Российская Академия Наук
Институт системного программирования

IP Encapsulating Security Payload (ESP)

Протокол инкапсулирующей защиты данных на уровне IP

**RFC 4303 “ IP Encapsulating Security Payload (ESP)”
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 4303, регламентирующего формат и правила обработки зашифрованных сообщений 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: 4303
Obsoletes: 2406
Category: Standards Track

S. Kent
BBN Technologies
December 2005

IP Encapsulating Security Payload (ESP)

Протокол инкапсулирующей защиты данных 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 инкапсулирующей защиты данных (ESP - Encapsulating Security Payload), который разработан с целью предоставления смеси сервисов безопасности в IPv4 и IPv6. ESP используется для обеспечения конфиденциальности, аутентификации источника данных, целостности в режиме без установления соединения, сервиса противодействия повторному воспроизведению и ограниченной конфиденциальности потоков трафика. Данный документ отменяет действие документа RFC 2406 (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].

Заголовок инкапсулирующей защиты данных (ESP) разработан для предоставления смеси сервисов безопасности в IPv4 и IPv6 [DH98]. Протокол ESP может применяться самостоятельно, в комбинации с протоколом AH [Ken-AH], 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. Более подробную информацию об использовании ESP и AH в различной сетевой среде см. в документе по архитектуре безопасности [Ken-Arch].

Заголовок ESP вставляется после IP-заголовка и до заголовка следующего уровня (в транспортном режиме) или перед инкапсулированным IP-заголовком (в туннельном режиме). Ниже эти режимы описываются более подробно.

Протокол ESP может использоваться для обеспечения конфиденциальности, аутентификации источника данных, целостности в режиме без установления соединения, сервиса противодействия повторному воспроизведению (вид частичной целостности последовательности) и (ограниченной) конфиденциальности потоков трафика. Набор предоставляемых сервисов зависит от опций, выбранных в процессе установления контекста безопасности (SA - Security Association), и от местоположения реализации в топологии сети.

Для обеспечения конфиденциальности протокол ESP допускает использовать только шифрование. Однако необходимо заметить, что в общем случае такой подход обеспечит защиту только от пассивных атакующих злоумышленников. Использование шифрования без стойкого механизма обеспечения целостности, работающего поверх него (либо в ESP, либо отдельно с помощью AH), может сделать сервис конфиденциальности незащищенным от некоторых видов активных атак [Bel96, Kra01]. Более того, базовый сервис целостности, например AH, примененный до шифрования, не обязательно защищает от активных атак конфиденциальность, реализуемую только шифрованием [Kra01]. ESP допускает применение контекстов безопасности, осуществляющих только шифрование, поскольку такой сервис может предложить значительно более высокую производительность и все еще обеспечить адекватную безопасность, например, когда независимо от него аутентификация/защита целостности предлагается на более высоких уровнях. Однако данный стандарт не требует от реализаций ESP предложения только сервиса шифрования.

Аутентификация источника данных и целостность в режиме без установления соединения - это объединенные сервисы, которые сообща далее будут называться "целостностью". (Этот термин применяется потому что вычисление, выполняемое для каждого пакета, непосредственно обеспечивает целостность в режиме без установления соединения; аутентификация источника данных обеспечивается косвенно в результате привязки ключа, используемого для проверки целостности, к идентификатору партнера IPsec. Обычно такая привязка осуществляется благодаря использованию общего симметричного ключа). Протокол ESP, реализующий только целостность, должен (MUST) предлагаться в качестве опции выбора сервиса, например, он должен согласовываться протоколами управления SA и должен (MUST) конфигурироваться посредством интерфейсов управления. Во многих контекстах протокол ESP, реализующий только целостность, представляет собой привлекательную

альтернативу протоколу AH, например, поскольку его быстрее обрабатывать и он лучше поддается конвейеризации во многих реализациях.

Хотя конфиденциальность и целостность могут предлагаться независимо, обычно совместно с ESP будут применять оба этих сервиса, т.е. пакеты будут защищаться как в отношении конфиденциальности, так и в отношении целостности. Таким образом, имеется три возможных комбинации сервисов безопасности ESP, затрагивающие эти сервисы:

- только конфиденциальность (может (MAY) поддерживаться)
- только целостность (должна (MUST) поддерживаться)
- конфиденциальность и целостность (должна (MUST) поддерживаться)

Сервис противодействия повторному воспроизведению может выбираться для некоторого контекста безопасности (SA) только если для данного SA выбран сервис целостности. Выбор этого сервиса полностью отдан на усмотрение приемника и, таким образом, не должен согласовываться. Однако для того чтобы расширенные порядковые номера использовались в интероперабельной форме, протокол ESP устанавливает требование к протоколам управления SA такое, что они должны быть способными согласовывать эту возможность (см. ниже п. 2.2.1).

В общем случае сервис конфиденциальности потоков трафика (TFC - traffic flow confidentiality) оказывается эффективным только если ESP применяется в такой форме, которая скрывает конечные адреса источника и места назначения корреспондентов, например, в туннельном режиме между защитными шлюзами, и только если значительные потоки трафика между партнерами IPsec (либо естественным образом, либо в результате создания маскирующего трафика) скрывают характеристики конкретных потоков трафика отдельных абонентов. (ESP может применяться как часть системы TFC более высокого уровня, например, луковичная маршрутизация [Syverson], но такие системы выходят за рамки данного стандарта). Новые возможности TFC, представляемые ESP, упрощают эффективное формирование и отбраковку фиктивного трафика, а также лучшее заполнение реального трафика таким способом, который обеспечивает обратную совместимость.

В разд. 7 дается короткий обзор отличий данного документа от документа RFC 2406.

2. Формат пакета инкапсулирующей защиты данных

(Внешний) заголовок протокола (IPv4, IPv6, или заголовок расширения), непосредственно предшествующий заголовку ESP должен (SHALL) содержать значение 50 в поле Protocol (IPv4) или Next Header (IPv6, заголовка расширения) (см. web-страницу IANA по адресу http://www.iana.org/assignments/protocol-numbers). На рис. 1 представлен высокоуровневый формат пакета ESP. Пакет начинается с двух четырехбайтовых полей (поля Security Parameters Index (SPI) и поля Sequence Number). За этими полями следует поле Payload Data, которое имеет собственную подструктуру, зависящую от выбора алгоритма шифрования и режима, а также от использования заполнения TFC, и рассматривается более подробно далее. За полем Payload Data следуют поля Padding и Pad Length, а также поле Next Header. Завершает пакет необязательное поле Integrity Check Value (ICV).

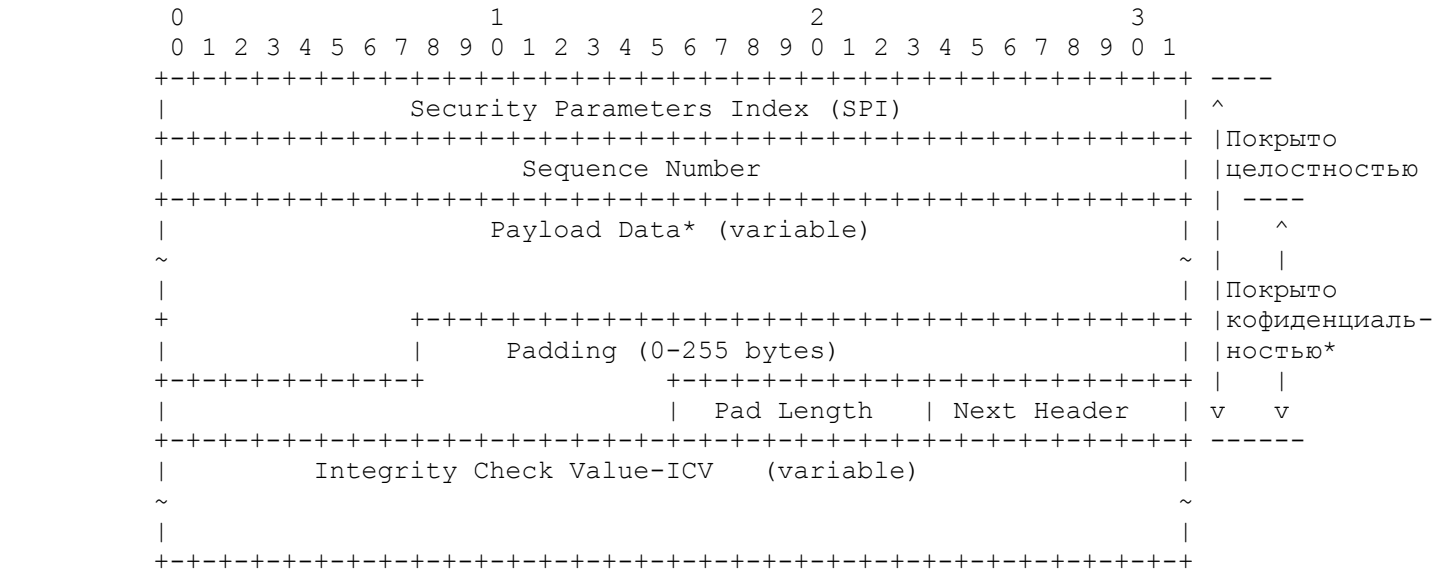


Рис. 1. Высокоуровневый формат пакета ESP

   * Данные для криптографической синхронизации, например, вектор инициализации (IV - Initialization Vector, см. подразд. 2.3), в случае их включения в поле Payload, обычно сами по себе не зашифровываются, хотя часто о них говорится как о части зашифрованного текста.

(Передаваемая) заключительная часть пакета (трейлер) ESP состоит из полей Padding, Pad Length, и Next Header. Как описано ниже, в вычисление целостности включаются дополнительные неявные данные заключительной части пакета ESP (которые не передаются).

Если выбран сервис целостности, то вычисление целостности охватывает поля SPI, Sequence Number, Payload Data и заключительную часть ESP (явную и неявную).

Если выбран сервис конфиденциальности, то зашифрованный текст включает поле Payload Data (за исключением данных для криптографической синхронизации, которые могут быть включены в это поле) и (явную) заключительную часть ESP.

Как отмечалось выше, поле Payload Data может иметь собственную подструктуру. Алгоритм шифрования, требующий явного вектора инициализации (IV), например, режим сцепления блоков шифра (CBC - Cipher Block Chaining), часто присоединяет к блоку данных вектор инициализации в виде префикса, чтобы он был защищен этим значением. Некоторые режимы алгоритмов объединяют шифрование и целостность в одну операцию; в данном документе такие режимы алгоритмов называются "алгоритмами комбинированного режима". Чтобы приспособиться к алгоритмам комбинированного режима необходимо, чтобы сам алгоритм в явном виде описывал подструктуру блока данных, используемую для передачи данных целостности.

Некоторые алгоритмы комбинированного режима обеспечивают целостность только тем данным, которые зашифрованы, в то время как другие алгоритмы могут обеспечивать целостность некоторых дополнительных данных, которые не зашифровываются для передачи. Поскольку обеспечение целостности полей SPI и Sequence Number является требованием сервиса целостности, но сами они не зашифровываются, необходимо гарантировать, чтобы всякий раз, когда выбирается этот сервис, этим полям обеспечивалась целостность независимо от применяемого типа алгоритма комбинированного режима.

Когда применяется какой-либо алгоритм комбинированного режима, предполагается, что сам алгоритм возвращает как расшифрованный открытый текст, так и указание на то, прошел ли, или не прошел контроль целостности. Для алгоритмов комбинированного режима поле ICV, которое обычно появляется в конце пакета ESP (когда выбирается целостность), может быть опущено. Когда поле ICV опускается и

выбирается целостность, то ответственность за кодирование в поле Payload Data средства для проверки целостности пакета ложится на алгоритм комбинированного режима.

Если алгоритм комбинированного режима обеспечивает целостность только данным, которые зашифровываются, необходимо продублировать поля SPI и Sequence Number в поле Payload Data.

Наконец, для обеспечения конфиденциальности потоков трафика предоставляется новая возможность вставки заполнения после поля Payload Data и перед заключительной частью пакета ESP. Такая подструктура поля Payload Data показана на рис. 2. (Примечание: На этой диаграмме показаны "биты на линии". Так что даже в случае использования расширенных порядковых номеров, передаваться будут только 32 бита порядкового номера (см. п. 2.2.1)).

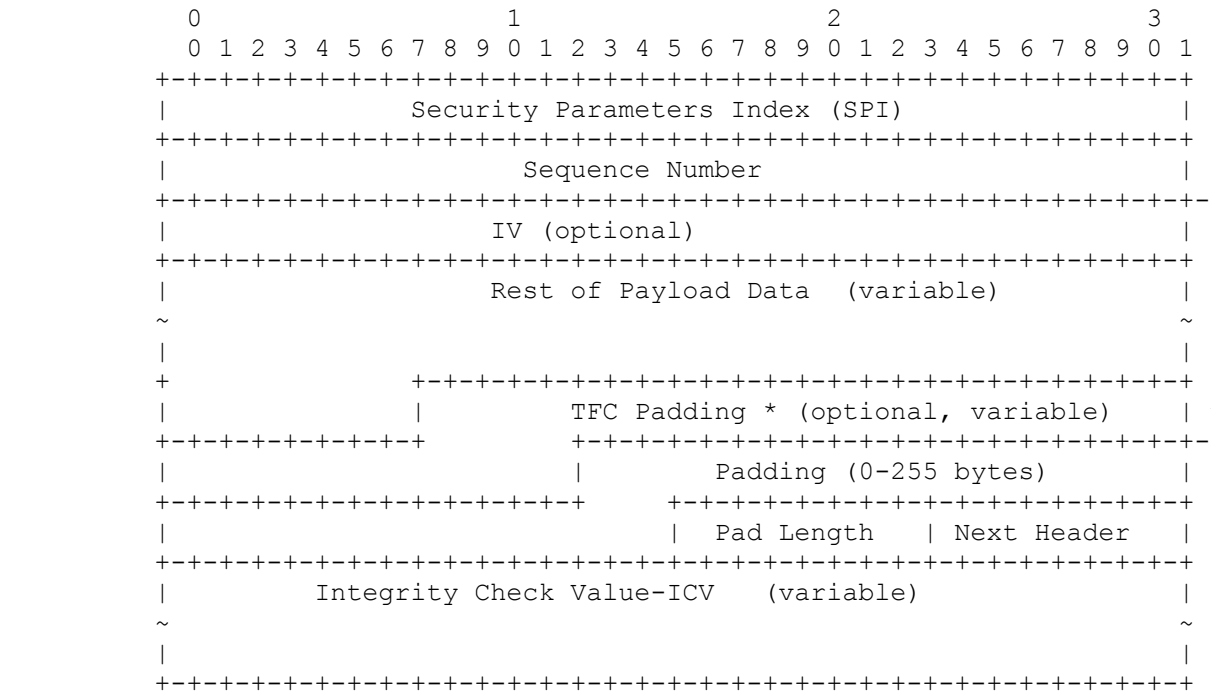


Рис. 2. Подструктура блока данных

* Если используется туннельный режим, то для обеспечения конфиденциальности потоков трафика (TFC - Traffic Flow Confidentiality) реализация IPsec после поля Payload Data и перед полем Padding (0-255 байтов) может добавить заполнитель (TFC padding) (см. подразд. 2.4).

Если применяется алгоритм комбинированного режима, то явное поле ICV, показанное на рис. 1 и 2, может быть опущено (см. ниже пп. 3.3.2.2). Поскольку алгоритмы и режимы фиксируются когда устанавливается SA, детальный формат пакетов ESP для данного SA (включая подструктуру поля Payload Data) фиксируется для всего трафика, передающегося по этому SA.

Представленные ниже таблицы ссылаются на поля упомянутых ранее рисунков и иллюстрируют как некоторые категории опций алгоритмов, каждая из которых имеет свою особую модель обработки, влияют на указанные выше поля. Детали обработки описываются в последующих подразделах.

Таблица 1. Отдельные алгоритмы шифрования и целостности

	Количество байтов	Обяза-тельность [1]	Что покрывает шифрование	Что покрывает целостность	Что передается	
SPI	4	M		Да	откр.текст	
Seq# (младшие биты)	4	M		Да	откр.текст	p
						a
IV	переменное	O		ДА	откр.текст	y
IP datagram [2]	переменное	M или D	Да	Да	шифр[3]	l
TFC padding [4]	переменное	O	Да	Да	шифр[3]	o
						a
Padding	0-255	M	Да	Да	шифр[3]	d
Pad Length	1	M	Да	Да	шифр[3]	
Next Header	1	M	Да	Да	шифр[3]	
Seq# (старшие биты)	4	если ESN [5]		Да	не передается	
ICV Padding	переменное	если необходимо		Да	не передается	
ICV	переменное	M [6]			откр.текст	

[1] M = mandatory (обязательное);
O = optional (необязательное);
D = dummy (фиктивное)
[2] Если туннельный режим -> IP-дейтаграмма
Если транспортный режим -> следующий заголовок и данные
[3] Зашифрованный текст, если выбрано шифрование
[4] Может использоваться только если блок данных специфицирует свою "действительную" длину
[5] См. п. 2.2.1
[6] Обязательное, если используется отдельный алгоритм целостности

Таблица 2. Алгоритмы комбинированного режима

	Количество байтов	Обяза-тельность [1]	Что покрывает шифрование	Что покрывает целостность	Что передается	
SPI	4	M			откр.текст	
Seq# (младшие биты)	4	M			откр.текст	p
						a
IV	переменное	O		ДА	откр.текст	y
IP datagram [2]	переменное	M или D	Да	Да	шифр	l
TFC padding [4]	переменное	O	Да	Да	шифр	o
						a
Padding	0-255	M	Да	Да	шифр	d
Pad Length	1	M	Да	Да	шифр	
Next Header	1	M	Да	Да	шифр	
Seq# (старшие биты)	4	если ESN [4]		Да	[5]	
ICV Padding	переменное	если необходимо		Да	[5]	
ICV	переменное	O [6]			откр.текст	

[1] M = mandatory (обязательное);
O = optional (необязательное);
D = dummy (фиктивное)
[2] Если туннельный режим -> IP-дейтаграмма
Если транспортный режим -> следующий заголовок и данные
[3] Может использоваться только если блок данных специфицирует свою "действительную" длину
[4] См. п. 2.2.1
[5] Выбор алгоритма определяет передаются ли они, но в любом случае результат невидим для ESP
[6] Спецификация алгоритма определяет, присутствует ли данное поле

В следующих подразделах описываются поля в формате заголовка. Признак "Optional" означает, что данное опускается, если эта опция не выбрана, т.е. оно не передается в пакете или не участвует в вычислении ICV (см подразд. 2.7). В процессе установления контекста безопасности определяется, выбирается или не выбирается соответствующая опция. Таким образом, формат пакетов ESP для данного SA фиксируется на все время жизни этого SA. В отличие от необязательных полей, "обязательные" поля всегда присутствуют в формате пакета ESP для всех контекстов безопасности.

Примечание: Все криптографические алгоритмы, используемые в IPsec, предполагают, что их входные данные поступают в каноническом сетевом порядке байтов, и формируют свои выходные данные в каноническом сетевом порядке байтов. IP-пакеты также передаются в сетевом порядке байтов.

Пакет ESP не содержит номера версии, поэтому если возникает беспокойство относительно обратной совместимости, то оно должно (MUST) закрываться посредством механизма сигнализации между двумя партнерами IPsec, чтобы гарантировать совместимость версий ESP (например, с помощью протокола Internet Key Exchange (IKEv2) [Kau05]) или посредством отдельного механизма конфигурирования.

2.1. Поле Security Parameters Index (SPI)

Индекс параметров безопасности (SPI) представляет собой произвольное 32-битовое значение, которое используется получателем для идентификации SA, к которому привязывается входящий пакет. Поле SPI является обязательным.

Для индивидуального (unicast) контекста безопасности SPI может использоваться сам по себе для определения SA, или он может использоваться вместе с типом протокола IPsec (в данном случае с ESP). 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. Этот механизм отображения входного трафика на индивидуальные SA должен (MUST) поддерживаться всеми реализациями ESP.

Если реализация IPsec поддерживает многоадресную рассылку (multicast), то она должна (MUST) для отображения входящих IPsec-дейтаграмм на SA 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 центральный 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 групповую 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, используется 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 (что указывается в элементе SAD), 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 некоторый элемент SAD дает совпадение, то обработать входящий пакет ESP 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 некоторый элемент SAD дает совпадение, то обработать входящий пакет ESP 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 некоторый элемент SAD дает совпадение, то обработать входящий пакет ESP в соответствии с этим совпадающим элементом SAD. В противном случае отбросить пакет и записать в журнал событие, подлежащее аудиту.

На деле реализация может (MAY) выбрать любой метод для ускорения этого поиска, 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 групповую рассылку для конкретных источников (SSM - Source-Specific Multicast) [HC03], используют трехэлементный кортеж идентификатора SA, состоящий из SPI, группового адреса места назначения и адреса источника. Групповой SA для произвольных источников (Any-Source Multicast) в качестве идентификатора 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 RFC. Нулевое (0) значение SPI 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).

2.2. 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. ESP не предоставляет никаких средств для синхронизации счетчиков пакетов между

несколькими отправителями или для осмысленного управления счетчиком пакетов получателя и окном в контексте нескольких отправителей. Таким образом, для SA с несколькими отправителями свойства ESP по противодействию повторному воспроизведению не доступны (см. пункты 3.3.3 и 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 ESP должны (MUST) быть способны выполнять обработку, описанную в п. 3.3.3 и в п. 3.4.3. Таким образом, отправитель должен (MUST) передавать это поле, но получатель не обязан им руководствоваться. (См. обсуждение обработки порядковых номеров ниже в подразделе "Обработка входящих пакетов" п. 3.4.3).

Счетчик отправителя и счетчик получателя инициализируются нулем при установлении SA. (Первый пакет, посланный с помощью данного SA, будет иметь порядковый номер 1; см. дополнительные детали порождения порядковых номеров в п. 3.3.3). 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.

2.2.1. Расширенные (64-битовые) порядковые номера

Для поддержки высокоскоростных реализаций IPsec расширенные порядковые номера (ESN - Extended Sequence Number) должны (SHOULD) реализовываться в качестве расширения текущего 32-битового поля порядкового номера. 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. (Подробности см. в приложении A "Расширенные (64-битовые) порядковые номера"). В открытом тексте заголовка ESP каждого пакета передаются только младшие 32 бита порядкового номера. Старшие 32 бита являются частью счетчика порядковых номеров, поддерживаются как передатчиком, так и приемником, и включаются в вычисление ICV (если выбран сервис целостности). Если применяется отдельный алгоритм целостности, то старшие биты включаются в неявную заключительную часть ESP, но не передаются, аналогично битам заполнителя алгоритма целостности. Если применяется алгоритм комбинированного режима, то выбор алгоритма определяет, передаются ли старшие биты ESN, или они включаются неявно в вычисление. Детали обработки см. в пп. 3.3.2.2.

2.3. Поле Payload Data

Блок данных (Payload Data) – это поле переменной длины, содержащее данные (из первоначального IP-пакета), описываемые полем Next Header. Поле Payload Data является обязательным и его длина измеряется целым количеством байтов. Если алгоритм, используемый для шифрования блока данных, требует данных для криптографической синхронизации, например, вектора инициализации (IV - Initialization Vector), то эти данные передаются в явном виде в поле Payload Data, но в ESP не рассматриваются как отдельное поле, т.е. передача явного вектора инициализации для ESP не видна (см. рис. 2). Любой алгоритм шифрования, требующий таких явных данных синхронизации для каждого пакета, должен (MUST) указывать длину, какую-либо структуру для таких данных, а также местоположение этих данных как часть документа RFC, специфицирующего использование этого алгоритма совместно с ESP. (Обычно IV находится непосредственно перед зашифрованным текстом, см. рис.2). Если такие данные синхронизации являются неявными, то алгоритм извлечения этих данных должен (MUST) быть частью документа RFC, определяющего алгоритм. (В случае включения данных для криптографической синхронизации, например, вектора инициализации, в поле Payload Data, они обычно

сами не зашифровываются (см. таблицы 1 и 2), хотя иногда говорят, что они являются частью зашифрованного текста).

Заметим, что начало заголовка протокола следующего уровня должно (MUST) быть выровнено по отношению к началу заголовка ESP следующим образом. Для IPv4 такое выравнивание должно быть кратно четырем байтам. Для IPv6 выравнивание должно быть кратно восьми байтам.

В отношении обеспечения выравнивания (реального) зашифрованного текста при наличии IV заметим следующее:

- В некоторых режимах работы, базирующихся на IV, приемник рассматривает IV как начало зашифрованного текста, подавая его прямо на вход алгоритма. В этих режимах выравнивание начала (реального) зашифрованного текста не является проблемой в приемнике.

- В некоторых случаях приемник читает IV отдельно от зашифрованного текста. В этих случаях выполнение выравнивания (реального) зашифрованного теста должно (MUST) определяться в спецификации алгоритма.

2.4. Поле Padding (для шифрования)

Поле Padding (заполнение) требуется или мотивируется двумя основными факторами.

- Если применяется алгоритм шифрования, который требует чтобы открытый текст был кратен некоторому количеству байтов, например, размеру блока в блочном шифре, то поле Padding используется для дополнения открытого текста (включающего поля Payload Data, Padding, Pad Length и Next Header) до размера, требуемого алгоритмом.

- Кроме того, заполнитель может потребоваться независимо от требований алгоритма шифрования для того чтобы гарантировать, что получающийся зашифрованный текст заканчивается на четырехбайтовой границе. В частности, как проиллюстрировано выше на рисунках формата пакета ESP, поля Pad Length и Next Header должны быть правильно выровнены в четырехбайтовом слове, чтобы обеспечить выравнивание поля ICV (если оно имеется) по четырехбайтовой границе.

Заполнение сверх длины, требуемой алгоритмом или определяемой указанными выше причинами выравнивания, может использоваться для сокрытия действительной длины блока данных при поддержке TFC. Однако описываемое поле Padding слишком ограничено для эффективного применения в целях TFC и, поэтому, не должно использоваться для этой цели. Вместо этого, когда требуется TFC, должен использоваться отдельный описанный ниже механизм (см. подразд. 2.7).

Отправитель может (MAY) добавить от 0 до 255 байтов заполнения. Включение поля Padding в пакет ESP не является обязательным (это предмет указанных выше требований), но все реализации должны (MUST) поддерживать формирование и присоединение заполнения.

- С целью обеспечения того, чтобы количество бит, которые должны быть зашифрованы, было кратно размеру блока алгоритма (первый маркер абзаца выше), вычисление длины заполнителя применяется к полю Payload Data, исключая какой-либо IV, но включая поля заключительной части ESP. Если режим комбинированного алгоритма для обеспечения целостности требует передачи полей SPI и Sequence Number, например, дублирования полей SPI и Sequence Number в поле Payload Data, то дублированные версии этих элементов данных и любые связанные с обеспечением целостности данные, эквивалентные ICV, включаются в вычисление длины заполнителя. (Если выбирается опция ESN, то старшие 32 бита ESN тоже включаются в вычисление, если алгоритм комбинированного режима требует их передачи для обеспечения целостности).

- С целью обеспечения того, чтобы ICV был выровнен по четырехбайтовой границе (второй маркер абзаца выше), вычисление длины заполнителя применяется к полю Payload Data, включая поля IV, Pad Length и Next Header. Если применяется алгоритм комбинированного режима, то все дублированные данные, а также данные, эквивалентные ICV, включаются в Payload Data, покрываемое вычислением длины заполнителя.

Если байты заполнителя необходимы, но алгоритм шифрования не специфицирует его содержимое, то по умолчанию должна (MUST) использоваться следующая обработка. Байты заполнителя инициализируются последовательностью однобайтовых целых значений без знака. Первый байт заполнителя, присоединяемый к открытому тексту, нумеруется единицей, а последующие байты заполнителя образуют монотонно возрастающую последовательность: 1, 2, 3, ... Когда применяется такая схема заполнения, приемник должен (SHOULD) инспектировать поле Padding. (Такая схема была выбрана из-за ее простоты, несложности реализации в аппаратуре, а также потому, что она предлагает ограниченную защиту от определенных видов атак типа "вырезать и вставить" при отсутствии других мер обеспечения целостности, если приемник контролирует значения заполнителя после дешифрования).

Если алгоритм шифрования или алгоритм комбинированного режима налагает ограничения на значения байтов, используемых для заполнения, они должны (MUST) быть специфицированы в документе RFC, определяющем применение алгоритма совместно с ESP. Если алгоритм требует контроля значений байтов, используемых для заполнения, это также должно (MUST) специфицироваться в этом документе RFC.

2.5. Поле Pad Length

Поле Pad Length (длина заполнения) указывает количество байтов заполнения, которые непосредственно предшествуют ему в поле Padding. Диапазон действительных значений меняется от 0 до 255, где значение ноль указывает, что байты заполнения отсутствуют. Как указывалось выше, это поле не включает байты заполнения TFC. Поле Pad Length является обязательным.

2.6. Поле Next Header

Поле Next Header (следующий заголовок) является обязательным и представляет собой 8-битовое поле, которое определяет тип данных, содержащихся в поле Payload Data, например, пакет IPv4 или IPv6, или заголовок и данные следующего уровня. 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.

Для упрощения быстрого формирования и отбрасывания заполняющего (фиктивного) трафика при поддержке конфиденциальности потоков трафика (см. подразд. 2.4) для обозначения "фиктивного" пакета, должно (MUST) использоваться значение протокола 59 (которое означает "следующий заголовок отсутствует"). Передатчик должен (MUST) быть способен формировать фиктивные пакеты, помеченные этим значением в поле следующего протокола, а приемник должен (MUST) быть готов такие пакеты отбрасывать без указания ошибки. Все другие поля заголовка и заключительной части ESP (SPI, Sequence Number, Padding, Pad Length, Next Header и ICV) в фиктивных пакетах должны (MUST) присутствовать, но отсутствует необходимость в правильном построении открытой части блока данных, за исключением поля Next Header, например, остальная часть поля Payload Data может содержать только случайные байты. Фиктивные пакеты отбрасываются без каких-либо сомнений.

Чтобы позволить использовать эту возможность для каждого отдельного SA, реализации должны (SHOULD) обеспечивать локальные директивы управления. Эти директивы должны позволять пользователю определять, должна ли эта возможность использоваться, а также предоставлять директивы с параметрами; например, директива может позволять администратору формировать фиктивные пакеты случайной или фиксированной длины.

ОБСУЖДЕНИЕ: Чтобы маскировать отсутствие реального трафика, фиктивные пакеты могут вставляться со случайными интервалами. Можно также "формировать" реальный трафик так, чтобы обеспечить соответствие некоторому распределению, добавляя фиктивный трафик в соответствии с параметрами распределения. Также как и с возможностью заполнения длины пакета для безопасности потоков трафика (TFS), наиболее безопасным подходом является формирование фиктивных пакетов с любой скоростью, которая требуется для поддержки постоянной скорости по SA. Если все пакеты имеют один и тот же размер, то SA создает видимость потока данных с постоянной битовой скоростью, аналогично тому, что предлагают канальные шифраторы на уровнях 1 и 2. Однако во многих контекстах это будет вероятно непрактично, например, когда имеется несколько активных SA, поскольку это предполагает для сайта сокращение допустимой пропускной способности в зависимости от количества SA, и будет подрывать преимущества пакетной коммутации. Реализации должны (SHOULD) предоставлять директивы, позволяющие администраторам управлять формированием фиктивных пакетов для целей TFC.

2.7. Поле Traffic Flow Confidentiality (TFC) Padding

Как отмечалось выше, поле Padding ограничено длиной в 255 байтов. Вообще такая длина заполнения не адекватна требованиям конфиденциальности потоков трафика (TFC) для сокрытия характеристик трафика. Для удовлетворения требований TFC в блоке данных специально предоставляется необязательное поле.

Реализация IPsec должна (SHOULD) иметь возможность заполнять трафик путем добавления байтов после конца поля Payload Data, но до начала поля Padding. Однако это заполнение (ниже называемое заполнением TFC) может добавляться только если поле Payload Data содержит спецификацию длины IP-дейтаграммы. Это всегда справедливо в туннельном режиме и может быть справедливо в транспортном режиме в зависимости от того, содержит ли протокол следующего уровня (например, IP, UDP, ICMP) явную информацию о длине. Эта информация о длине позволит приемнику отбросить заполнитель TFC, поскольку истинная длина поля Payload Data будет известна. (Поля заключительной части ESP размещаются отсчитывая назад от конца пакета ESP. Соответственно, если добавляется заполнитель TFC, то поле, содержащее спецификацию длины IP-дейтаграммы, не должно (MUST NOT) модифицироваться для отражения этого заполнения. В данном стандарте не устанавливается никаких требований к значению этого заполнения.

В принципе, существующие реализации IPsec могли использовать в прозрачной манере такую возможность и раньше. Однако, поскольку приемники могут быть не готовы иметь дело с таким заполнением, для обеспечения обратной совместимости протокол управления SA должен (MUST) согласовать этот сервис до того момента, когда передатчик начнет его использовать. Вместе с соглашением об использовании идентификатора протокола 59, которое описано выше в подразд. 2.6, для поддержки TFC реализация ESP способна формировать фиктивные и реальные пакеты, которые демонстрируют значительно большую изменчивость длины.

Реализации должны (SHOULD) предоставлять директивы локального управления, чтобы позволить использовать эту возможность для каждого отдельного SA. Эти директивы должны позволять пользователю определять, должна ли эта возможность использоваться, а также предоставлять для этой возможности директивы с параметрами.

2.8. Поле Integrity Check Value (ICV)

Значение для контроля целостности представляет собой поле переменной длины, при вычислении которого используется заголовок ESP, а также поля Payload Data и заключительная часть ESP. Неявные поля заключительной части ESP (заполнитель целостности и старшие биты ESN, если применяются) включаются в вычисление ICV. Поле ICV является необязательным. Оно присутствует только если выбран сервис целостности и обеспечивается либо отдельным алгоритмом целостности, либо алгоритмом комбинированного режима, который использует ICV. Длина поля специфицируется алгоритмом целостности, выбранным и ассоциированным с SA. Спецификация алгоритма целостности должна (MUST) определять длину ICV, правила сравнения, а также шаги обработки для проверки правильности.

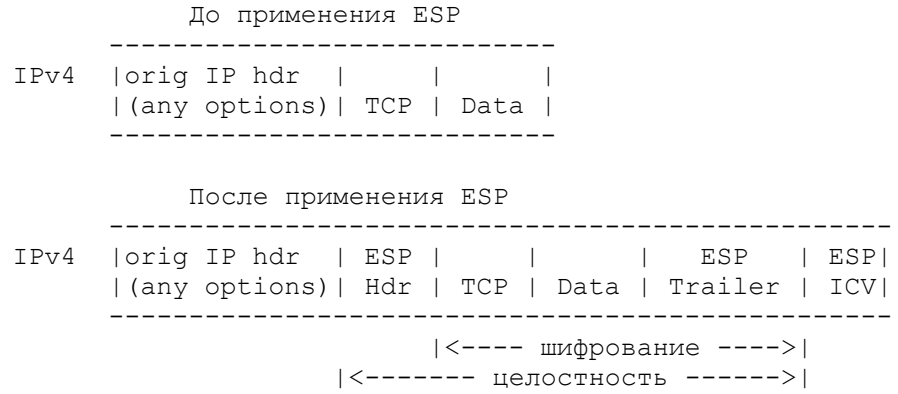
3. Обработка протокола инкапсулирующей защиты

3.1. Местоположение заголовка ESP

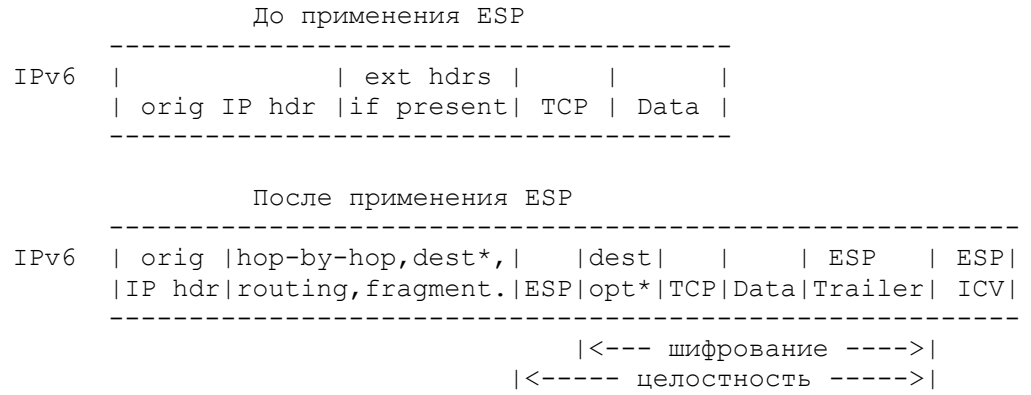
ESP может применяться в двух известных режимах: в транспортном режиме или в туннельном режиме.

3.1.1. Обработка транспортного режима

В транспортном режиме ESP вставляется после IP-заголовка и перед протоколом следующего уровня, например, TCP, UDP, ICMP и т.д.). В контексте IPv4 это требует помещения ESP 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 ESP транспортного режима для типового пакета IPv4 по принципу "до и после" его применения. (На этой и последующей диаграммах этого пункта показано поле ICV, наличие которого определяется выбором сервиса безопасности, а также алгоритма/режима).



В контексте IPv6 ESP 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 ESP в зависимости от требуемой семантики. Однако поскольку ESP защищает только поля, следующие за заголовком ESP, то обычно желательно помещать заголовок (заголовки) расширения опций места назначения после заголовка ESP. Следующая диаграмма иллюстрирует выбор местоположения ESP транспортного режима для типового пакета IPv6.



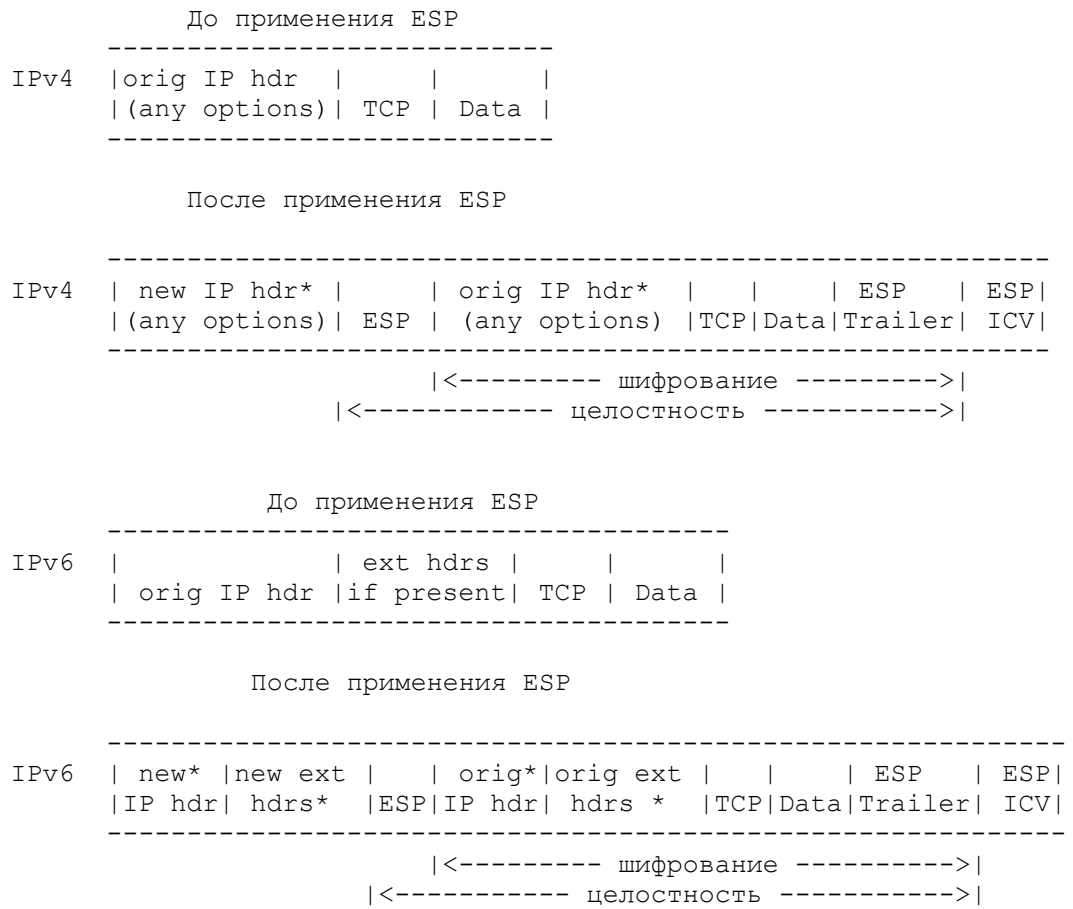
* = Если присутствует, может находиться до ESP, после ESP, или и до и после ESP

Заметим, что в транспортном режиме для реализаций типа "вставки в стек" или "вставки в линию", как определено в документе по архитектуре безопасности, входящие и исходящие IP-фрагменты могут потребовать от реализации IPsec

выполнения дополнительного реассемблирования или дополнительной фрагментации, чтобы обеспечивать соответствие данной спецификации и прозрачную поддержку IPsec. В случае использования нескольких интерфейсов для выполнения таких операций в таких реализациях требуется особая тщательность.

3.1.2. Обработка туннельного режима

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. IPv6 поверх IPv4 и IPv4 поверх IPv6. В туннельном режиме ESP защищает весь внутренний IP-пакет, включая весь внутренний IP-заголовок. Месторасположение ESP в туннельном режиме по отношению к внешнему IP-заголовку то же самое, что и для ESP в транспортном режиме. Следующая диаграмма иллюстрирует выбор местоположения ESP 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

Чтобы упростить обновление требований к алгоритмам независимо от самого протокола, обязательные для реализации алгоритмы, используемые в ESP, описываются в отдельном документе RFC. Могут (MAY) поддерживаться и дополнительные алгоритмы, выходящие за рамки обязательных для ESP. Заметим, что хотя как конфиденциальность, так и целостность являются необязательными, по крайней мере должен (MUST) быть выбран один из этих сервисов, поэтому оба алгоритма не должны быть (MUST NOT) одновременно нулевыми ("NULL").

3.2.1. Алгоритмы шифрования

Алгоритм шифрования, используемый для защиты пакета ESP, определяется SA, по которому пакет посылается/принимается. Поскольку IP-пакеты могут прибывать неупорядоченно (потеря пакетов), для обеспечения дешифрирования каждый пакет должен содержать данные, необходимые приемнику для установления криптографической синхронизации. Эти данные могут передаваться явно в поле блока данных, например, вектор инициализации (как описано выше), или эти данные могут быть извлечены из открытой части заголовка пакета (внешнего IP или ESP). (Заметим, что если для извлечения IV используется информация открытого заголовка, то эта информация может стать критически важной для безопасности, и таким образом граница защиты, связанная с процессом шифрования, может расшириться. Например, если для извлечения IV используется порядковый номер ESP, то логика формирования порядковых номеров должна определяться в виде части реализации алгоритма шифрования. В случае FIPS 140-2 [NIST01] это может существенно расширить область действия определения криптографического модуля). Поскольку ESP способствует обеспечению заполнения открытого текста, алгоритмы шифрования, применяемые в ESP, могут демонстрировать характеристики либо блочного, либо потокового режимов. Заметим, что поскольку шифрование (конфиденциальность) может быть (MAY) факультативным сервисом (например, ESP, обеспечивающий только целостность), то этот алгоритм может быть (MAY) нулевым ("NULL") [Ken-Arch].

Чтобы позволить реализациям ESP вычислять заполнение для шифрования, требуемое алгоритмом шифрования в блочном режиме, а также определять влияние на алгоритм MTU, документ RFC для каждого алгоритма шифрования, используемого в ESP, должен определять модуль заполнения для этого алгоритма.

3.2.2. Алгоритмы целостности

Алгоритм целостности, применяемый для вычисления ICV, определяется контекстом безопасности по которому передается/принимается пакет. Как и в случае алгоритмов шифрования, любой алгоритм целостности, применяемый в ESP, должен предусматривать меры для обработки пакетов, которые прибывают неупорядоченно, и для приспособления к потере пакетов. То же самое указанное выше предостережение, применяется к использованию любых открытых данных, упрощающих синхронизацию приемником алгоритмов целостности. Заметим, что поскольку сервис целостности может (MAY) быть факультативным, этот алгоритм может быть нулевым ("NULL").

Чтобы позволить реализации ESP вычислить требуемое неявное заполнение алгоритмов целостности, документ RFC для каждого алгоритма, используемого в ESP, должен определять модуль заполнения для этого алгоритма.

3.2.3. Алгоритмы комбинированного режима

Если применяется алгоритм комбинированного режима, то предоставляется как сервис конфиденциальности, так и сервис целостности. Как и в случае алгоритмов шифрования, алгоритм комбинированного режима должен предусматривать меры для криптографической синхронизации каждого пакета, чтобы обеспечить дешифрирование пакетов, которые поступают неупорядоченно, и приспособиться к потере пакетов. Средство, с помощью которого алгоритм комбинированного режима обеспечивает целостность блока данных, а также для полей SPI и (Extended) Sequence Number, варьируется при выборе разных алгоритмов. Чтобы обеспечить унифицированный, независимый от алгоритма подход для вызова алгоритмов комбинированного режима, подструктура блока данных не определяется. Например, поля SPI и Sequence Number могут дублироваться внутри конверта с зашифрованным текстом, а ICV быть присоединен к заключительной части ESP. Ни один из этих элементов не должен быть виден снаружи.

Чтобы позволить реализации ESP определять влияние на MTU алгоритма комбинированного режима, документ RFC для каждого алгоритма, используемого в ESP, должен специфицировать (простую) формулу, которая рассчитывает размер

зашифрованного блока данных как функцию от размера открытого блока данных и размера порядкового номера.

3.3. Обработка исходящих пакетов

В транспортном режиме отправитель инкапсулирует информацию протокола следующего уровня между полями заголовка ESP и заключительной части ESP и сохраняет заданный IP-заголовок (а также любые заголовки расширения IP в контексте IPv6). В туннельном режиме внешний и внутренний IP-заголовок/заголовки расширения могут быть взаимосвязаны различными способами. Построение внешнего IP-заголовка/заголовков расширения в процессе инкапсуляции описывается в документе по архитектуре безопасности.

3.3.1. Поиск контекста безопасности

ESP применяется к исходящему пакету только после того как реализация IPsec определит, что этот пакет связан с SA, который требует обработки ESP. 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. Шифрование пакета и вычисление значения для контроля целостности (ICV)

В данном пункте мы говорим о том, что из-за последствий форматирования шифрование применяется всегда. Это делается в предположении, что "отсутствие конфиденциальности" предлагается с помощью нулевого алгоритма шифрования (RFC 2410). Имеется несколько алгоритмических возможностей.

3.3.2.1. Отдельные алгоритмы конфиденциальности и целостности

Если применяются отдельные алгоритмы конфиденциальности и целостности, то отправитель действует следующим образом:

1. Инкапсулирует (в поле Payload ESP):
   - для транспортного режима – только исходную информацию протокола следующего уровня.
   - для туннельного режима – всю исходную IP-дейтаграмму.

2. Добавляет необходимое заполнение – необязательные заполнение TFC и заполнение для шифрования

3. Зашифровывает результат, используя ключ, алгоритм шифрования и режим алгоритма, специфицированные для данного SA, а также любые данные, необходимые для криптографической синхронизации.

   - Если для криптографической синхронизации указываются явные данные, например, IV, они подаются на вход алгоритма шифрования согласно спецификации алгоритма и помещаются в поле Payload.

   - Если для криптографической синхронизации используются неявные данные, они создаются и подаются на вход алгоритма шифрования согласно спецификации алгоритма.

   - Если выбирается целостность, то сначала, до применения алгоритма целостности, выполняется шифрование, и это шифрование не охватывает поле ICV. Такой порядок обработки упрощает приемнику быстрое обнаружение и отклонение повторно воспроизводимых или фальшивых пакетов до выполнения дешифрирования пакетов, и поэтому потенциально уменьшает влияние атак на доступность (DoS - denial of service). Он допускает также возможность параллельной обработки пакетов в приемнике, т.е. дешифрирование может выполняться параллельно с контролем целостности. Заметим, что поскольку ICV не

защищен шифрованием, то для вычисления ICV должен применяться управляемый ключом алгоритм целостности.

4. Вычисляет ICV над пакетом ESP за минусом поля ICV. Таким образом вычисление ICV охватывает поля SPI, Sequence Number, Payload Data, Padding (если имеется), Pad Length и Next Header. (Заметим, что последние четыре поля будут представлены в зашифрованном виде, поскольку шифрование выполняется первым). Если для SA разрешается опция ESN, то для целей этого вычисления после поля Next Header присоединяются, но не передаются, старшие 32 бита порядкового номера.

Для некоторых алгоритмов целостности строка байтов, над которой выполняется вычисление ICV, должна быть кратной размеру блока, специфицированному алгоритмом. Если длина пакета ESP (описанного выше) не удовлетворяет требованиям алгоритма по размеру блока, то к концу пакета ESP должно (MUST) быть присоединено неявное заполнение. (Это заполнение добавляется после поля Next Header или после старших 32 бит порядкового номера, если выбирается ESN). Размер блока (а, следовательно, и длина заполнителя) указывается в спецификации алгоритма целостности. Это заполнение вместе с пакетом не передается. Чтобы определить, требуется ли описанное выше неявное заполнение, необходимо (MUST) обратиться к документу, определяющему алгоритм целостности. Если этот документ не дает ответ на поставленный вопрос, то по умолчанию предполагается, что неявное заполнение требуется (поскольку это необходимо для установления соответствия между длиной пакета и размером блока алгоритма). Если байты заполнения необходимы, но алгоритм не специфицирует содержимое заполнителя, то октеты заполнения должны (MUST) иметь нулевое значение.

3.3.2.2. Комбинированные алгоритмы конфиденциальности и целостности

Если применяется комбинированный алгоритм конфиденциальности/целостности, то отправитель действует следующим образом:

1. Инкапсулирует в поле Payload Data ESP:
   - для транспортного режима – только исходную информацию протокола следующего уровня.
   - для туннельного режима – всю исходную IP-дейтаграмму.

2. Добавляет необходимое заполнение – необязательные заполнение TFC и заполнение для шифрования

3. Зашифровывает результат и защищает его целостность, используя ключ, алгоритм комбинированного режима, специфицированный для данного SA, и любые данные, необходимые для криптографической синхронизации.

   - Если для криптографической синхронизации указываются явные данные, например, IV, они подаются на вход алгоритма комбинированного режима согласно спецификации алгоритма и помещаются в поле Payload.

   - Если для криптографической синхронизации используются неявные данные, они создаются и подаются на вход алгоритма шифрования согласно спецификации алгоритма.

   - Порядковый номер (или, при необходимости, расширенный порядковый номер) и SPI подаются на вход алгоритма, поскольку они должны быть включены в вычисление значения для контроля целостности. Способ, посредством которого эти значения включаются в это вычисление, является функцией применяемого алгоритма комбинированного режима и, таким образом, не специфицируется в данном стандарте.

   - Когда применяется алгоритм комбинированного режима,(явное) поле ICV может (MAY) быть частью формата пакета ESP. Если оно не используется, аналогичное поле обычно будет частью зашифрованного блока данных. Местоположение любого поля контроля целостности и способ, посредством которого порядковый номер и SPI включаются в

вычисление целостности, должны (MUST) быть определены в документе RFC, который определяет использование алгоритма комбинированного режима в ESP.

3.3.3. Формирование порядковых номеров

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). Таким образом, обычное поведение реализации ESP требует от отправителя установления нового SA, когда порядковый номер (или ESN) зацикливается, или упреждения зацикливания этого значения.

Если ключ, используемый для вычисления ICV распространяется вручную, реализация, соответствующая данному стандарту, не должна (SHOULD NOT) предоставлять сервис противодействия повторному воспроизведению. Если пользователь решит применять противодействие повторному воспроизведению вместе с SA, которые управляются вручную, то счетчик порядковых номеров в отправителе должен (MUST) поддерживаться в правильном состоянии вопреки локальным перезагрузкам и т.п. до тех пор, пока не будет заменен ключ (см. разд. 5).

Если противодействие повторному воспроизведению запрещается (как указано выше), отправителю не надо наблюдать или обнулять этот счетчик. 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), то в поле Sequence Number передаются только 32 младших бита порядкового номера, хотя как отправитель, так и получатель поддерживают полные 64-битовые счетчики. Старшие 32 бита включаются в контроль целостности способом, определяемым конкретным алгоритмом/режимом, например, когда применяется отдельный алгоритм целостности, старшие 32 бита могут добавляться после поля Next Header.

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.

3.3.4. Фрагментация

Если необходимо, фрагментация выполняется после обработки ESP в рамках реализации IPsec. Таким образом, ESP транспортного режима применяется только к целым IP-дейтаграммам (а не к IP-фрагментам). IP-пакет, к которому применен ESP,

сам по себе может быть фрагментирован по пути маршрутизаторами, и такие фрагменты должны быть реассемблированы до обработки ESP в получателе. В туннельном режиме ESP применяется к IP-пакету, который может быть фрагментом IP-дейтаграммы. Например, шлюз безопасности или реализация IPsec типа "вставки в стек" или "вставки в линию" (как определено в документе по архитектуре безопасности) может применить ESP туннельного режима к таким фрагментам.

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 ESP 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 упростить определение Path MTU (PMTU). В любом случае, реализация ESP 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

3.4.1. Реассемблирование

Если необходимо, реассемблирование выполняется до обработки ESP. Если пакет, предлагаемый ESP 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: Для реассемблирования пакетов текущая спецификация IPv4 не требует либо обнуления поля OFFSET, либо сброса флага MORE FRAGMENTS. 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 заголовок ESP, получатель определяет соответствующий (однонаправленный) SA посредством поиска в SAD. Для определения индивидуальных (unicast) SA этот поиск базируется на SPI, или на SPI и поле протокола, как описано в подразд. 2.1. Если реализация поддерживает групповой (multicast) трафик, при поиске используется также адрес места назначения (в дополнение к SPI), и может быть использован также адрес отправителя, как описано в подразд. 2.1. (Этот процесс описывается более подробно в документе по архитектуре безопасности). Элемент SAD для этого SA указывает также, будет ли контролироваться поле Sequence Number, применяются ли для этого SA 32- или 64-битовые порядковые номера и должно ли присутствовать (явное поле) ICV. Кроме того, элемент SAD будет определять алгоритмы и ключи, которые должны использоваться для дешифрирования и вычисления ICV (в случае необходимости).

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 получения пакета, Source Address, Destination Address, Sequence Number и открытый Flow ID (в IPv6).

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

3.4.3. Проверка порядковых номеров

Все реализации ESP должны (MUST) поддерживать сервис противодействия повторному воспроизведению, хотя его использование может разрешаться или не разрешаться получателем для каждого SA. Этот сервис не должен (MUST NOT) разрешаться, если для этого SA не разрешен также сервис целостности ESP, поскольку в противном случае поле Sequence Number не будет защищено целостностью. 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 SA (индивидуальных или групповых) с несколькими отправителями. При отсутствии согласования (или при ручном конфигурировании) механизма противодействия повторному воспроизведению для таких SA рекомендуется, чтобы контроль порядковых номеров для таких SA был отключен в отправителе и получателе (путем согласования или ручного конфигурирования), как отмечено ниже.

Если получатель не разрешает противодействие повторному воспроизведению для некоторого SA, не выполняется никакого входного контроля порядковых номеров. 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. 3.3.3), в случае применения протокола установления SA, получатель должен (SHOULD) уведомить отправителя во время установления SA о том, что получатель не будет обеспечивать защиту противодействия повторному воспроизведению.

Если получатель разрешил сервис противодействия повторному воспроизведению для данного SA, то во время установления SA счетчик полученных пакетов для 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 ESP, применяемый к пакету после его сопоставления с SA.

ESP допускает двухэтапную проверку порядковых номеров пакетов. Эта возможность оказывается важной всякий раз, когда реализация ESP (обычно часть ее криптографического модуля) не способна выполнять дешифрирование и/или контроль целостности с той же скоростью, что и интерфейс (интерфейсы) к незащищенным сетям. Если реализация может поддерживать работу с такой "скоростью линии", то нет необходимости в выполнении описанного ниже предварительного этапа проверки.

Предварительный контроль порядковых номеров осуществляется используя значение Sequence Number в заголовке ESP и выполняется до контроля целостности и дешифрирования. Если этот предварительный контроль не проходит, то пакет опускается, тем самым позволяя избежать необходимости выполнения приемником каких-либо криптографических операций. Если предварительный контроль порядковых номеров проходит успешно, то получатель все еще не может модифицировать значение своего локального счетчика, поскольку в этот момент еще не проверена целостность порядкового номера.

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

"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, проверяются по списку принятых в рамках окна пакетов. 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 больший разрыв, то, как описано в приложении, 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, и если применяется отдельный алгоритм целостности, то получатель переходит к проверке целостности. Если применяется алгоритм комбинированного режима, то контроль целостности осуществляется во время дешифрирования. В любом случае, если проверка целостности 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. (Если используется алгоритм комбинированного режима, то защищенный целостностью порядковый номер должен также совпадать с порядковым номером, который используется для защиты от повторного воспроизведения).

Когда применяются 32-битовые порядковые номера, минимальный размер окна должен (MUST) поддерживаться равным 32 пакетам; размер окна в 64 пакета является предпочтительным и должен (SHOULD) применяться по умолчанию. Получателем может (MAY) быть выбран другой размер окна (больший минимального). (Получатель не уведомляет отправителя о размере окна). Размер окна приема должен увеличиваться для высокоскоростной среды независимо от проблем с обеспечением. 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

Как и в случае исходящей обработки, для входной обработки имеется несколько возможностей в зависимости от свойств применяемых алгоритмов.

3.4.4.1. Отдельные алгоритмы конфиденциальности и целостности

Если применяются отдельные алгоритмы конфиденциальности и целостности, то обработка происходит следующим образом:

1. Если выбрана целостность, то получатель вычисляет ICV над пакетом минус ICV, 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, Sequence Number и открытый FlowID (в IPv6).

   Примечание по реализации:

   Реализации могут использовать любую последовательность шагов, которые приведут к тому же результату, что и следующая последовательность шагов.

Начать с удаления и сохранения значения поля ISV. Затем проверить общую длину пакета ESP минус поле ISV. Если требуется неявное заполнение, основанное на размере блока алгоритма целостности, то присоединить нулевые байты к концу пакета ESP непосредственно вслед за полем Next Header, или после старших 32 бит порядкового номера, если выбраны ESN. Выполнить вычисление ICV и сравнить результат с сохраненным значением, используя правила сравнения, определенные спецификацией алгоритма.

2. Получатель расшифровывает поля ESP Payload Data, Padding, Pad Length и Next Header, используя указанные в SA ключ, алгоритм шифрования, режим алгоритма и данные для криптографической синхронизации (если они имеются). Как и п. 3.3.2, мы говорим здесь о том, что из-за последствий форматирования всегда применяется шифрование. Это делается в предположении, что "отсутствие конфиденциальности" предлагается с помощью нулевого алгоритма шифрования (RFC 2410).

   - Если для криптографической синхронизации указываются явные данные, например, IV, они берутся из поля Payload и поступают на вход алгоритма дешифрирования согласно спецификации алгоритма.

   - Если для криптографической синхронизации указываются неявные данные, то создается локальная версия IV и подается на вход алгоритма дешифрирования согласно спецификации алгоритма.

3. Получатель обрабатывает любое заполнение, как определено в спецификации алгоритма шифрования. Если применялась схема заполнения по умолчанию (см. подразд. 2.4), получатель должен (SHOULD) проверить поле Padding до удаления заполнения прежде чем передать расшифрованные данные следующему уровню.

4. Получатель проверяет поле Next Header. Если его значение равно "59" (следующий заголовок отсутствует), то (фиктивный) пакет отбрасывается без дальнейшей обработки.

5. Получатель восстанавливает исходную IP-дейтаграмму:

   - для транспортного режима – из внешнего IP-заголовка плюс исходной информации протокола следующего уровня в поле ESP Payload
   - для туннельного режима – из всей IP-дейтаграммы в поле Payload.

   Точные шаги для восстановления первоначальной дейтаграммы зависят от режима (транспортный или туннельный) и описываются в документе по архитектуре безопасности. Как минимум, в контексте IPv6, получатель должен (SHOULD) гарантировать, что расшифрованные данные выровнены по 8-байтовой границе, чтобы упростить обработку протоколом, указанным в поле Next Header. Эта обработка "отбрасывает" все (факультативное) заполнение TFC, которое было добавлено для обеспечения конфиденциальности потоков трафика. (Если имеется, оно было вставлено после IP-дейтаграммы (или кадра транспортного уровня) и перед полем Padding (см. подразд. 2.4)).

Если контроль целостности и дешифрирование выполняются параллельно, то контроль целостности должен (MUST) завершиться до передачи расшифрованного пакета для последующей обработки. Такой порядок обработки упрощает быстрое обнаружение и отбрасывание приемником повторно воспроизведенных или фальшивых пакетов до дешифрирования пакета и, следовательно, потенциально снижает влияние атак на доступность.

Примечание: Если приемник выполняет дешифрирование параллельно с контролем целостности, то необходимо проявить внимание, чтобы исключить возможное состояние гонок в отношении обращения к пакету и извлечения расшифрованного пакета.

3.4.4.2. Комбинированные алгоритмы конфиденциальности и целостности

Если применяется комбинированный алгоритм конфиденциальности и целостности, то получатель действует следующим образом:

1. Расшифровывает и осуществляет контроль целостности полей ESP Payload Data, Padding, Pad Length и Next Header, используя указанные в SA ключ, алгоритм, режим алгоритма и данные для криптографической синхронизации (если таковые имеются). Поле SPI из заголовка ESP и значение счетчика пакетов (получателя) (установленное в соответствии с требованиями обработки, описанными в п. 3.4.3) подаются на вход этого алгоритма в соответствии с требованиями контроля целостности.

    - Если для криптографической синхронизации указываются явные данные, например, IV, они берутся из поля Payload и поступают на вход алгоритма дешифрирования согласно спецификации алгоритма.

    - Если для криптографической синхронизации указываются неявные данные, то создается локальная версия IV и подается на вход алгоритма дешифрирования согласно спецификации алгоритма.

2. Если контроль целостности, выполняемый алгоритмом комбинированного режима не проходит, то получатель должен (MUST) отбросить полученную IP-дейтаграмму как неправильную; это событие подлежит аудиту. Данные журнала аудита должны (SHOULD) включать значение SPI, дату и время получения пакета, Source Address, Destination Address, Sequence Number и открытый Flow ID (в IPv6).

3. Обрабатывает любое заполнение, как определено в спецификации алгоритма шифрования, если алгоритм этого еще не сделал.

4. Получатель проверяет поле Next Header. Если его значение равно "59" (следующий заголовок отсутствует), то (фиктивный) пакет отбрасывается без дальнейшей обработки.

5. Восстанавливает исходную IP-дейтаграмму (туннельный режим) или кадр транспортного уровня (транспортный режим) из поля ESP Payload Data. Этот процесс неявно отбрасывает любое (необязательное) заполнение, которое было добавлено для обеспечения конфиденциальности потоков трафика. (Если заполнение TFC присутствует, оно было вставлено после блока данных IP и перед полем Padding (см. подразд. 2.4)).

4 Аудит

Не все системы, реализующие ESP, будут осуществлять аудит. Однако, если ESP встраивается в систему, которая поддерживает аудит, то реализация ESP также должна (MUST) поддерживать аудит и должна (MUST) позволять системному администратору разрешать или не разрешать проводить аудит ESP. По большей части степень детализации аудита является локальным вопросом. Однако несколько событий, подлежащих аудиту, определены в данной спецификации, и для каждого из этих событий определен минимальный набор информации, которая должна (SHOULD) быть включена в журнал аудита.

- Для сеанса отсутствует действующий контекст безопасности. Элемент журнала аудита для такого события должен (SHOULD) включать значение SPI, дату и время получения пакета, Source Address, Destination Address, Sequence Number и открытый Flow ID (в IPv6).

- Пакет, предлагаемый ESP для обработки, оказывается IP-фрагментом, т.е. поле OFFSET не равно нулю или установлен флаг MORE FRAGMENTS. Элемент журнала аудита для такого события должен (SHOULD) включать значение SPI, дату и время получения пакета, Source Address, Destination Address, Sequence Number и открытый Flow ID (в IPv6).

- Попытка передачи пакета, который приведет к переполнению порядковых номеров. Элемент журнала аудита для такого события должен (SHOULD) включать значение SPI, текущие дату и время, Source Address, Destination Address, Sequence Number и открытый Flow ID (в IPv6).

- Полученный пакет не проходит контроль противодействия повторному воспроизведению. Элемент журнала аудита для такого события должен (SHOULD) включать значение SPI, дату и время получения пакета, Source Address, Destination Address, Sequence Number и Flow ID (в IPv6).

- Не проходит контроль целостности. Элемент журнала аудита для такого события должен (SHOULD) включать значение SPI, дату и время получения пакета, Source Address, Destination Address, Sequence Number и Flow ID (в IPv6).

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. Отсутствует требование к получателю передавать какое-либо сообщение подразумеваемому отправителю в ответ на обнаружение события, подлежащего аудиту, 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 ESP, 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 (несмотря на локальные перезагрузки и т.д.) до тех пор, пока ключ не будет заменен, и вероятно не будет осуществляться автоматическое восстановление, если приближается переполнение счетчика. 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 алгоритмы, обязательные для использования в ESP, описываются в отельном документе RFC [Eas04]. Дополнительные алгоритмы, помимо обязательных для ESP, могут (MAY) поддерживаться.

Поскольку использование шифрования в ESP является необязательным, то для обеспечения соответствия со способом согласования сервисов ESP, требуется также поддержка нулевого ("NULL") алгоритма шифрования. Поддержка версии ESP, предоставляющей только сервис конфиденциальности, является факультативной возможностью. Если реализация предлагает такой сервис, она должна (MUST) также поддерживать согласование нулевого ("NULL") алгоритма целостности. Заметим, что хотя при некоторых обстоятельствах, отмеченных выше, каждый из алгоритмов целостности и шифрования может быть нулевым ("NULL"), оба они не должны (MUST NOT) быть одновременно нулевыми.

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 документа RFC 2406

Данный документ отличается от документа RFC 2406 во многих отношениях:

- Степень обязательности предоставления сервиса конфиденциальности без обеспечения целостности (confidentiality-only service) – теперь MAY вместо MUST.

- SPI – модифицирован для определения унифицированного алгоритма поиска в SAD индивидуальных (unicast) и групповых (multicast) контекстов безопасности, покрывающего более широкий диапазон технологий групповой 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.

- Payload data – расширенная модель для приспособления к алгоритмам комбинированного режима.

- Заполнение для улучшения конфиденциальности потоков трафика – добавлено требование способности присоединять байты после конца блока данных IP, до начала поля Padding.

- Next Header – добавлено требование способности формировать и отбрасывать фиктивные пакеты с заполнением (значение поля Next Header = 59).

- ICV – расширенная модель для приспособления к алгоритмам комбинированного режима.

- Алгоритмы – добавлены алгоритмы режима комбинированной конфиденциальности.

- Указание на обязательные алгоритмы перенесено в отдельный документ.

- Обработка входящих и исходящих пакетов – теперь имеются два направления: (1) раздельные алгоритмы конфиденциальности и целостности и (2) алгоритмы режима комбинированной конфиденциальности. Из-за добавления алгоритмов комбинированного режима разделы шифрования/дешифрирования и целостности объединены как для обработки входящих пакетов, так и для обработки исходящих пакетов.

8. Соображения по обратной совместимости

В ESP отсутствует номер версии и отсутствуют механизмы, позволяющие партнерам IPsec определить или согласовать, какую версию ESP каждый из них использует или должен использовать. В данном разделе обсуждаются проблемы обратной совместимости, являющиеся следствием этого.

Во-первых, если не применяется ни одно из новых свойств, доступных в ESP v3, то формат пакета ESP в версиях v2 и v3 идентичен. Если применяется алгоритм шифрования комбинированного режима (свойство, которое поддерживается только в ESP v3), то результирующий формат пакета может отличаться от спецификации ESP v2. Однако партнер, реализующий только ESP v2, никогда не будет согласовывать такой алгоритм, поскольку они определены для использования только в контексте ESP v3.

Согласование расширенных порядковых номеров (ESN) поддерживается IKE v2 и адресовалось IKE v1 в виде дополнения ESN к домену интерпретации (DOI) IKE v1.

В новом ESP (v3) мы предусматриваем две меры для лучшей поддержки конфиденциальности потоков трафика (TFC):

- произвольное заполнение после конца IP-пакета
- соглашение по отбрасыванию пакетов, использующих Next Header = 59

Первое свойство не должно создавать проблем в получателе, поскольку поле общей длины IP указывает конец IP-пакета. Таким образом, любые байты заполнения TFC, следующие за концом пакета, должны удаляться в некотором месте в процессе обработки IP-пакета, после обработки ESP, даже если программные средства IPsec не удаляют такое заполнение. Таким образом, это свойство ESP v3 отправитель может использовать независимо от того, реализует ли получатель ESP v2 или ESP v3.

Второе свойство позволяет отправителю посылать блок данных, представляющий собой произвольную строку байтов, которые не обязательно составляют правильный IP-пакет внутри туннеля для целей TFC. Открытым остается вопрос, что будет делать получатель ESP v2, когда поле Next Header в пакете ESP будет содержать значение "59". Он может отбросить пакет, когда обнаружит неправильный IP-заголовок, и записать в журнал это событие, но это безусловно не может привести к отказу получателя, поскольку такое поведение создало бы уязвимость к атаке на доступность по отношению к трафику, полученному от аутентифицированных партнеров. Таким образом, это свойство представляет собой оптимизацию, которую отправитель ESP v3 может использовать независимо от того, реализует ли получатель ESP v2 или ESP v3.

9. 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.

10. Ссылки

10.1. 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.

[Pos81] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

10.2. Информативные ссылки

[Bel96] Steven M. Bellovin, "Problem Areas for the IP Security Protocols", Proceedings of the Sixth Usenix Unix Security Symposium, July, 1996.

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

[Kau05] Kaufman, C., Ed., "The Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[Ken-AH] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

[Kra01] Krawczyk, H., "The Order of Encryption and Authentication for Protecting Communications (Or: How Secure Is SSL?)", CRYPTO' 2001.

[NIST01] Federal Information Processing Standards Publication 140-2 (FIPS PUB 140-2), "Security Requirements for Cryptographic Modules", Information Technology Laboratory, National Institute of Standards and Technology, May 25, 2001.

[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.

[Syverson] P. Syverson, D. Goldschlag, and M. Reed, "Anonymous Connections and Onion Routing", Proceedings of the Symposium on Security and Privacy, Oakland, CA, May 1997, pages 44-54.

Приложение A. Расширенные (64-битовые) порядковые номера

A1. 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.

A2. 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).

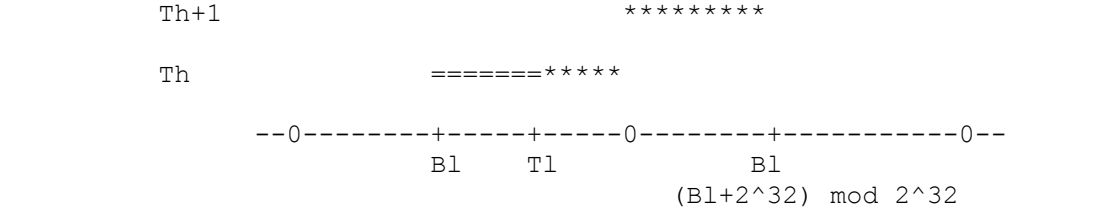


Рис. 1. Случай A.

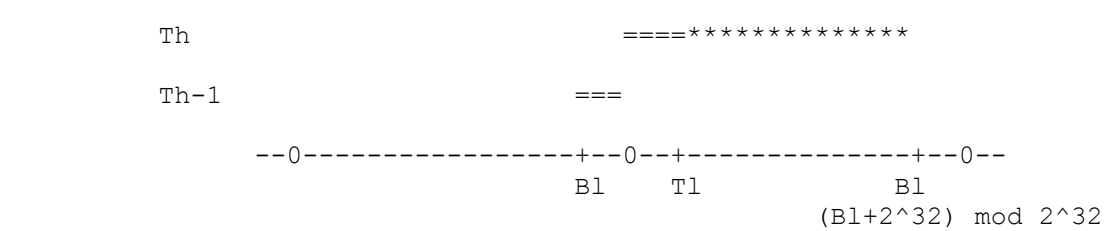


Рис. 2. Случай B.

A2.1. 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).

A2.2. 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

A2.3. 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
```

A3. 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.

A3.1. 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.

A3.2. 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 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

Полное определение авторских прав

Определение интеллектуальной собственности

Acknowledgement

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