DH98 (IPv6)]
0	0	10	Experimental Measurement	
1	2	13	Experimental Flow Control	
1	0	14	Experimental Access Ctl	
0	0	15	???	
1	0	16	IMI Traffic Descriptor	
1	0	19	Address Extension	

Примечание: Использование опции Router Alert потенциально несовместимо с использованием IPsec. Хотя опция является неизменяемой, ее использование предполагает, что каждый маршрутизатор, находящийся на пути пакета, будет "обрабатывать" пакет и, следовательно, может изменить пакет. Это будет происходить по принципу интервал за интервалом по мере передачи от одного маршрутизатору к другому. До обработки приложением, которому направляется содержимое опции (например, протоколами Resource Reservation Protocol (RSVP)/Internet Group Management Protocol (IGMP)), пакет должен столкнуться с обработкой AH. Однако обработка AH потребует, чтобы каждый маршрутизатор, находящийся на пути пакета, был членом группового SA, определяемого индексом параметров безопасности (SPI). Это может создать проблемы для пакетов, которые маршрутизируются от источника не строго, и это требует методов многоадресной рассылки, которые в настоящее время не доступны.

Примечание: Добавление или исключение меток безопасности (например, Basic Security Option (BSO), Extended Security Option (ESO), или Commercial Internet Protocol Security Option (CIPSO)) системами, находящимися на пути пакета, противоречит классификации этих опций IP как неизменяемых, и несовместимо с использованием IPsec.

Примечание: Опции End of Options List должны (SHOULD) при необходимости повторяться, чтобы гарантировать, что отсутствуют неспецифицированные байты, которые могли бы использоваться для скрытого канала.

A2. Заголовки расширения IPv6

В данной таблице показана классификация заголовков расширения IPv6 в отношении их "изменчивости".

Имя опции/заголовка расширения	Ссылка
Изменяемая, но предсказуемая – включена в вычисление ICV	
Routing (Type 0)	[DH98]
Является ли опция изменяемой (в процессе передачи изменяется непредсказуемо) определяет специальный бит	
Hop-by-Hop options	[DH98]
Destination options	[DH98]
Не применим	
Fragmentation	[DH98]

Опции – Опции IPv6 в заголовках расширения Hop-by-Hop Extension Header и Destination Extension Header содержат бит, который указывает, может ли в процессе передачи опция (непредсказуемо) изменяться. Для любой опции, содержимое которой может по пути измениться, при вычислении или при проверке ICV все поле "Option Data" должно рассматриваться как последовательность нулевых октетов. Поля Option Type and Opt Data Len включаются в вычисление ICV. Все опции, для которых этот бит указывает неизменность, включаются в вычисление ICV. Более подробную информацию см. в спецификации IPv6 [DH98].

Маршрутизация (тип 0) – Заголовок маршрутизации IPv6 Routing Header "Type 0" во время передачи от источника к месту назначения будет переустанавливать в пакете поля адресов. Однако содержимое пакета, которое появится в получателе, известно отправителю и всем промежуточным узлам. Поэтому заголовок расширения IPv6 Routing Header "Type 0" включается в вычисление ICV как изменяемый, но предсказуемый. До выполнения вычисления ICV отправитель должен установить в этом поле такой порядок, в котором оно появится в получателе.

Фрагментация – Фрагментация происходит после исходящей обработки IPsec (подразд. 3.3), а реассемблирование осуществляется до входящей обработки IPsec (подразд. 3.4). Таким образом, заголовок расширения Fragmentation Extension Header, если он имеется, не виден IPsec.

Заметим, что на принимающей стороне реализация IP, когда она выполняет реассемблирование, может оставить заголовок Fragmentation Extension Header на месте. Если это происходит, то когда AH принимает пакет, до выполнения обработки ICV, AH должен (MUST) "удалить" этот заголовок (или перескочить через него) и заменить поле "Next Header" в предыдущем заголовке на поле "Next Header", находящееся в заголовке Fragmentation Extension Header.

Заметим, что на передающем конце реализация IP может передать коду IPsec пакет с заголовком Fragmentation Extension Header с полем Offset равным 0 (первый фрагмент) и флагом More Fragments Flag равным 0 (последний фрагмент). Если это происходит, то перед выполнением обработки ICV, AH должен (MUST) сначала "удалить" этот заголовок (или перескочить через него) и заменить поле "Next Header" в предыдущем заголовке на поле "Next Header", находящееся в заголовке Fragmentation Extension Header.

Приложение B. Расширенные (64-битовые) порядковые номера

B1. Общее представление

В данном приложении описывается схема расширенных порядковых номеров (ESN - Extended Sequence Number), предназначенная для использования в IPsec (ESP и AH), в которой употребляются 64-битовые порядковые номера, но в каждом пакете передаются только младшие 32 бита. Оно охватывает схему окна, используемую для обнаружения повторно воспроизводимых пакетов, а также вопросы определения старших битов порядкового номера, которые используются как для отклонения повторно воспроизводимых пакетов, так и для вычисления ICV. В нем обсуждается также механизм для обработки потери синхронизации относительно старших (не передаваемых) бит.

B2. Окно противодействия повторному воспроизведению

Получатель поддерживает окно противодействия повторному воспроизведению размера W. Это окно устанавливает ограничение на то, насколько далеко неупорядоченный пакет может находиться по отношению к пакету с наибольшим порядковым номером, который был до этого аутентифицирован. (Для минимального или рекомендованного размеров этого окна не устанавливается никаких ограничений сверх 32 и 64 пакетов, которые уже установлены для окна с 32-битовыми порядковыми номерами. Однако предполагается, что разработчик реализации будет масштабировать эти значения в соответствии со скоростью интерфейса, поддерживаемой реализацией, использующей опцию ESN. Кроме того, описанный ниже алгоритм предполагает, что ширина окна не превосходит 2^31 пакетов). Все 2^32 порядковых номеров, связанные с любым значением старших 32 бит (Seqh) будут ниже называться подпространством порядковых номеров. В следующей таблице перечислены соответствующие переменные и их определения.

Имя переменной	Размер (бит)	Смысл
W	32	Размер окна
T	64	Наибольший порядковый номер, аутентифицированный до этого момента времени, верхняя граница окна
Tl	32	Младшие 32 бита T
Th	32	Старшие 32 бита T
B	64	Нижняя граница окна
Bl	32	Младшие 32 бита B
Bh	32	Старшие 32 бита B
Seq	64	Порядковый номер принятого пакета
Seql	32	Младшие 32 бита Seq
Seqh	32	Старшие 32 бита Seq

При выполнении контроля противодействия повторному воспроизведению или при определении того, какие старшие биты использовать для аутентификации входящего пакета, имеются два случая:

+ Случай A: Tl >= (W - 1). В этом случае окно находится в одном подпространстве порядковых номеров (см. рис. 1).
+ Случай B: Tl < (W - 1). В этом случае окно охватывает два подпространства порядковых номеров (см. рис. 2).

На представленных ниже рисунках нижняя линия ("----") показывает два последовательных подпространства порядковых номеров, на которой нули указывают начало каждого подпространства. Две более короткие линии, находящиеся над ней, показывают старшие применяющиеся биты. Последовательность символов "====" представляет окно. Последовательность символов "****" представляет будущие порядковые номера, например те, которые находятся за пределами аутентифицированного текущего наибольшего порядкового номера(ThTl).

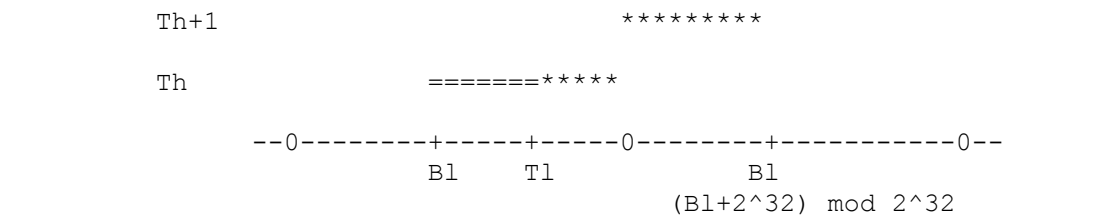


Рис. 1. Случай A.

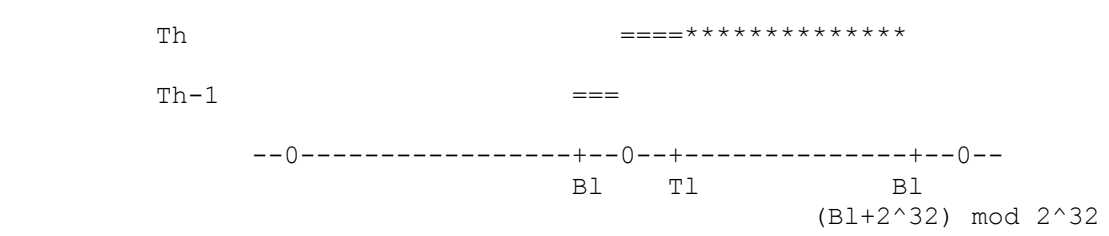


Рис. 2. Случай B.

B2.1. Управление и использование окна противодействия повторному воспроизведению.

Окно противодействия повторному воспроизведению можно рассматривать как строку битов, для которой 'W' определяет длину этой строки. W = T - B + 1 и не может превышать значения 2^32 - 1. Самые нижние биты соответствуют B, а самые верхние биты соответствуют T, и все порядковые номера от Bl до Tl представляются соответствующими битами. Значение бита указывает, был ли пакет с этим порядковым номером получен и аутентифицирован, так что повторные передачи могут быть обнаружены и отклонены.

Когда принимается и объявляется действительным пакет с 64-битовым порядковым номером (Seq), большим, чем T,

+ B увеличивается на (Seq - T)
+ (Seq - T) бит отбрасываются от нижнего конца окна
+ (Seq - T) бит добавляются к верхнему концу окна
+ верхний бит устанавливается для указания того, что пакет с этим порядковым номером был принят и аутентифицирован
+ новые биты в диапазоне от T до верхнего бита устанавливаются для указания того, что никаких пакетов с этими номерами еще не поступило
+ T устанавливается на новый порядковый номер

При контроле повторно воспроизводимых пакетов

+ В случае A: Если Seql >= Bl (где Bl = Tl - W + 1) AND Seql <= Tl, то проверить соответствующий бит в окне, чтобы увидеть, не встречался ли уже этот Seql. Если уже встречался, отбросить пакет. Если не встречался, выполнить контроль целостности (см. приложение B2.2 ниже для определения SeqH).

+ В случае B: Если Seql >= Bl (где Bl = Tl - W + 1) OR Seql <= Tl, то проверить соответствующий бит в окне, чтобы увидеть, не встречался ли уже этот Seql. Если уже встречался, отбросить пакет. Если не встречался, выполнить контроль целостности (см. приложение B2.2 ниже для определения SeqH).

B2.2. Определение старших бит (Seqh) порядкового номера

Поскольку с пакетом будет пересылаться только 'Seql', то получатель должен сделать заключение и отследить подпространство порядковых номеров, в которое попадает каждый пакет, т.е. определить значение Seqh. Следующие уравнения

определяют, как отбирать Seqh в "нормальных условиях"; обсуждение того, как восстанавливаться от чрезмерной потери пакетов см. в приложении B3.

+ В случае A (рис. 1):
  Если Seql >= Bl (где Bl = Tl - W + 1), то Seqh = Th
  Если Seql < Bl (где Bl = Tl - W + 1), то Seqh = Th + 1

+ В случае B (рис. 2):
  Если Seql >= Bl (где Bl = Tl - W + 1), то Seqh = Th - 1
  Если Seql < Bl (где Bl = Tl - W + 1), то Seqh = Th

B2.3. Пример на псевдокоде

Описанные выше алгоритмы противодействия повторному воспроизведению и контроля целостности иллюстрируются следующим псевдокодом. Значения 'Seql', 'Tl', 'Th' и 'W' являются 32-битовыми целыми числами без знака. Арифметика выполняется по mod 2^32.

```
        If (Tl >= W - 1)                            Case A
            If (Seql >= Tl - W + 1)
                Seqh = Th
                If (Seql <= Tl)
                    If (pass replay check)
                        If (pass integrity check)
                            Set bit corresponding to Seql
                            Pass the packet on
                        Else reject packet
                    Else reject packet
                Else
                    If (pass integrity check)
                        Tl = Seql (shift bits)
                        Set bit corresponding to Seql
                        Pass the packet on
                    Else reject packet
            Else
                Seqh = Th + 1
                If (pass integrity check)
                    Tl = Seql (shift bits)
                    Th = Th + 1
                    Set bit corresponding to Seql
                    Pass the packet on
                Else reject packet
        Else                                        Case B
            If (Seql >= Tl - W + 1)
                Seqh = Th - 1
                If (pass replay check)
                    If (pass integrity check)
                        Set the bit corresponding to Seql
                        Pass packet on
                    Else reject packet
                Else reject packet
            Else
                Seqh = Th
                If (Seql <= Tl)
                    If (pass replay check)
                        If (pass integrity check)
                            Set the bit corresponding to Seql
                            Pass packet on
                        Else reject packet
                    Else reject packet
                Else
                    If (pass integrity check)
                        Tl = Seql (shift bits)
                        Set the bit corresponding to Seql
```

```
                    Pass packet on
                Else reject packet
```

B3. Обработка потери синхронизации из-за значительной потери пакетов

Если имеется необнаруженная потеря 2^32 или большего количества последовательных пакетов по одному SA, то передатчик и приемник потеряют синхронизацию старших битов, т.е. уравнения приложения B2.2. не смогут дать правильное значение. Если эта проблема не будет обнаружена и решена, последующие пакеты по этому SA не пройдут контроль аутентификации и будут отброшены. Любая реализация IPsec (ESP или AH), которая поддерживает опцию ESN, должна (SHOULD) выполнять следующую процедуру.

Заметим, что этот вид длительной потери трафика вероятно не может происходить, если любая существенная часть трафика по данному SA является TCP, поскольку источник не сможет получать подтверждения (ACKs) и приостановит передачу задолго до того, как будут потеряны 2^32 пакетов. Кроме того, для любого двунаправленного приложения, даже работающего через UDP, такой длительный простой вероятно приведет к запуску некоторого вида тайм-аута. Однако в однонаправленном приложении, работающем через UDP, может отсутствовать обратная связь, которая станет причиной автоматического обнаружения потери такого масштаба, отсюда мотивация разработки метода восстановления для такого случая.

Решение, которому мы отдали предпочтение, было выбрано так, чтобы:

+ минимизировать влияние на нормальную обработку трафика.
+ избежать создания возможности проведения новой атаки на доступность, например такой, которая может возникнуть из-за возможности атакующим злоумышленником отвлекать ресурсы на процесс повторной синхронизации.
+ ограничить механизм восстановления приемником, поскольку противодействие повторному воспроизведению является сервисом только для приемника, а передатчик в общем случае не осведомлен о том, использует ли приемник порядковые номера для поддержки этого дополнительного сервиса. Предпочтительно, чтобы механизм восстановления был локальным по отношению к приемнику. Это учитывает также обратную совместимость.

B3.1. Запуск повторной синхронизации

Для каждого SA получатель регистрирует количество последовательных пакетов, которые не прошли аутентификацию. Это сумма используется для запуска процесса повторной синхронизации, который должен выполняться в фоновом режиме или используя отдельный процессор. Прием по SA действительного пакета обнуляет этот счетчик. Значение счетчика, используемое для запуска процесса повторной синхронизации, является локальным вопросом. Отсутствует требование поддержки индивидуальных значений для разных SA, хотя разработчик реализации может принять такое решение.

B3.2. Процесс повторной синхронизации

Когда достигается указанная выше точка запуска, выбирается "плохой" пакет, для которого снова выполняется аутентификация посредством последовательно больших значений старшей половины порядкового номера (Seqh). Эти значения формируются путем инкрементирования на единицу для каждой попытки. Количество попыток должно быть ограничено на случай, если это пакет из "прошлого" или подложный пакет. Выбор предельного значения попыток является локальным вопросом. (Поскольку значение Seqh неявно помещается после блока данных AH (или ESP), эту процедуру можно оптимизировать путем выполнения над пакетом алгоритма целостности вплоть до конечной точки блока данных, затем вычислить значения различных предполагаемых ISV путем варьирования значения of Seqh). Успешная аутентификация пакета с помощью этой процедуры сбрасывает счетчик последовательных неудач и устанавливает значение T равным значению порядкового номера принятого пакета.

Такое решение требует поддержки только в части приемника, тем саамы предоставляя возможность для обратной совместимости. Кроме того, поскольку попытки повторной синхронизации будут либо осуществляться в фоновом режиме, либо использовать дополнительный процессор, такое решение не влияет на обработку трафика, и атака на доступность не может отвлекать ресурсы от обработки трафика.

Адрес автора

Stephen Kent
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
USA

Phone: +1 (617) 873-3988
EMail: kent@bbn.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.