Network Working Group
Request for Comments: 4303
Obsoletes: 2406
Category: Standards Track

S. Kent
BBN Technologies
December 2005

Инкапсуляция защищенных данных IP (ESP)

IP Encapsulating Security Payload (ESP)

Статус документа

Это документ содержит проект стандарта протокола Internet для сообщества Internet и служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации и статус протокола можно узнать из текущей версии документа Internet Official Protocol Standards (STD 1). Допускается свободное распространение документа.

Аннотация

Этот документ описывает обновлённую версию протокола ESP[1], разработанного для обеспечения различных услуг защиты в среде IPv4 и IPv6. Протокол ESP используется для обеспечения конфиденциальности, аутентификации источника данных, контроля целостности без организации специальных соединений, предотвращения повторного использования пакетов (форма контроля порядковых номеров) и ограниченной конфиденциальности потоков трафика. Данный документ отменяет действие RFC 2406 (ноябрь 1998).

Оглавление

[1]Encapsulating Security Payload - инкапсуляция защищённых данных.

1. Введение

В документе предполагается, что читатель достаточно знаком с терминами и концепциями, изложенными в документе «Архитектура защиты для протокола IP» [Ken-Arch], далее называемом для краткости описанием архитектуры. В частности, читателю следует понимать определения услуг по защите, обеспечиваемых ESP[1] [Ken-ESP] и AH, концепцию защищённых связей, способы использования ESP вместе с аутентификационным заголовком AH, а также различные опции управления ключами, поддерживаемые для ESP и AH.

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с RFC 2119 [Bra97].

Заголовок ESP разработан для обеспечения смешанных услуг по защите информации в среде IPv4 и IPv6 [DH98]. ESP может использоваться автономно, в комбинации с AH [Ken-AH] или в режиме вложенности (см. документ по архитектуре защиты [Ken-Arch]). Услуги по защите могут обеспечиваться между парой взаимодействующих хостов, парой защитных шлюзов, а также между защитным шлюзом и хостом. Более детальная информация об использовании ESP и AH в различных сетевых средах приведена в документе по архитектуре защиты [Ken-Arch].

Заголовок ESP помещается после заголовка IP и перед заголовком протокола следующего уровня (транспортный режим) или перед инкапсулированным заголовком IP (туннельный режим). Детальное описание обоих режимов приведено ниже.

Протокол ESP может использоваться для обеспечения конфиденциальности, аутентификации источника данных, контроля целостности без организации специальных соединений, предотвращения повторного использования пакетов (форма контроля порядковых номеров) и (ограниченной) конфиденциальности потоков трафика. Набор предоставляемых услуг зависит от опций, выбранных в момент организации защищённой связи (SA[2]), и местоположения реализации протокола в сетевой топологии.

В ESP допускается использование только функций шифрования для обеспечения конфиденциальности. Однако следует отметить, что в общем случае шифрование будет обеспечивать лишь защиту от пассивных атак. Использование шифрования без строгого контроля целостности (в ESP или с помощью AH) может сделать шифрованные услуги уязвимыми для некоторых форм активных атак [Bel96, Kra01]. Более того, нижележащие службы контроля целостности (такие, как AH), использованные до шифрования, не обеспечивают достаточной защиты конфиденциальных данных от активных атак при использовании только шифрования [Kra01]. ESP позволяет использовать SA только с шифрованием, поскольку в этом случае обеспечивается более высокая производительность в сочетании с адекватной защитой (например, при независимой реализации услуг проверки аутентификации и целостности данных). Однако стандарт не требует от реализаций ESP предлагать лишь услуги шифрования.

Идентификация источника данных и контроль целостности без организации специальных соединений являются связанными услугами и совместно называются услугами по обеспечению целостности (integrity[3]). ESP с обеспечением только услуг целостности **должны** предлагаться как опция выбора услуг (например, это должно согласовываться в протоколах управления SA и **должно** быть настраиваемым с использованием интерфейса управления). ESP с поддержкой лишь целостности являются привлекательной альтернативой AH в различных контекстах (например, по причине более высокой производительности или большей пригодности для канализации во многих приложениях).

Хотя конфиденциальность и целостность могут обеспечиваться независимо, ESP обычно поддерживает оба типа услуг (т. е., пакеты будут защищаться как в части конфиденциальности, так и в части целостности). Таким образом, существует три варианта услуг ESP:

- только конфиденциальность **(может** поддерживаться);
- только целостность **(должна** поддерживаться);
- конфиденциальность и целостность (**должна** поддерживаться)

Поддержка предотвращения повторного использования пакетов может быть выбрана для SA только вместе с поддержкой функций целостности для этой SA. Выбор этой услуги полностью отдаётся на откуп получателю и не требует согласования. Однако для использования расширенных порядковых номеров (ESN) требуется согласование этой опции - ESP требует от протоколов управления SA поддержки возможности такого согласования (см. параграф 2.2.1).

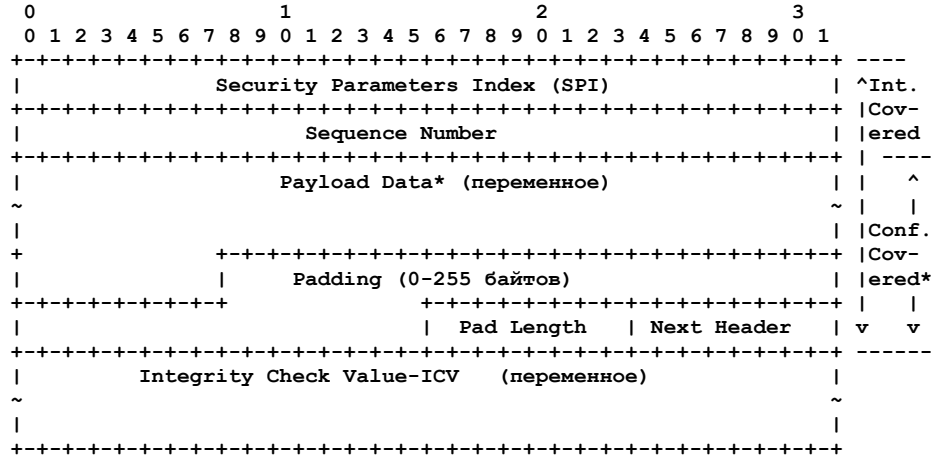
Услуги по обеспечению конфиденциальности потоков трафика (TFC[4]) в общем случае эффективны только при таком развёртывании ESP, когда обеспечивается сокрытие адресов исходных отправителей и получателей (например, в туннеле между защитными шлюзами) и только при потоке трафика между партнёрами IPsec (естественного или генерируемого в целях маскировки), достаточном для сокрытия конкретного потока индивидуальных абонентов[5]. Новые функции TFC, включённые в ESP, облегчают генерацию и отбрасывание маскирующего трафика и обеспечивают более

[1] Encapsulating Security Payload - инкапсулированные защищённые данные.
[2] Security Association.
[3] Этот термин используется потому, что (на базе пакетов) выполняемые расчёты будут обеспечивать контроль целостности напрямую без организации специальных соединений. Идентификация источника данных выполняется опосредованно в результате привязывания ключа, используемого для контроля целостности к партнёру IPsec. Обычно такая привязка обеспечивается за счет использования разделяемого симметричного ключа.
[4] Traffic flow confidentiality.

эффективное заполнение для реального трафика. При этом обеспечивается совместимость с более ранними версиями.

В разделе 7 кратко перечислены отличия данной спецификации от RFC 2406.

2. Формат пакетов ESP

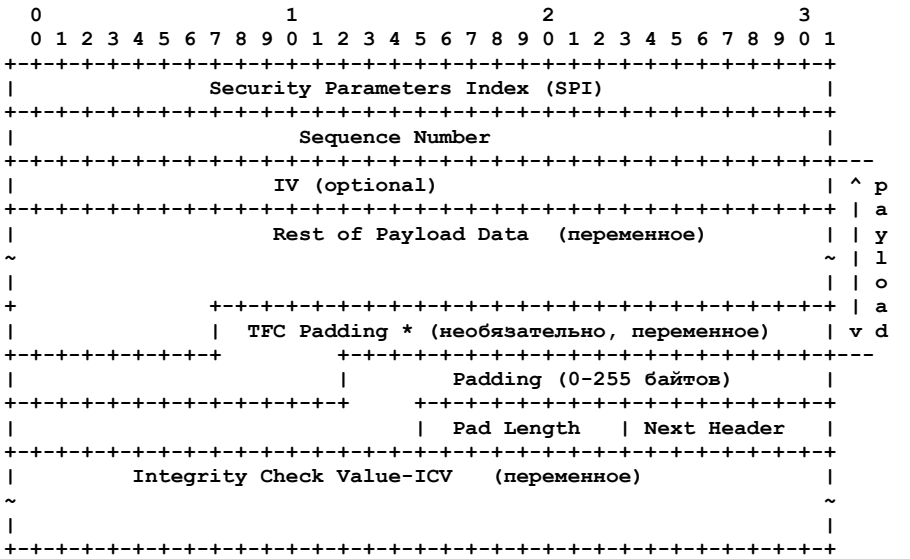


* При наличии в поле Payload данных криптографической синхронизации (например, вектора инициализации - IV, описанного в параграфе 2.3) эти данные обычно не шифруются, хотя о них зачастую говорят, как о части шифрованных данных.

Рисунок 1. Формат пакета ESP на верхнем уровне.

В заголовок (внешний) протокола (IPv4, IPv6, Extension), непосредственно предшествующий заголовку ESP, **следует** включать значение 50[1] в поле Protocol (IPv4) или Next Header (IPv6, Extension). Рисунок 1 показывает верхний уровень формата пакетов ESP. Пакет начинается с двух 4-байтовых полей (SPI [2] и Sequence Number[3]). Вслед за ними размещается поле данных Payload Data, структура которого зависит от выбора алгоритма шифрования и режима, а также заполнения TFC, детально описанного ниже. Вслед за полем Payload Data размещается поле запол-нения (Padding), поле размера заполнения (Pad Length) и поле Next Header (следующий заголовок). Дополнительно в конце пакета может помещаться значение ICV[4].

Трейлер (передаваемый) ESP содержит поля Padding, Pad Length и Next Header. В дополнение к этим полям включаются неявные данные трейлера ESP (не передаются), используемые для контроля целостности, как описано ниже.



* При использовании туннельного режима реализация IPsec может добавлять заполнение TFC (см. параграф 2.4) после поля Payload Data и перед полем Padding.

Рисунок 2. Субструктура данных (Payload Data).

При выборе контроля целостности контрольная сумма дополняет поля SPI, Sequence Number, Payload Data и трейлер ESP (явный и неявный).

При выборе услуг конфиденциальности шифруется поле Payload Data (за исключением данных криптографической синхронизации, которые могут быть включены, но не шифруются) и (явный) трейлер ESP.

---

[5]Система ESP может быть развёрнута, как часть системы TFC более высокого уровня (например, Onion Routing [Syverson]), но такие системы выходят за пределы настоящего стандарта.

[1]Значения идентификаторов протоколов приведены на странице http://www.iana.org/assignments/protocol-numbers.

[2]Security Parameters Index - список параметров защиты.

[3]Sequence Number - порядковый номер.

[4]Integrity Check Value - контрольная сумма для проверки целостности.

Как отмечено выше, поле Payload Data может иметь дополнительную структуру. Алгоритмы шифрования, которым требуется явный вектор инициализации IV[5] (например, CBC[6]), часто используют эти данные в качестве префикса защищаемых данных (Payload Data). Некоторые алгоритмы объединяют шифрование и контроль целостности в одну операцию - здесь такие алгоритмы будут называться комбинированными. Приспособление таких алгоритмов требует от алгоритма явного описания структуры Payload Data, используемой для передачи данных контроля целостности.

Некоторые комбинированные алгоритмы обеспечивают целостность только для шифрованных данных, тогда как другие могут обеспечивать целостность неких дополнительных данных, которые не шифруются для передачи. Поскольку поля SPI и Sequence требуют контроля целостности и не шифруются, необходимо обеспечить их целостность, независимо от выбранных услуг и стиля работы комбинированного алгоритма.

При использовании комбинированного алгоритма предполагается, что этот алгоритм сам по себе будет возвращать шифрованные данные и результат проверки целостности. Для комбинированных алгоритмов значение ICVЭ обычно находящее в конце пакета ESP (когда выбран контроль целостности), можно опустить. Когда выбран контроль целостности и ICV опускается, ответственность за кодирование эквивалента ICV в поле Payload Data и проверку целостности пакета ложится на комбинированный алгоритм.

Если комбинированный алгоритм обеспечивает только целостность данных, которые уже зашифрованы, необходимо реплицировать значения полей SPI и Sequence Number, как часть Payload Data.

В заключение добавляются байты заполнения для сохранения конфиденциальности потоков трафика после поля Payload Data и перед трейлером ESP. Рисунок 2 показывает структуру поля Payload Data для таких случаев[3].

При использовании комбинированного алгоритма явное поле ICV, показанное на рисунках 1 и 2, может отсутствовать (см. параграф 3.3.2.2). Поскольку алгоритмы и режимы задаются при организации SA, формат пакетов ESP для данной SA (включая структуру Payload Data) фиксирован для всего трафика данной SA.

Приведённые ниже таблицы описывают поля предшествующих рисунков и иллюстрируют, как несколько категорий опций алгоритмов с различными моделями обработки воздействуют на упомянутые выше поля. Детали обработки рассматриваются ниже.

Таблица 1. Раздельные алгоритмы шифрования и контроля целостности.

	Число байтов	Требуется[4]	Шифруется	Учитывается в ICV	Передаётся	
SPI	4	M		+	Без шифрования	
Seq# (младшие биты)	4	M		+	Без шифрования	
IV	переменное	O		+	cipher[5]	Payload
IP datagram[6]	переменное	M или D	+	+	cipher[5]	Payload
TFC padding	переменное	O	+	+	cipher[5]	Payload
Padding	0 - 255	M	+	+	cipher[5]	
Pad Length	1	M	+	+	cipher[5]	
Next Header	1	M	+	+	cipher[5]	
Seq# (старшие биты)	4	При ESN[7]		+	Не передаётся	
ICV Padding	переменное	Если используется		+	Не передаётся	
ICV	переменное	M[8]			Без шифрования	

Таблица 2. Комбинированные алгоритмы шифрования и контроля целостности.

	Число байтов	Требуется[9]	Шифруется	Учитывается в ICV	Передаётся	
SPI	4	M		+	Без шифрования	
Seq# (младшие биты)	4	M		+	Без шифрования	
IV	переменное	O		+	cipher	Payload
IP datagram[10]	переменное	M или D	+	+	cipher	Payload
TFC padding[11]	переменное	O	+	+	cipher	Payload
Padding	0 - 255	M	+	+	cipher	
Pad Length	1	M	+	+	cipher	
Next Header	1	M	+	+	cipher	
Seq# (старшие биты)	4	При ESN[12]		+	[13]	
ICV Padding	переменное	Если используется		+	[13]	
ICV	переменное	O[14]			Без шифрования	

В последующих параграфах рассматриваются поля заголовка. Необязательные поля могут быть опущены (т. е., отсутствуют как при передаче, так и при расчете ICV - см. параграф 2.7), если соответствующая опция не выбрана. Обязательные поля присутствуют в пакетах ESP всегда для каждой SA. Формат пакетов ESP для данной SA фиксирован на протяжении всего срока существования данной SA.

Примечание. Все криптографические алгоритмы, используемые в IPsec, предполагают на входе канонический сетевой порядок байтов (см. Приложение к RFC 791 [Pos81]) и генерируют результаты с использованием этого же порядка.

---

[5]Initialization Vector.
[6]Cipher Block Chaining - сцепка шифрованных блоков.
[3]На рисунке показаны «биты в среде передачи» - даже при использовании расширенных порядковых номеров передаваемые пакеты включают только младшие 32 бита порядкового номера (см. параграф 2.2.1).
[4]M = обязательно; O = опция; D = фикция.
[5]Ciphertext, если выбрано шифрование.
[6]В туннельном режиме дейтаграмма IP, в транспортном - следующий заголовок и данные.
[7]См. параграф 2.1.1
[8]Обязательно при использовании отдельного механизма контроля целостности.
[9]M = обязательно; O = опция; D = фикция
[10]В туннельном режиме дейтаграмма IP, в транспортном - следующий заголовок и данные.
[11]Может использоваться только при указании реального размера данных (payload).
[12]См. параграф 2.1.1
[13]Передача этого поля определяется алгоритмом, но в любом случае поле является «невидимым» для ESP.
[14]Присутствие этого поля определяется спецификацией алгоритма.

ESP не включает номера версии, следовательно при возникновении вопросов о совместимости с предыдущими версиями, эти вопросы **должны** решаться с использованием механизмов сигнализации (например, IKEv2) [Kau05] или настройка конфигурации по отдельному каналу) между партнёрами IPsec, обеспечивающих совместимость версий ESP.

2.1. Security Parameters Index (SPI) - список параметров защиты

SPI представляет собой произвольное 32-битовое значение, используемое получателем для идентификации SA, с которой связан входящий пакет. Поле SPI является обязательным.

Для индивидуальных SA, значение SPI может само по себе идентифицировать SA или использоваться в комбинации с типом протокола IPsec (в данном случае ESP). Поскольку для индивидуальных SA значение SPI генерируется получателем, решение вопроса о достаточности этого значения для идентификации SA или необходимости использования в комбинации с типом протокола IPsec определяется локальными условиями. Этот механизм отображения входящего трафика на индивидуальные SA **должен** поддерживаться всеми реализациями ESP.

В реализациях IPsec, поддерживающих групповую адресацию, **должны** поддерживаться групповые SA с использованием описанного ниже алгоритма отображения входящих дейтаграмм IPsec на SA. Разработчикам, поддерживающим только индивидуальный трафик, не обязательно реализовать механизм демультиплексирования.

Во многих защищённых multicast-архитектурах (например, [RFC3740]) центральный контроллер группы/сервер ключей сам выделяет для группы значение SPI. Выделение SPI не согласуется и не координируется с подсистемами управления ключами (например, IKE) на конечных узлах группы. Следовательно, возникает возможность совпадения значений SPI для групповой и индивидуальной SA. Поддерживающие групповую адресацию реализации IPsec **должны** корректно демультиплексировать входящий трафик даже в случаях совпадения значений SPI.

Каждая запись в базе данных защищённых связей (SAD[1]) [Ken-Arch] должна указывать, по каким критериям в дополнение к SPI отыскивается SA – получатель, получатель и отправитель. Для групповых SA поле протокола не используется при поиске SA. Для каждого входящего пакета с защитой IPsec реализация должна произвести поиск в SAD и найти запись, наиболее точно соответствующую идентификатору SA. Если обнаруживается более одной записи SAD, соответствующей значению SPI, выбирается запись по наиболее точному соответствию получателя или получателя и отправителя (как указано в записи SAD). Таким образом, логический порядок поиска в SAD имеет вид:

1. Поиск в базе SAD соответствия {SPI, адрес получателя, адрес отправителя}. Если запись SAD найдена, входящий пакет AH обрабатывается с найденной записью SAD. В противном случае выполняется п. 2.
2. Поиск в базе SAD соответствия {SPI, адрес получателя}. Если запись SAD найдена, входящий пакет AH обрабатывается с найденной записью SAD. В противном случае выполняется п. 3.
3. Поиск в базе SAD соответствия {SPI}, если получатель выбрал поддержку одного пространства SPI для AH и ESP, или {SPI, протокол} в противном случае. Если запись SAD найдена, входящий пакет AH обрабатывается с найденной записью SAD. В противном случае пакет отбрасывается с записью в журнал аудита.

На практике реализация **может** выбрать любой метод ускорения поиска, но наблюдаемое извне поведение **должно** соответствовать описанному выше поиску в SAD. Например, программные реализации могут индексировать хэш-таблицу SPI. Записи SAD в хэш-таблице сортируются в связный список, в котором записи для SA с большим соответствием располагаются ближе к началу, а с меньшим соответствием - ближе к концу списка. В аппаратных реализациях поиск максимального соответствия может ускоряться встроенными средствами с использованием общедоступной технологии TCAM[2].

Индикация использования адресов отправителя и получателя при поиске соответствия для отображения входящего трафика IPsec на SA **должна** выполняться при настройке конфигурации SA вручную или путем согласования параметров с использованием протокола управления SA (например, IKE или GDOI[3] [RFC3547]). Обычно группы SSM[4] [HC03] используют трехкомпонентный идентификатор SA, включающий SPI, групповой адрес получателей и адрес отправителя. SA группы Any-Source Multicast требует в качестве идентификатора только SPI и групповой адрес получателей.

Значения SPI в диапазоне от 1 до 255 зарезервированы IANA для использования в будущем. Эти значения не будут распределяться агентством IANA, пока их использование не будет оговорено в специальном RFC. Значение SPI = 0 зарезервировано для локального, связанного с реализацией, применения и его **недопустимо** передавать в сеть. Например, реализация управления ключами может использовать SPI=0 для идентификации отсутствия защищенных связей[5] в период, когда реализация IPsec запрашивает новую SA для объекта управления ключами, но данная SA еще не организована.

2.2. Sequence Number - порядковый номер

Это 32-битовое поле, трактуемое, как целое число без знака, содержит значение счётчика пакетов, которое увеличивается на 1 для каждого переданного пакета (счётчик пакетов для SA). Для индивидуальных SA и групповых SA с одним отправителем, последний **должен инкрементировать** данное поле для каждого переданного пакета. Использование одной SA множеством отправителей допустимо, хотя в общем случае не рекомендуется. ESP не предоставляет возможностей синхронизации порядковых номеров между множеством отправителей или осмысленного счётчика пакетов на стороне получателя и не обеспечивает окна в контексте множества отправителей. Таким образом, для SA с множеством отправителей функции предотвращения повторного использования пакетов ESP становятся недоступными (см. параграфы 3.3.2 и 3.4.3).

Это поле является обязательным и **должно** присутствовать даже в тех случаях, когда получатель не пользуется услугами по предотвращению повторного использования пакетов для конкретной SA. Обработка поля Sequence

[1]Security Association Database.
[2]Ternary Content-Addressable Memory - ассоциативная память.
[3]Group Domain of Interpretation.
[4]Source-Specific Multicast.
[5]No Security Association Exists.

Number осуществляется по усмотрению получателя, но все реализации ESP **должны** обеспечивать возможность обработки, описанной в параграфах 3.3.3. Генерация порядковых номеров и 3.4.3. Проверка порядковых номеров. Таким образом, отправитель **должен** передавать это поле, но получатель не обязан принимать его во внимание (см. обсуждение проверки порядковых номеров в параграфе 3.4.3. Проверка порядковых номеров.

Счётчики на стороне отправителя и получателя инициализируются нулевым значением при создании SA (первый пакет, переданный с использованием данной SA будет иметь порядковый номер 1; генерация порядковых номеров более подробно описана в параграфе 3.3.3). Если предотвращение повторного использования пакетов включено (используется по умолчанию), передаваемые порядковые номера никогда не должны повторяться. Таким образом, счётчики пакетов на стороне отправителя и получателя **должны** сбрасываться (путём создания новой SA и нового ключа) до передачи пакета с порядковым номером 2^32 в каждой SA.

2.2.1. Extended Sequence Number - расширенный порядковый номер (64 бита)

В высокоскоростных реализациях IPsec **следует** предлагать новую опцию для расширения 32-битового поля порядкового номера. Использование поля ESN[1] **должно** согласовываться протоколом управления SA. Отметим, что в IKEv2 это согласование происходит неявно - использование ESN включено по умолчанию, пока явно не выбраны 32-битовые порядковые номера. Поддержка ESN возможна как для индивидуальных, так и для групповых SA.

ESN позволяет использовать для SA 64-битовые порядковые номера (см. Приложение A: Расширенные порядковые номера (64 бита)). В заголовке ESP каждого пакета для минимизации издержек передаются только младшие 32 бита расширенного порядкового номера, а старшие 32 бита учитываются, как часть порядкового номера, отправителем и получателем и включаются в расчет ICV (если контроль целостности используется). Если реализован отдельный алгоритм контроля целостности, старшие биты включаются в неявный трейлер ESP, но не передаются по аналогии с битами заполнения алгоритма контроля чётности. При использовании комбинированного алгоритма, последний определяет судьбу старших битов ESN - передавать или неявно включать в расчёт. Детали обработки рассматриваются в параграфе 3.3.2.2.

2.3. Payload Data - данные

Поле переменного размера Payload Data содержит данные (из исходного пакета IP), указываемые полем Next Header. Поле Payload Data является обязательным и размер его составляет целое число байтов. Если алгоритм, используемый для шифрования данных, требует данных криптографической синхронизации (например, вектор инициализации - IV), эти данные явно передаются в поле Payload, но не рассматриваются в качестве отдельного поля в ESP (т. е., передача явного IV невидима для ESP - см. Рисунок 2). Любой алгоритм шифрования, требующий таких явных данных синхронизации для каждого пакета, **должен** указывать размер и структуру таких данных, а также их местоположение в RFC, содержащем спецификацию использования алгоритма с ESP. Обычно IV помещается непосредственно перед зашифрованным текстом (см. Рисунок 2). Если данные синхронизации передаются неявно, алгоритм их выделения **должен** быть описан в RFC с определением алгоритма шифрования. Если неявные данные криптографической синхронизации (например, вектор инициализации - IV) включаются в поле Payload, обычно эти данные не шифруются (см. таблицы 1 и 2), хотя в некоторых случаях о них говорят, как о части шифрованных данных.

Отметим, что начало заголовка протокола следующего уровня **должно** быть выровнено относительно заголовка ESP - для IPv4 выравнивание выполняется по 4-байтовой границе, для IPv6 - по 8-байтовой.

В части выравнивания (действительно) зашифрованных данных при наличии IV отметим следующее:

- Для некоторых режимов работы на базе IV получатель трактует IV, как начало шифрованных данных, передавая вектор инициализации в алгоритм напрямую. В таких случаях идентификация начала (действительно) шифрованных данных не вызывает проблем на приёмной стороне.
- В некоторых случаях получатель считывает IV независимо от шифрованных данных. В таких случаях алгоритм **должен** указывать способ идентификации начала (действительно) шифрованных данных.

2.4. Padding - заполнение (для шифрования)

Использование поля Padding обусловлено двумя основными факторами:

- Если алгоритм шифрования требует, чтобы размер шифруемых данных был кратен некоторому целому числу байтов (например, размер блока при блочном шифровании), поле Padding используется для требуемого алгоритмом выравнивания нешифрованных данных (поля Payload Data, Padding, Pad Length и Next Header) до требуемого алгоритмом размера.
- Заполнение может потребоваться и без связи с алгоритмом шифрования для обеспечения выравнивания зашифрованных данных по 4-байтовой границе. В частности, поля Pad Length и Next Header должны быть выравниваться по 4-байтовой границе, как показано выше на рисунке, описывающем формат пакетов ESP, для обеспечения выравнивания поля ICV (если оно используется) по 4-байтовой границе.

Независимо от приведённых выше требований заполнение может служить для сокрытия действительного размера зашифрованных данных с целью поддержки конфиденциальности потока трафика (TFC). Однако описываемое здесь поле Padding слишком мало для эффективной реализации TFC, поэтому его не следует использовать с такой целью. Для обеспечения конфиденциальности потока следует использовать специальный механизм, описанный ниже (см. параграф 2.7).

Отправитель **может** добавлять в поле заполнения от 0 до 255 байтов. Включение поля Padding в пакет ESP является необязательным (в соответствии с приведёнными выше требованиями), но каждая реализация **должна** поддерживать генерацию и восприятие поля заполнения.

- При использовании заполнения для выравнивания размера шифрованных данных в соответствии с требованиями алгоритма к размеру блока (первое требование выше) при расчёте заполнения принимается во внимание поле Payload Data без IV, но со включением всех трейлерных полей ESP. Если комбинированный

[1] Extended Sequence Number - расширенный порядковый номер.

алгоритм требует передачи SPI и Sequence Number для контроля целостности (например, репликации SPI и Sequence Number в поле Payload Data), тогда реплицированные версии этих полей, а также все связанные данные эквивалента ICV включаются в расчёт размера заполнения. Если выбрана опция ESN, старшие 32 бита ESN также учитываются при расчёте заполнения, если комбинированный алгоритм требует их передачи для контроля целостности.

- Для выравнивания поля ICV по 4-байтовой границе (второе требование выше) при расчете заполнения учитывается поле Payload Data, включая IV, Pad Length и Next Header. При использовании комбинированного алгоритма все реплицируемые данные и эквивалент ICV учитываются в Payload Data для расчета заполнения.

Если поле Padding требуется, но алгоритм шифрования не задаёт содержимого заполнения, **должна** использоваться описанная ниже обработка, принятая по умолчанию. Байты Padding инициализируются последовательностями целочисленных значений (1 байт, без знака). Первый байт заполнения, добавляемый в конце шифрованных данных, имеет номер 1, а номера последующих байтов монотонно возрастают на единицу, образуя последовательность 1, 2, 3, ... При использовании такой схемы заполнения получателю **следует** проверять поле Padding (эта схема была выбрана за ее относительную простоту, возможность аппаратной реализации и наличие некоторой защиты от ряда форм атак «cut and paste» в отсутствии механизмов контроля целостности, если получатель проверяет заполнение до расшифровки).

Если алгоритм шифрования или комбинированный алгоритм вносят ограничения в выбор значений байтов заполнения, эти ограничения **должны** быть указаны в RFC, определяющем использование алгоритма с ESP. Если алгоритм требует проверки значений байтов заполнения, это требование также **должно** быть включено в упомянутый документ RFC.

2.5. Pad Length - размер заполнения

Поле Pad Length показывает число байтов заполнения, непосредственно предшествующих данному полю в поле Padding. Значение поля лежит в диапазоне от 0 до 255, где нулевое значение говорит об отсутствии байтов заполнения. Как было отмечено выше, в этом поле не учитываются байты заполнения TFC. Поле Pad Length является обязательным.

2.6. Next Header - следующий заголовок

Восьмибитовое обязательное поле Next Header показывает тип данных, содержащихся в поле Payload Data (например, пакет IPv4 или IPv6, заголовок и данные следующего уровня). Значения этого поля выбираются из списка номеров протоколов IP[1], представленного на сайте IANA[2]. Например, значение 4 показывает протокол IPv4, значение 41 - IPv6, а 6 - протокол TCP.

Для упрощения быстрой генерации и отбрасывания трафика заполнения, используемого для обеспечения конфиденциальности потока (см. параграф 2.4), в «бутафорских» пакетах **должен** указываться идентификатор протокола 59 (нет следующего заголовка). Передающая сторона **должна** обеспечивать возможность генерации «бутафорских» пакетов с указанным значением поля, а получатель **должен** быть готов к отбрасыванию таких пакетов, без индикации ошибки. Все остальные заголовки и трейлерные поля ESP (SPI, Sequence Number, Padding, Pad Length, Next Header, ICV) **должны** присутствовать в «бутафорских» пакетах, но нешифрованную часть данных (кроме поля Next Header), следует делать бессмысленной (например, включать в эту часть поля Payload Data случайные байты). «Бутафорские» пакеты отбрасываются без какого-либо ущерба.

Реализациям **следует** обеспечивать локальные средства управления использованием «бутафорских» пакетов на уровне SA. Средствам управления следует давать пользователю возможность включения и выключения этой функции, а также управления параметрами (например, средства управления могут позволять администратору выбирать случайный или фиксированный размер «бутафорских» пакетов).

Обсуждение. «бутафорские» пакеты могут помещаться в поток со случайными интервалами для маскировки отсутствия реального трафика. Они могут также «формовать» реальный трафик в соответствии с некоторым распределением. Наибольший уровень защиты потоков трафика (TFS[3]) будет обеспечивать генерация «бутафорских» пакетов со скоростью, обеспечивающей постоянную скорость передачи данных для SA. Если все пакеты имеют одинаковый размер, SA показывает постоянную скорость потока данных, аналогично средствам шифрования каналов на уровне 1 или 2. Однако, неочевидно, что такой подход будет подходить во всех случаях (например, при наличии множества активных SA такое решение будет приводить к неоправданному расходу полосы, сводящему на нет преимущества использования коммутации пакетов вместо коммутации каналов). Разработчикам **следует** обеспечивать средства управления, позволяющие локальному администратору управлять генерацией «бутафорских» пакетов для целей TFC.

2.7.Заполнение TFC

Как было отмечено выше, размер поля Padding ограничен 255 байтами. В общем случае этого не достаточно для адекватного сокрытия характеристик трафика с целью обеспечения конфиденциальности. Для решения задач обеспечения конфиденциальности потоков (TFC[4]) используется специальное необязательное поле.

Реализациям IPsec **следует** обеспечивать возможность заполнения трафика путём добавления байтов после поля Payload Data, но до начала поля Padding. Однако такое заполнение (его часто называют заполнением TFC) может добавляться только в тех случаях, когда поле Payload Data содержит размер дейтаграммы IP. Это требование всегда выполняется в туннельном режиме и может выполняться в транспортном, если протокол следующего уровня (например, IP, UDP, ICMP) содержит точное значение размера. Информация о размере позволяет получателю пакета отбросить заполнение TFC, поскольку известен истинный размер поля Payload Data (поля трейлера ESP находятся путём отсчёта от конца пакета ESP). Соответственно, при добавлении заполнения TFC значение поля, содержащего размер дейтаграммы IP **недопустимо** менять с учётом этого заполнения. Данный стандарт не включает требований к содержимому заполнения.

---

[1] Реестр «IP Protocol Numbers».
[2] В настоящее время список доступен по ссылке http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml. *Прим. перев.*
[3] Traffic Flow Security - защита потока трафика.
[4] Traffic Flow Confidentiality - конфиденциальность потока трафика.

В принципе, существующие реализации IPsec могли использовать такую возможность и ранее с сохранением прозрачности. Однако по причине того, что получатели не были готовы к работе с таким заполнением, протокол управления SA **должен** был согласовывать такую возможность до того, как отправитель начнёт её применять, поскольку нужно было обеспечить совместимость с более ранними реализациями. В комбинации с описанным в параграфе 2.6 соглашением об использовании ID 59 реализация ESP может генерировать «пустышки» (dummy) и реальные пакеты с большее широкими пределами изменения размеров в целях поддержки TFC.

Реализациям **следует** поддерживать локальные средства управления для включения этой функции на уровне SA. Эти средства должны позволять пользователю управлять применением данной функции, а также обеспечивать параметрический контроль параметров.

2.8. ICV - контроль целостности

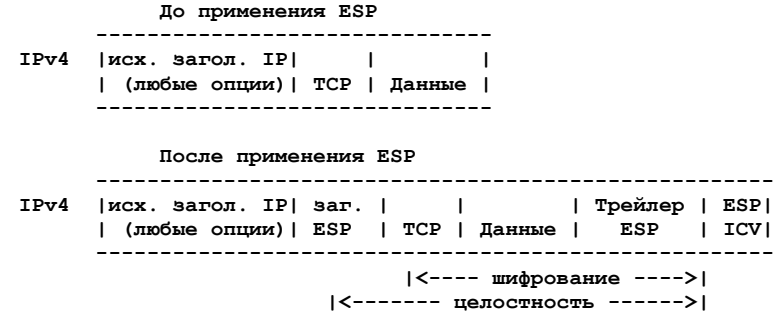
Поле переменного размера ICV[1] вычисляется для заголовка ESP, поля Payload и трейлерных полей ESP. Неявные поля трейлера ESP (заполнение и старшие биты ESN) принимаются во внимание при расчёте ICV. Поле ICV является необязательным. Оно присутствует лишь в тех случаях, когда контроль целостности включён и обеспечивается с использованием отдельного алгоритма или комбинированного алгоритма с поддержкой ICV. Размер поля определяется выбранным алгоритмом контроля целостности и связывается с SA. Спецификация алгоритма контроля целостности **должна** задавать размер ICV, а также правила сравнения и порядок проверки корректности значений.

3. Обработка ESP

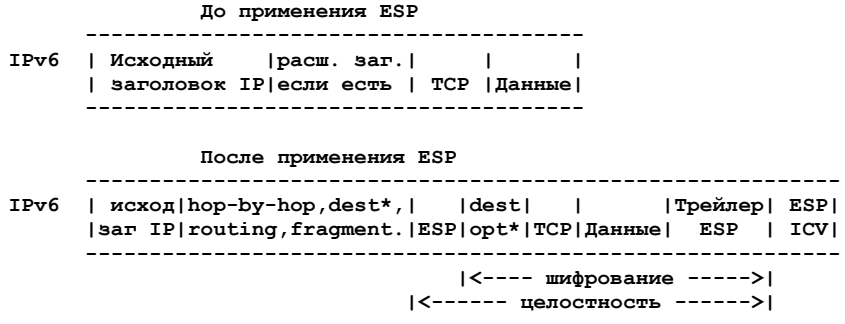
3.1. Расположение заголовка ESP

ESP может работать в двух режимах - транспортном и туннельном.

3.1.1. Транспортный режим



В транспортном режиме[2] ESP помещается после заголовка IP и перед заголовком следующего уровня (например, TCP, UDP, ICMP и т. п.). В контексте IPv4 это означает размещение ESP после заголовка IP (и всех опций), но перед протоколом следующего уровня (если для пакета используется также AH, заголовок аутентификации применяется к заголовку ESP, полю Payload, трейлеру ESP и ICV). На рисунке показано расположение ESP в типичном заголовке IPv4 (на этом и следующих рисунках данного параграфа показано поле ICV, реальное наличие которого связано с функциями защиты и выбранными алгоритмом и режимом).



* - при наличии может размещаться до или после ESP или в обоих местах

В контексте IPv6 заголовок ESP представляется как передаваемые «насквозь» данные и ему, таким образом, следует размещаться после заголовков расширения hop-by-hop, routing и fragmentation. Заголовки расширения опций адресата могут размещаться до, после и по обе стороны от заголовка ESP, в зависимости от требуемой семантики. Однако, поскольку ESP защищает только поля, расположенные после заголовка ESP, в общем случае желательно помещать опции адресата после заголовка ESP. На рисунке справа показан заголовок ESP для типичного пакета IPv6 при использовании транспортного режима.

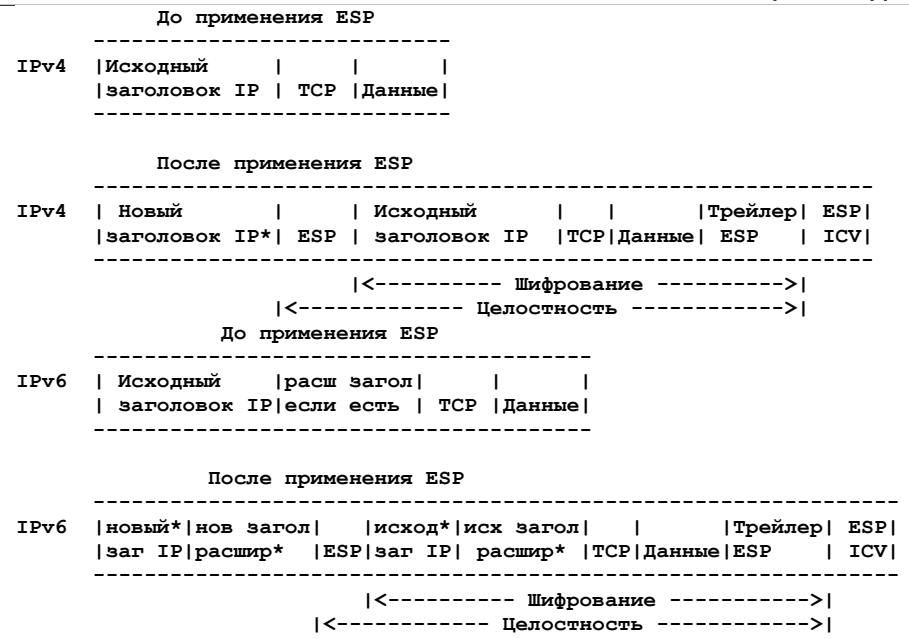
Отметим, что в транспортном режиме для реализаций bump-in-the-stack и bump-in-the-wire, как указано в описании архитектуры защиты, входящие и исходящие фрагменты IP могут требовать от реализации IPsec выполнения дополнительных операций сборки/фрагментации в соответствии с данной спецификацией и для обеспечения прозрачной поддержки IPsec. При выполнении таких операций требуются особые меры осторожности в тех случаях, когда используется множество интерфейсов.

3.1.2. Туннельный режим

В туннельном режиме «внутренний» заголовок IP указывает исходные (IP) адреса отправителя и получателя, а «внешний» заголовок IP содержит адреса «партнёров» IPsec (защитных шлюзов). Допускается различие версий

[1] Integrity Check Value - значение контроля целостности.

[2] Отметим, что транспортный режим не следует ограничивать использованием лишь для транспортных протоколов TCP и UDP.



*Конструкция внешнего заголовка/расширения и изменение внутреннего обсуждаются в документе по архитектуре защиты. Эти элементы не обязательны.

внутреннего и внешнего заголовков IP (т. е., возможна передача IPv6 по протоколу IPv4 и IPv4 по IPv6). В туннельном режиме ESP защищает вложенный пакет IP целиком, включая внутренний заголовок IP. Положение ESP в туннельном режиме относительно внешнего заголовка IP совпадает с положением ESP в транспортном режиме. На рисунке показано положение ESP для туннельного режима в заголовках типичных пакетов IPv4 и IPv6.

3.2. Алгоритмы

Обязательные для реализации алгоритмы, используемые с ESP, описаны в отдельном RFC для упрощения процедур обновления требований к алгоритмам, независимо от протокола. Кроме обязательного набора для ESP **могут** поддерживаться и другие алгоритмов. Отметим, что несмотря на опциональных характер поддержки конфиденциальности и целостности, по крайней мере одна из этих служб **должна** быть выбрана и, следовательно, **недопустимо** устанавливать значение NULL одновременно для обоих алгоритмов.

3.2.1. Алгоритмы шифрования

Алгоритм шифрования, используемый для защиты пакета ESP задаётся на уровне SA в которой пакет передаётся/принимается. Поскольку пакеты IP могут доставляться с нарушением порядка и потерей части пакетов, в каждом пакете должны содержаться все данные, требуемые получателю для выполнения криптографической синхронизации при расшифровке. Эти данные могут передаваться явно в поле payload (например, как описанный выше вектор инициализации IV) или выделяться из незашифрованной части[1] заголовка пакета (внешнего заголовка IP или заголовка ESP). Поскольку ESP использует заполнение незашифрованной части, используемый ESP алгоритм шифрования может демонстрировать характеристики блочного или потокового режима. Поскольку шифрование (конфиденциальность) **может** быть опциональной услугой (например, в ESP с обеспечением только контроля целостности), этот алгоритм **может** быть пустым (NULL) [Ken-Arch].

Чтобы позволить реализации ESP рассчитывать заполнение для шифрования, требуемое блочными алгоритмами, и определять влияние алгоритма на MTU, в RFC для каждого алгоритма, используемого с ESP, должен задаваться модуль заполнения.

3.2.2. Алгоритмы контроля целостности

Алгоритм контроля целостности, применяемый для расчёта ICV, задаётся в SA, используемой для передачи/приёма пакетов. Как и алгоритм шифрования, алгоритм контроля целостности, используемый с ESP, должен обеспечивать обработку пакетов, доставленных с нарушением порядка, и быть устойчивым к потере пакетов. Приведённые выше замечания по поводу использования незашифрованных данных для синхронизации получателя применимы и к алгоритму контроля целостности. По причине необязательности использования алгоритма, для него **может** быть установлено значение NULL.

Чтобы позволить реализации ESP рассчитать любое неявное заполнение, используемое алгоритмом контроля целостности, RFC для каждого алгоритма, используемого с ESP, должен указывать модуль заполнения для алгоритма.

3.2.3. Комбинированные алгоритмы

При использовании комбинированного алгоритма предоставляются услуги обеспечения конфиденциальности и целостности. Как и алгоритм шифрования, комбинированный алгоритм должен обеспечивать обработку пакетов, доставленных с нарушением порядка, и быть устойчивым к потере пакетов. Способы обеспечения комбинированным алгоритмом целостности данных, а также полей SPI и (Extended) Sequence Number, могут меняться в зависимости от алгоритма. Для обеспечения однородного и независимого от алгоритма решения по использованию комбинированных

[1]Отметим, что при использовании незашифрованной части заголовка для создания IV, эта информация может стать критичной для защиты и, таким образом, граница защиты, связанная с процессом шифрования, может расширяться. Например, при использовании ESP Sequence Number для определения IV логика генерации значения Sequence Number (программная или аппаратная) будет оцениваться, как часть реализации алгоритма. В случае FIPS 140-2 [NIST01] это будет существенно расширять границы оценки криптографического модуля.

алгоритмов, дополнительная структура полей данных не определяется. Например, поля SPI и Sequence Number могут реплицироваться в шифрованный «конверт», ICV может добавляться после трейлера ESP. Эти детали не видны снаружи.

Чтобы позволить ESP определять влияние комбинированного алгоритма на MTU, RFC для каждого алгоритма, используемого с ESP, должен указывать (простую) формулу, которая определяет размер шифрованных данных, как функцию размеров шифруемой информации и порядковых номеров.

3.3. Обработка исходящих пакетов

В транспортном режиме отправитель инкапсулирует информацию протокола следующего уровня между заголовком ESP и трейлером ESP, а также сохраняет указанный заголовок IP (и все расширенные заголовки IP в контексте IPv6). В туннельном режиме внешний и внутренний заголовки/расширения IP могут располагаться в различных вариантах. Создание внешнего заголовка/расширений IP на этапе инкапсуляции описано в документе по архитектуре защиты.

3.3.1. Нахождение SA

ESP применяется к исходящему пакету только после того, как реализация IPsec определяет, что пакет связан с SA, вызвавшей обработку ESP. Процесс определения применимой к исходящему трафику обработки IPsec описан в документе по архитектуре защиты.

3.3.2. Шифрование пакетов и расчёт ICV

В этом параграфе мы говорим, что шифрование используется всегда, поскольку оно оказывает влияние на форматирование. Следует понимать, что возможна работа «без шифрования» при использовании пустого (NULL) алгоритма шифрования (RFC 2410). Существует несколько вариантов алгоритмов.

3.3.2.1. Раздельные алгоритмы конфиденциальности и целостности

При использовании раздельных алгоритмов шифрования и контроля целостности отправитель выполняет перечисленные ниже операции.

- Инкапсуляция (в поле ESP Payload Data):
  - для транспортного режима - исходная информация протокола следующего уровня;
  - для туннельного режима - исходная дейтаграмма IP.
- Добавление требуемого заполнения - необязательное заполнение TFC и заполнение для шифрования.
- Шифрование результата с использованием ключа, алгоритма и режима, заданных для SA, и требуемых для криптографической синхронизации данных.
- Если указаны явные данные криптографической синхронизации (например, IV), они являются входной информацией для алгоритма шифрования в соответствии с его спецификацией и помещаются в поле Payload.
- Если используются неявные данные криптографической синхронизации, эти данные создаются и передаются на вход алгоритма шифрования в соответствии с его спецификацией.
- Если выбран контроль целостности, сначала выполняется шифрование, которое не затрагивает поля ICV. Такой порядок обработки упрощает быстрое обнаружение и отбрасывание повторно используемых или фиктивных пакетов до их расшифровки, потенциально снижая влияние атак на службы (DoS). Это также обеспечивает возможность параллельной обработки пакетов на принимающей стороне (т. е., дешифровка может выполняться одновременно с контролем целостности). Отметим, что результатом отсутствия защиты поля ICV с помощью шифрования, является необходимость использования для расчёта ICV алгоритма контроля целостности с поддержкой ключей.
- Расчет ICV для пакета ESP без учёта самого поля ICV. Таким образом, при вычислении ICV принимаются во внимание поля SPI, Sequence Number, Payload Data, Padding (если используется), Pad Length и Next Header[1]. Если для SA выбрана опция ESN, старшие 32 бита порядкового номера добавляются при расчёте после поля Next Header, но не передаются в пакете.

Для некоторых алгоритмов контроля целостности строка, для которой выполняется расчёт ICV, должна иметь размер, кратный значению, задаваемому алгоритмом. Если размер пакета ESP (перечисленные выше поля) не соответствует размеру блока, в конце пакета ESP **должно** добавляться неявное заполнение (после поля Next Header или после старших 32 битов порядкового номера, если используется ESN). Размер блока и, следовательно, величина заполнения задаётся спецификацией алгоритма контроля целостности. Заполнение не передаётся в пакете. Для определения необходимости использования неявного заполнения **требуется** обращаться к документу, определяющему алгоритм контроля целостности. Если этот документ не даёт ответа на вопрос, по умолчанию используется неявное заполнение в соответствии с принятым для алгоритма размером блока (размер пакета делается кратным размеру блока). Если заполнение требуется, но алгоритм не задаёт его содержимого, для заполнения **должны** использоваться октеты заполнения с нулевым значением.

3.3.2.2. Комбинированные алгоритмы конфиденциальности и целостности

При использовании комбинированных алгоритмов шифрования/контроля целостности отправитель выполняет перечисленные ниже операции.

- Инкапсуляция (в поле ESP Payload Data):
- для транспортного режима - исходная информация протокола следующего уровня;
- для туннельного режима - исходная дейтаграмма IP.

[1] Отметим, что 4 последних поля будут зашифрованы, поскольку шифрование выполняется до расчёта ICV.

- Добавление требуемого заполнения, включая необязательное заполнение TFC и заполнение для шифрования.
- Шифрование и защита целостности полученного результата с использованием ключа и комбинированного алгоритма, заданных для SA, а также требуемых данных криптографической синхронизации.
- Если указаны явные данные криптографической синхронизации (например, IV), они являются входной информацией для комбинированного алгоритма в соответствии с его спецификацией и помещаются в поле Payload.
- Если используются неявные данные криптографической синхронизации, эти данные создаются и передаются на вход алгоритма шифрования в соответствии с его спецификацией.
- Поля Sequence Number (или Extended Sequence Number) и SPI являются входной информацией для алгоритма, поскольку эти поля используются для контроля целостности. Это означает, что способ включения этих полей зависит от используемого комбинированного алгоритма и не включается в данный стандарт.
- Явное поле ICV **может** быть частью пакета ESP при использовании комбинированных алгоритмов. Если оно не используется, обычно в шифрованных данных имеется аналогичное поле. Расположение полей контроля целостности и способ включения полей Sequence Number и SPI в расчёт контрольной суммы **должны** определяться в RFC, содержащих спецификацию комбинированных алгоритмов, используемых с ESP.

3.3.3. Генерация порядковых номеров

Счётчик отправителя инициализируется нулевым значением при организации SA. Отправитель инкрементирует счётчик порядковых номеров (или ESN) для данной SA и помещает младшие 32 бита номера в поле Sequence Number. Таким образом, первый пакет для данной SA получает порядковый номер 1.

Если включена функция предотвращения повторного использования пакетов (включена по умолчанию), отправитель проверяет, не повторяется ли порядковый номер перед вставкой значения в поле Sequence Number. Иными словами, для отправителя **недопустимо** передавать пакет в SA, если эта передача будет приводить к повторному использованию порядкового номера. Попытка передачи пакета, которая будет вызывать переполнение (переход на новый цикл отсчёта) счётчика порядковых номеров приводит к внесению записи в журнал аудита. В эту запись **следует** включать значение SPI, текущую дату и время, адреса отправителя и получателя, а для IPv6 ещё и нешифрованное представление Flow ID.

Отправитель предполагает, что предотвращение повторного использования включено по умолчанию, пока получатель однозначно не укажет обратное (см. параграф 3.4.3) или эта функция была отключена вручную при выборе конфигурации SA. Таким образом, в типичном случае реализация ESP говорит отправителю о необходимости организации новой SA, когда значение Sequence Number (или ESN) достигает максимума и должно вернуться к нулю.

Если ключи, используемые для расчёта ICV, распространяются вручную, приложениям **не следует** использовать услуги по предотвращению повтора пакетов. Если пользователь выбрал поддержку предотвращения повторного использования, счётчик порядковых номеров на стороне отправителя **должен** обеспечивать корректность значений при локальных перезагрузках и т. п., пока ключ не будет сменен (см. раздел 5).

Если функция предотвращения повторов отключена (как описано выше) отправителю не нужно заботиться о мониторинге переполнения (сброса в 0) счётчика порядковых номеров. Однако отправитель будет по-прежнему инкрементировать значение счётчика и после максимального значения счётчик будет сброшен в 0. Такой вариант поведения рекомендуется для групповых SA со множеством отправителей, если между отправителями и получателями не согласовано использование механизма предотвращения повторов (выходящего за рамки данного стандарта).

Если выбрано использование ESN (см. Приложение), в поле Sequence Number передаются только 32 младших бита расширенного порядкового номера, хотя отправитель и получатель поддерживают полные 64-битовые счетчики ESN. При этом старшие 32 бита порядкового номера учитываются алгоритмом контроля чётности (например, эти 32 бита могут добавляться при расчёте контрольной суммы после поля Next Header, если реализован отдельный алгоритм контроля целостности).

Примечание. Если отправитель отказался от использования функции предотвращения повторов для SA, ему **не следует** согласовывать использование ESN в протоколе управления SA. Использование ESN вызывает у получателя необходимость поддержки окна anti-replay (для определения корректного значения старших битов ESN, которые используются при расчёте ICV), что вступает в противоречие с отказом от предотвращения повторов для SA.

3.3.4. Фрагментация

Если требуется фрагментация IP, она выполняется после обработки ESP в реализации IPsec. Таким образом, в транспортном режиме ESP применяется только к целым дейтаграммам IP[1] (не фрагментам). Пакет IPv4, к которому применили ESP, может быть фрагментирован маршрутизаторами на пути и в таком случае фрагменты должны быть собраны до обработки ESP на приёмной стороне (этого не возникает для IPv6, где фрагментация по инициативе маршрутизаторов невозможна). В туннельном режиме ESP применяется к пакетам IP, содержимое которых может представлять собой фрагменты пакетов IP. Например, шлюз или реализация IPsec bump-in-the-stack или bump-in-the-wire (см. документ по архитектуре защиты) может использовать ESP для таких фрагментов в туннельном режиме.

Фрагментация, выполняемая реализацией IPsec или маршрутизаторами на пути доставки между партнёрами IPsec, существенно снижает производительность. Более того, необходимость сборки фрагментов на приёмной стороне до выполнения операций ESP, порождает возможность организации атак на отказ служб. Таким образом, реализация ESP **может** выбрать отказ от поддержки фрагментации и маркировать передаваемые пакеты флагом DF[2] для облегчения

[1]Как отмечено в конце параграфа 3.1.1, реализации bump-in-the-stack и bump-in-the-wire могут сначала выполнять сборку фрагментов, созданных локальным уровнем IP, потом выполнять обработку IPsec и снова фрагментировать полученный в результате пакет.
В случае IPv6 реализации bump-in-the-stack и bump-in-the-wire должны проверять все расширенные заголовки, а также значения флага More и поля Fragment Offset для обнаружения фрагментирования. Если фрагментирование используется, пакеты должны быть собраны до выполнения операций IPsec.
[2]Не фрагментировать. *Прим. перев.*

определения PMTU[1]. В любом случае, реализация ESP **должна** поддерживать генерацию сообщений ICMP PMTU (или использование эквивалентной внутренней сигнализации) для минимизации издержек на фрагментирование. Детали требований, связанных с фрагментацией рассматриваются в документе по архитектуре защиты.

3.4. Обработка входящих пакетов

3.4.1. Сборка фрагментов

Если нужна сборка фрагментов[2], она выполняется до обработки AH. Если переданный на обработку AH пакет оказывается фрагментом IP (т. е., поле Offset имеет ненулевое значение или установлен флаг More Fragments), получатель **должен** отбрасывать такой пакет и делать запись в журнале аудита. В запись **следует** включать значение SPI, дату и время, адреса отправителя и получателя, а также Flow ID для IPv6.

3.4.2. Нахождение SA

При получении пакета, содержащего заголовок ESP, получатель определяет подходящую (одностороннюю) SA путём просмотра SAD. Для индивидуальных SA определение основано на значении SPI, в дополнение к которому может использоваться поле протокола, как описано в параграфе 2.1. Если реализация поддерживает групповой трафик, при определении SA используется также адрес получателя (в дополнение к SPI) и может применяться адрес отправителя, как описано в параграфе 2.1 (более подробно этот процесс описан в документе по архитектуре защиты). Запись SAD для SA показывает также использование поля Sequence Number и его размер (32 или 64 бита) для данной SA. Кроме того, запись SAD для SA задаёт алгоритм(ы), используемый для расчёта ICV и показывает, нужно ли проверять значения ICV.

Если для пакета не найдено защищённой связи, получатель должен отбросить пакет с записью в журнал аудита. В запись **следует** включать значение SPI, дату и время, адреса отправителя и получателя, а также Flow ID для IPv6.

Отметим, что трафик управления SA (такой, как пакеты IKE) не требуется обрабатывать на базе SPI, т. е., этот трафик может демультиплексироваться отдельно (например, на основе полей Next Protocol и Port).

3.4.3. Проверка порядковых номеров

Все реализации ESP **должны** поддерживать предотвращение повторного использования пакетов[3], хотя использование этой функции может быть включено или отключено получателем на уровне SA. Этот сервис **недопустимо** включать, пока не включены услуги контроля целостности ESP для данной SA, поскольку без этого не обеспечивается защита целостности поля Sequence Number. Функции предотвращения повторного использования применимы как к индивидуальным, так и к групповым SA. Однако данный стандарт не задаёт механизмов защиты от повторного использования пакетов для SA со множеством отправителей (групповых или индивидуальных). При отсутствии согласования (или настройки вручную) механизма предотвращения повторного использования для таких SA отправителю и получателю рекомендуется проверить запрет использования поля Sequence Number для таких SA (запрет организуется путём согласования или вручную), как описано ниже.

Если получатель не включил предотвращение повторного использования для SA, на входе не проверяются значения поля Sequence Number. Однако с точки зрения отправителя предотвращение повторного использования по умолчанию включено. Чтобы избавить отправителя от ненужной передачи и мониторинга порядковых номеров (см. параграф 3.3.3), получателю **следует** уведомить отправителя об отказе от поддержки предотвращения повторного использования на этапе организации SA, если применяется протокол организации SA.

Если получатель включил предотвращение повторного использования для SA, он **должен** установить значение счётчика пакетов для данной SA нулевым на момент организации SA. Для каждого принятого пакета получатель **должен** проверять, что поле Sequence Number в пакете не совпадает с порядковым номером ни одного из пакетов, полученных в данной SA. Эту проверку **следует** проводить до выполнения каких-либо операций ESP по отношению к данному пакету сразу после проверки принадлежности пакета к SA для ускорения отбрасывания дубликатов.

ESP позволяет двухэтапную проверку порядковых номеров в пакетах. Такая возможность важна для реализаций ESP (обычно для их шифровальной части), не способных выполнять расшифровку и/или проверку целостности пакетов со скоростью интерфейса подключения к незащищённой сети. Если реализация может работать со скоростью интерфейса, выполнение описанной ниже предварительной проверки становится ненужным.

Предварительная проверка поля Sequence Number обусловлена использованием порядкового номера в заголовке ESP и выполняется до проверки целостности и расшифровки пакета. Если предварительная проверка дала отрицательный результат, пакет отбрасывается, что позволяет избавиться от ненужных криптографических операций на приёмной стороне. При положительном результате проверки получатель ещё не может менять свой локальный счётчик, поскольку целостность порядкового номера пока не подтверждена.

Дубликаты отбрасываются с помощью «скользящего» окна приёма. Реализация такого окна осуществляется локально, но описанная ниже функциональность должна поддерживаться всем реализациями.

«Правый» край окна представляет наибольшее проверенное значение поля Sequence Number для данной SA. Пакеты с номерами, выходящими за «левый» край окна, отбрасываются. Попадающие в окно пакеты проверяются на предмет совпадения порядковых номеров с номерами принятых пакетов для окна. При использовании опции ESN для SA явно передаются только младшие 32 бита расширенного порядкового номера, но получатель использует и старшие 32 бита номера для SA (от локального счётчика) при проверке порядковых номеров. При восстановлении полного порядкового номера, если значение младших 32 битов порядкового номера из принятого пакета меньше младших 32 битов значения счетчика порядковых номеров на стороне получателя, последний предполагает, что значение старших 32 битов номера было инкрементировано, т. е., перемещает номер в новое «подпространство». Этот алгоритм допускает

[1]Path MTU - размер максимального передаваемого блока для пути.
[2]При сборке пакетов текущая спецификация IPv4 **не** требует обнуления поля Offset и сброса флага More Fragments. Для корректной обработки собранных из фрагментов пакетов IPsec (вместо отбрасывания, принятого для фрагментов) реализация IP должна выполнять обе указанные операции после сборки пакета из фрагментов.
[3]Anti-replay service.

интервал приёма для отдельной SA до 2^{32}-1 пакетов. Если интервал становится больше, **могут** использоваться эвристические проверки для ресинхронизации порядковых номеров на приёмной стороне, как описано в Приложении).

Если полученный пакет попадает в окно и не является дубликатом или пакет относится к правому краю окна и применяется отдельный алгоритм контроля целостности, получатель выполняет проверку целостности. Если используется комбинированный алгоритм, проверка целостности выполняется вместе с дешифрованием. При отрицательном результате проверки целостности получатель **должен** отбросить полученную дейтаграмму IP, как некорректную, с записью в журнал аудита. В запись **следует** включать значение SPI, дату и время, адреса отправителя и получателя, а также Flow ID для IPv6. Окно приёма обновляется только при положительном результате проверки целостности (при использовании комбинированного алгоритма защищённое значение поля Sequence Number должно также совпадать с порядковым номером защиты от повторного использования пакетов).

Должны поддерживаться окна **минимального** размера в 32 пакета, но по умолчанию **следует** поддерживать окна размером 64 пакета. Получатель **может** выбирать другие размеры окна (больше **минимального**). Получатель **не** информирует отправителя о выбранном размере окна. Для высокоскоростных сред размер окна приёма следует увеличивать. Минимальные и рекомендуемые размеры окна для высокоскоростных (например, мультимегабитных) устройств данный стандарт не задаёт.

3.4.4. Проверка ICV

Как и при обработке исходящих пакетов, имеется несколько вариантов, зависящих от используемого алгоритма.

3.4.4.1. Раздельные алгоритмы конфиденциальности и целостности

При использовании раздельных алгоритмов конфиденциальности и целостности выполняются перечисленные ниже операции.

1. Если выбрана функция контроля целостности, получатель рассчитывает значение ICV для пакета ESP без самого поля ICV, используя заданный алгоритм контроля целостности и сравнивает полученное значение со значением поля ICV а пакете. Подробное описание расчёта приведено ниже.

   Если рассчитанное значение ICV совпадает с полученным в пакете, это говорит о корректности дейтаграммы и она принимается. При отрицательном результате проверки получатель **должен** отбросить полученную дейтаграмму IP, как некорректную, и внести запись в журнал аудита. В запись **следует** включать значение SPI, дату и время получения пакета, адреса получателя и отправителя, порядковый номер и незашифрованное значение Flow ID (для IPv6).

Примечание для разработчиков.

Разработчики могут использовать любую последовательность действий, которая даёт такой же результат, как перечисленные здесь операции. Сначала значение ICV из принятого пакета сохраняется и заменяется нулём. После этого проверяется общий размер пакета ESP без учёта ICV. Если требуется неявное заполнение по размеру блока алгоритма контроля целостности, добавляются нулевые[1] байты в конце пакета ESP сразу же вслед за полем Next Header или после старших 32 битов порядкового номера в случае использования ESN. Выполняется расчёт ICV и полученный результат сравнивается с сохраненным значением. Правила сравнения задаются спецификацией алгоритма контроля целостности.

2. Получатель дешифрует в пакете ESP поля Payload Data, Padding, Pad Length и Next Header, используя ключ, алгоритм, режим и данные криптографической синхронизации (если они нужны) в соответствии с SA. Как и в параграфе 3.3.2, мы говорим о том, что шифрование применяется всегда, поскольку оно влияет на формат. При этом предполагается, что может использоваться режим «без шифрования» с пустым (NULL) алгоритмом шифрования (RFC 2410).

   - Если явно указаны данные криптографической синхронизации (например, IV), эти данные берутся из поля Payload и передаются на вход алгоритма дешифровки в соответствии со спецификацией алгоритма.
   - Если указаны неявные данные криптографической синхронизации, создаётся локальная версия IV и передаётся на вход алгоритма дешифровки в соответствии со спецификацией алгоритма.

3. Получатель обрабатывает поля заполнения (Padding) в соответствии со спецификацией алгоритма шифрования. Если используется принятая по умолчанию схема заполнения (см. параграф 2.4, получателю **следует** проверить поле Padding до удаления заполнения перед передачей расшифрованных данных на следующий уровень.
4. Получатель проверяет поле Next Header. Если значение поля равно 59 (нет следующего заголовка), (фиктивный) пакет отбрасывается без дальнейшей обработки.
5. Получатель восстанавливает исходную дейтаграмму IP:

   - для транспортного режима - внешний заголовок IP плюс исходная информация протокола следующего уровня в поле ESP Payload;
   - для туннельного режима - вся дейтаграмма IP в поле ESP Payload.

   Конкретные действия по восстановлению исходной дейтаграммы зависят от режима (транспортный или туннельный) и описаны в документе по архитектуре защиты. По минимуму в контексте IPv6 получателю **следует** обеспечить для расшифрованных данных выравнивание по 8-байтовой границе для упрощения обработки протокола, указанного в поле Next Header. При этой обработке «отбрасывается» все (необязательное) заполнение TFC[2], которое было добавлено для обеспечения конфиденциальности потока трафика.

[1] Значения байтов должны устанавливаться в соответствии со спецификацией алгоритма контроля целостности, т. е., не обязаны быть нулевыми во всех случаях. *Прим. перев.*

[2] Если это заполнение используется, оно размещено после дейтаграммы IP (или кадра транспортного уровня) и перед полем Padding (см. параграф 2.4).

Если проверка целостности выполняется параллельно с дешифровкой[3], контроль целостности **должен** быть завершен до передачи расшифрованного пакета на дальнейшую обработку. Такой порядок упрощает быстрое детектирование и отбрасывание повторных или обманных пакетов до их дешифровки, что потенциально может снижать воздействие атак на службы.

3.4.4.2. Комбинированные алгоритмы конфиденциальности и целостности

При использовании комбинированного алгоритма защиты конфиденциальности и целостности получатель выполняет перечисленные ниже операции.

1. Дешифровка и проверка целостности полей ESP Payload Data, Padding, Pad Length, Next Header с использованием ключа, алгоритма, режима и данных криптографической синхронизации (если они нужны), указанных SA. Значение SPI из заголовка ESP и значения счётчика пакетов на стороне получателя (преобразованное в соответствии с требованиями параграфа 3.4.3) являются входными данными для этого алгоритма, поскольку нужны для проверки целостности.

   - Если явно указаны данные криптографической синхронизации (например, IV), эти данные берутся из поля Payload и передаются на вход алгоритма дешифровки в соответствии со спецификацией алгоритма.

   - Если указаны неявные данные криптографической синхронизации, создаётся локальная версия IV и передаётся на вход алгоритма дешифровки в соответствии со спецификацией алгоритма.

2. При отрицательном результате проверки целостности, проведённой комбинированным алгоритмом получатель **должен** отбросить полученную дейтаграмму IP, как некорректную, внеся запись в журнал аудита. В журнальную запись **следует** включать значение SPI, дату и время приёма дейтаграммы, адреса отправителя и получателя, порядковый номер, а также нешифрованное значение Flow ID для IPv6.

3. Обработка поля заполнения (Padding) в соответствии со спецификацией алгоритма шифрования, если алгоритм уже не выполнил эту операцию.

4. Проверка поля Next Header. Если значение поля равно 59 (нет следующего заголовка), (фиктивный) пакет отбрасывается без дальнейшей обработки.

5. Восстановление исходной дейтаграммы IP (туннельный режим) или кадра транспортного уровня (транспортный режим) из поля ESP Payload Data. При этой операции неявно отбрасывается любое (необязательное[2]) заполнение, используемое для защиты конфиденциальности потока трафика.

4. Аудит

Не все системы, поддерживающие ESP, реализуют аудит. Однако, если ESP встраивается в систему, поддерживающую аудит, реализация ESP **должна** поддерживать аудит и также **должна** позволять администратору системы включать и отключать аудит для ESP. В большинстве случаев гранулярность аудита определяется локально. Однако некоторые события, заносимые в журнал аудита, задаются данной спецификацией и для каждого из этих событий указывается минимальный набор информации, которую **следует** включать в журнал аудита.

- Для сессии нет корректной защищённой связи (SA). В журнал аудита **следует** включать значение SPI, дату и время приёма дейтаграммы, адреса отправителя и получателя, порядковый номер, а также нешифрованное значение Flow ID для IPv6.

- Пакет, предложенный для обработки ESP, представляется фрагментом IP (отличное от нуля значение поля OFFSET или установлен флаг MORE FRAGMENTS). В журнал аудита **следует** включать значение SPI, дату и время приёма дейтаграммы, адреса отправителя и получателя, порядковый номер, а также нешифрованное значение Flow ID для IPv6.

- Попытка передачи пакета, ведущая к переполнению счётчика порядковых номеров. В журнал аудита **следует** включать значение SPI, дату и время приёма дейтаграммы, адреса отправителя и получателя, порядковый номер, а также нешифрованное значение Flow ID для IPv6.

- Полученный пакет не прошёл проверки на повторное использование. В журнал аудита **следует** включать значение SPI, дату и время приёма дейтаграммы, адреса отправителя и получателя, порядковый номер, а также нешифрованное значение Flow ID для IPv6.

- Не прошла проверка целостности. В журнал аудита **следует** включать значение SPI, дату и время приёма дейтаграммы, адреса отправителя и получателя, порядковый номер, а также нешифрованное значение Flow ID для IPv6.

В записи **можно** также включать дополнительную информацию и **можно** указывать в журнале информацию о других событиях, которые явно не упомянуты в данной спецификации. Получатель не обязан уведомлять отправителя о внесении записей в журнал аудита, поскольку такое требование создавало бы возможность организации атак на отказ служб.

5. Соответствие требованиям

Реализации, которые заявляют о своём соответствии или совместимости с данной спецификацией, **должны** полностью реализовать синтаксис и обработку ESP, описанные здесь, для индивидуального трафика, а также **должны** полностью выполнять все требования документа по архитектуре защиты [Ken-Arch]. В дополнение к этому, реализации, заявляющие поддержку группового трафика, **должны** соответствовать всем дополнительным требованиям, заданным для такого трафика. При ручном распределении ключей, используемых для расчёта ICV, корректная работа системы предотвращения повторного использования пакетов требует аккуратной поддержки состояния счётчика на передающей стороне при замене ключа, поскольку в этом случае невозможно восстановить работу после

[3]При параллельной дешифровке и проверке целостности необходимо принимать меры по предотвращению «гонки» в части доступа к пакету и преобразования дешифрованного пакета.

переполнения счётчика. Таким образом, совместимым со спецификацией реализациям **не следует** предоставлять такой сервис для SA с распространением ключей вручную.

Обязательные для реализации алгоритмы, используемые с ESP, описаны в отдельном документе [Eas04], для обеспечения возможности обновления алгоритмов независимо от протокола. Кроме обязательных для ESP алгоритмов **могут** поддерживаться дополнительные алгоритмы.

Поскольку шифрование в ESP не является обязательным, требуется поддержка «пустого» (NULL) алгоритма шифрования для обеспечения совместимости способов согласования сервиса ESP. Поддержка только услуг защиты конфиденциальности является опциональной. Если реализация предлагает такой сервис, она **должна** поддерживать использование пустого (NULL) алгоритма контроля целостности. Отметим, что хотя услуги защиты целостности и конфиденциальности сами по себе могут использовать алгоритм NULL при указанных выше условиях, **недопустимо** выбирать NULL для обоих услуг сразу.

6. Вопросы безопасности

Безопасность является основным аспектом данного протокола и вопросы безопасности рассматриваются во всем документе. Дополнительные аспекты использования протокола IPsec, связанные с обеспечением безопасности, рассматриваются в документе по архитектуре защиты.

7. Отличия от RFC 2406

Этот документ имеет несколько существенных отличий от RFC 2406.

- Предоставление только услуг защиты конфиденциальности - в данном документе **возможно**, а не **обязательно**.
- Изменено определение SPI для обеспечения возможности однотипного поиска в SAD для индивидуальных и групповых SA, совместимого со многими технологиями групповой передачи. Для выбора индивидуальных SA значение SPI может использоваться само по себе или в комбинации с протоколом по усмотрению получателя. Для выбора групповых SA значение SPI объединяется с адресом отправителя (и, опционально, с адресом получателя).
- Добавлены расширенные порядковые номера (ESN) для обеспечения 64-битовой нумерации на высокоскоростных соединениях. Разъяснены требования к отправителю и получателю для групповых SA и защищённых связей с множеством отправителей.
- Поле Payload - расширена модель для использования комбинированных алгоритмов.
- Заполнение для повышения конфиденциальности потока трафика - добавлено требование обеспечения возможности добавления байтов после завершения данных IP Payload и до начала поля Padding.
- Next Header – добавлено требование по обеспечению возможности генерации и отбрасывания фиктивных пакетов заполнения (Next Header = 59).
- ICV - расширена модель с учётом использования комбинированных алгоритмов.
- Алгоритмы - добавлена поддержка комбинированных алгоритмов защиты конфиденциальности.
- Ссылки на обязательные алгоритмы вынесены в отдельный документ.
- Обработки исходящих и входящих пакетов - сейчас существуют два варианта: (1) с раздельными алгоритмами защиты конфиденциальности и целостности и (2) с комбинированным алгоритмом. Добавление комбинированных алгоритмов привело к созданию разделов шифрования/дешифровки и контроля целостности для обработки как входящих, так и исходящих пакетов.

8. Совместимость с ранними версиями

В ESP нет номера версии и механизмов, позволяющих партнёрам IPsec определять и согласовывать используемые версии ESP. В этом параграфе рассмотрены вопросы совместимости с более ранними версиями.

Во-первых, если не реализовано ни одной из новых возможностей ESP v3, формат пакетов ESP будет идентичен для ESP v2 и ESP v3. При использовании комбинированного алгоритма (поддерживается только в ESP v3) формат пакетов может отличаться от формата пакетов ESP v2. Однако партнёр, поддерживающий только ESP v2 не будет согласовывать алгоритм, поскольку он определён только для использования в контексте ESP v3.

Согласование расширенных порядковых номеров (ESN) поддерживается IKE v2 и может быть решено для IKE v1 за счёт добавления ESN Addendum к IKE v1 DOI[1].

В новом ESP (v3) появились два новых метода повышения уровня конфиденциальности потоков трафика (TFC):

- произвольное заполнение после окончания пакета IP;
- условное отбрасывание с использованием Next Header = 59.

Первая возможность относится к тем, которые могут вызывать проблемы на приёмной стороне, поскольку поле общего размера пакета IP будет говорить о завершении пакета. Таким образом, все байты заполнения TFC после завершения пакета следует удалять в той же точке при обработке пакета IP после выполнения операций ESP, даже если программы IPsec не удалили это заполнение. Таким образом, эту возможность ESP v3 отправитель может применять независимо от того, использует получатель ESP v2 или ESP v3.

Вторая возможность позволяет отправителю передавать в данных произвольные строки байтов, которые не обязаны быть корректно сформированными пакетами IP (внутри туннеля для целей TFC). Возникает вопрос, что будет делать

[1]Domain of Interpretation - область интерпретации.

получатель ESP v2, когда поле Next Header в заголовке пакета ESP будет иметь значение 59. Он может отбросить пакет, обнаружив некорректный заголовок IP, и внести запись в журнал аудита и может продолжить нормальную работу, поскольку иное поведение создавало бы уязвимость к DoS-атакам с использованием трафика от аутентифицированных узлов. Таким образом, эта возможность является оптимизацией, которую отправитель ESP v3 может использовать независимо от того, какая версия реализована на приёмной стороне - ESP v2 или ESP v3.

9. Благодарности

Автор благодарит Ran Atkinson, чей вклад был очень важен на начальных этапах разработки IPsec, а также разработчиков первой серии стандартов IPsec RFC 1825 - 1827. Karen Seo заслуживает особой благодарности за помощь при редактировании этой и предыдущей версии спецификации. Автор также благодарит членов рабочих групп IPSEC и MSEC, которые внесли свой вклад в развитие спецификации протокола.

10. Литература

10.1. Нормативные документы

[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, O., «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. Дополнительная литература

[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[1], 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, O. 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 порядковые номера (64 бита)

A1. Обзор

В этом приложении описана схема расширенной порядковой нумерации (ESN) для IPsec (ESP и AH), где используются 64-битовые порядковые номера, но в каждом пакете передаются только младшие 32 бита номера. Описана схема окна, используемого для обнаружения повторно используемых пакетов, а также механизм определения старших битов порядкового номера, используемых для отбрасывания пакетов и расчёта ICV. Описан также механизм обработки случаев потери синхронизации для старших (не передаваемых в пакетах) битов порядкового номера.

A2. Окно Anti-Replay

Получатель будет поддерживать окно предотвращения повторного использования пакетов размером W[2]. Это окно будет ограничивать степень разупорядочивания пакетов при доставке без потери аутентификации. Все 2^{32} порядковых номера, связанных с любым фиксированным значением старших 32 битов (Seqh) будем называть подпространством порядковых номеров. В приведённой ниже таблице перечислены используемые переменные и даны их определения.

Имя	Размер в битах	Значение
W	32	Размер окна
T	64	Наибольший порядковый номер, аутентифицированный до настоящего времени - верхняя граница окна
Tl	32	32 младших бита T
Th	32	32 старших бита T
B	64	Нижняя граница окна
Bl	32	32 младших бита B
Bh	32	32 старших бита B

[1]Работа завершена и опубликована в RFC 4607. *Прим. перев.*

[2]Для окна при расширенной нумерации не вводится дополнительных требований по поддержке минимального или рекомендуемого размера окна, сверх тех требований (32 и 64 пакета, соответственно), которые уже заданы для окна при 32-битовой нумерации. Однако разработчикам предлагается масштабировать размер окна в соответствии со скоростью интерфейсов, поддерживаемых реализацией, использующей опцию ESN. Описанный ниже механизм предполагает, что размер окна не превышает 2^{31} пакета.

Seq	64	Порядковый номер полученного пакета
Seql	32	32 младших бита Seq
Seqh	32	32 старших бита Seq

При выполнении проверки на предмет повторного использования пакетов или определении старших битов номера для аутентификации входящего пакета возможны два случая.

- Случай A: Tl >= (W - 1) все окно находится в одном подпространстве порядковых номеров (Рисунок 3)

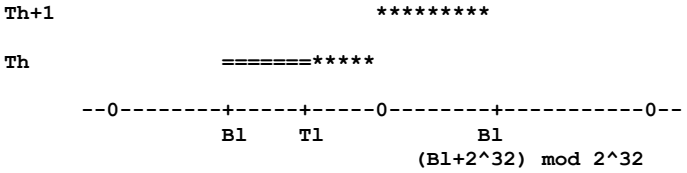


Рисунок 3. Случай A.

- Случай B: Tl < (W - 1) окно захватывает части двух смежных подпространств порядковых номеров (Рисунок 4).

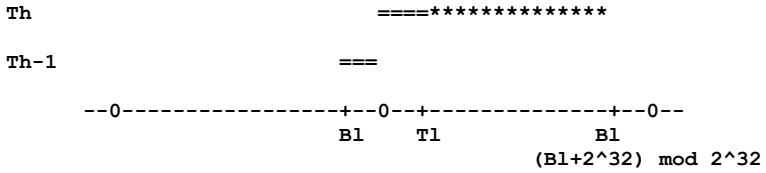


Рисунок 4. Случай B.

На рисунках нижняя линия ---- показывает смежные подпространства порядковых номеров, а 0 указывает начало каждого подпространства. Короткая двойная линия === показывает применимые старшие биты, а ==== представляет окно. Звёздочки **** обозначают грядущие номера, т. е., номера номера, превышающие максимальный аутентифицируемый в данный момент номер (ThTl).

A2.1. Использование окна Anti-Replay и управление им

Окно предотвращения повторов можно рассматривать как строку битов размером W (W = T - B + 1 и не может превышать 2^{32} - 1). Младший бит строки соответствует B, а старший - T и каждый порядковый номер от Bl до Tl представлен соответствующим битом. Значение бита показывает, был ли пакет с соответствующим номером принят и аутентифицирован, что позволяет обнаружить и отбросить повторные пакеты.

При получении и проверке корректности пакета с 64-битовым порядковым номером (Seq), превышающим T:

- B увеличивается на (Seq - T);
- отбрасываются (Seq - T) битов в левой части окна;
- добавляются (Seq - T) битов в правой части окна;
- устанавливается «верхний» бит для индикации приёма и аутентификации пакета с данным порядковым номером;
- сбрасываются новые биты между T и «верхним» битом для индикации отсутствия принятых пакетов с соответствующими порядковыми номерами;
- для T устанавливается значение нового порядкового номера.

Проверка пакетов на предмет повторного использования.

- Случай A: Если Seql >= Bl (где Bl = Tl - W + 1) **И** Seql <= Tl, проверяется соответствующий бит окна. Если пакет с номером Seql уже был принят (бит окна установлен), он отбрасывается. В противном случае проверяется целостность пакета. Проверка старших битов номера (Seqh) описана в параграфе A2.2.

Случай B: Если Seql >= Bl (где Bl = Tl - W + 1) **ИЛИ** Seql <= Tl, проверяется соответствующий бит окна. Если пакет с номером Seql уже был принят (бит окна установлен), он отбрасывается. В противном случае проверяется целостность пакета. Проверка старших битов номера (Seqh) описана в параграфе A2.2.

A2.2. Определение старших битов (Seqh) порядкового номера

```
+ Для случая A (Рисунок 3):
  Если Seql >= Bl (где Bl = Tl - W + 1), то Seqh = Th
  Если Seql <  Bl ( где Bl = Tl - W + 1), то Seqh = Th + 1

+ Для случая B (Рисунок 4):
  Если Seql >= Bl ( где Bl = Tl - W + 1), то Seqh = Th - 1
  Если Seql <  Bl ( где Bl = Tl - W + 1), то Seqh = Th
```

Поскольку в пакетах передаётся только значение Seql, получатель должен отслеживать подпространство порядковых номеров для каждого пакета (т. е., определять значение Seqh). Приведённые справа уравнения определяют выбор Seqh в «нормальных» условиях. В параграфе B3 рассматривается определение старших битов номера в условиях экстремальных потерь пакетов.

A2.3. Пример псевдокода

Приведённый ниже псевдокод иллюстрирует описанные выше алгоритмы предотвращения повторного использования и контроля целостности пакетов. Значения Seql, Tl, Th и W являются 32-битовыми целыми числами без знака. Используется арифметика по модулю 2^{32}.

```
Если (Tl >= W - 1)                          Случай A
    Если (Seql >= Tl - W + 1)
        Seqh = Th
```

```
            Если (Seql <= Tl)
                Если (проверка на предмет повтора прошла)
                    Если (проверка целостности прошла)
                        Установить бит, соответствующий Seql
                        Принять пакет
                    Иначе отбросить пакет
                Иначе отбросить пакет
            Иначе
                Если (проверка целостности прошла)
                    Tl = Seql (shift bits)
                    Установить бит, соответствующий Seql
                    Принять пакет
                Иначе отбросить пакет
        Иначе
            Seqh = Th + 1
            Если (проверка целостности прошла)
                Tl = Seql (shift bits)
                Th = Th + 1
                Установить бит, соответствующий Seql
                Принять пакет
            Иначе отбросить пакет
    Иначе                                   Случай B
        Если (Seql >= Tl - W + 1)
            Seqh = Th - 1
            Если (проверка на предмет повтора прошла)
                Если (pass integrity check)
                    Установить бит, соответствующий Seql
                    Принять пакет
                Иначе отбросить пакет
            Иначе отбросить пакет
        Иначе
            Seqh = Th
            Если (Seql <= Tl)
                Если (проверка на предмет повтора прошла)
                    Если (проверка целостности прошла)
                        Установить бит, соответствующий Seql
                        Принять пакет
                    Иначе отбросить пакет
                Иначе отбросить пакет
            Иначе
                Если (проверка целостности прошла)
                    Tl = Seql (shift bits)
                    Установить бит, соответствующий Seql
                    Принять пакет
                Иначе отбросить пакет
```

A3. Обработка потери синхронизации в результате больших потерь пакетов

При потере 2^{32} или более пакетов подряд для одной SA отправитель и получатель теряют синхронизацию старших битов порядкового номера, т. е., уравнения параграфа B2.2 не будут давать корректного значения. Пока эта проблема не будет обнаружена и разрешена, последующие пакеты для данной SA не могут быть аутентифицированы и будут отбрасываться. Описанную ниже процедуру восстановления синхронизации **следует** поддерживать во всех реализациях IPsec (ESP или AH), которые работают с ESN.

Отметим, что описанный вариант экстремальных потерь представляется маловероятным для SA, использующих протокол TCP, поскольку отправитель, не получающий пакетов ACK в ответ на переданные пакеты, будет останавливать передачу до того, как будут потеряны 2^{32}. И другие приложения с двухсторонним обменом данными (даже работающие по протоколу UDP) при таких экстремальных потерях будут включать тот или иной тайм-аут. Однако приложения с односторонним потоком трафика, работающие по протоколу UDP, могут не поддерживать средств автоматического детектирования экстремальных потерь пакетов и, следовательно, требуется обеспечить метод восстановления для таких ситуаций.

Предлагаемое решение призвано:

- минимизировать влияние на обработку нормального трафика;
- предотвратить создание новой возможности организации атак на отказ служб за счет неоправданной затраты ресурсов на ресинхронизацию;
- реализовать механизм восстановления только на принимающей стороне, поскольку отправитель обычно не знает, для каких порядковых номеров получателю требуется восстановление синхронизации; реализация механизмов восстановления на приёмной стороне является предпочтительной; кроме того, такое решение обеспечивает совместимость с ранними версиями.

A3.1. Включение ресинхронизации

Для каждой SA получатель запоминает число последовательных пакетов, для которых не прошла аутентификация. Это значение используется для включения процесса ресинхронизации, который следует выполнять в фоновом режиме или на отдельном процессоре. Приём корректного пакета для данной SA ведёт к сбросу счётчика некорректных пакетов в 0. Значение, при котором включается ресинхронизация, является локальным параметром. Не требуется поддерживать независимые значения порога ресинхронизации для каждой SA, но реализация вправе поддерживать их.

A3.2. Процесс ресинхронизации

Когда значение счётчика некорректных пакетов достигает заданного порога, выбирается «плохой» пакет, для которого процедура аутентификации повторяется с использованием следующего большего значения для старшей части

расширенного порядкового номера (Seqh). Значение старшей части номера увеличивается на 1 при каждой проверке. Число попыток проверки следует ограничивать на случай того, что выбранный для проверки пакет оказался «из прошлого» или является поддельным. Максимальное число попыток задаётся локальным параметром. Поскольку значение Seqh неявно помещается после данных AH (или ESP), может оказаться возможной оптимизация процедуры восстановления за счёт выполнения процедуры контроля целостности пакета с использованием нарастающих значений Seqh для расчёта ICV. При успешной аутентификации пакета с помощью описанной процедуры значение счётчика некорректных пакетов сбрасывается и устанавливается значение T, определённое по прошедшему проверку пакету.

Это решение требуется поддерживать только на приёмной части, следовательно, оно обеспечивает совместимость с прежними версиями. Поскольку процедура ресинхронизации осуществляется в фоновом режиме или выполняется на отдельном процессоре, она не будет оказывать влияния на обработку остального трафика и не создаёт дополнительной возможности организации атак на службы путём отвлечения ресурсов 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

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

Полное заявление авторских прав

Интеллектуальная собственность

Подтверждение

Финансирование функций RFC Editor обеспечено Internet Society.