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

S. Kent
BBN Technologies
December 2005

# Аутентификационный заголовок IP

**IP Authentication Header**

**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.

**Авторские права**



**Аннотация**

В этом документе описана обновлённая версия аутентификационного заголовка IP (AH[1]), который разработан для обеспечения услуг проверки подлинности (аутентификации) в IPv4 и IPv6. Этот документ отменяет действие RFC 2402 (ноябрь 1998).

# Оглавление



[1] IP Authentication Header - аутентификационный заголовок IP.

# 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 защищенных 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].

Аутентификационный заголовок IP (AH) используется для обеспечения целостности и аутентификации источника данных для дейтаграмм IP (далее для краткости будет использоваться термин «целостность») без организации специальных соединений и защиты против повторного использования пакетов. Второй, необязательный, сервис может выбираться получателем при создании защищенной связи (SA[2]). Протокол по умолчанию требует от отправителя увеличивать порядковые номера для предотвращения повторного использования пакетов, но этот механизм работает только при проверке порядковых номеров на приемной стороне. Для использования расширенных возможностей порядковой нумерации AH вносит требование к протоколу управления SA по обеспечению возможности согласования этой новой функции (см. параграф 2.5.1).

AH обеспечивает аутентификацию для всех возможных частей заголовка, а также для данных протокола следующего уровня. Однако некоторые поля заголовка IP могут изменяться на пути передачи и значения этих полей при получении пакета могут быть непредсказуемыми для отправителя. Такие поля нельзя защитить с помощью AH. Таким образом, защита заголовка IP с помощью AH является неполной (см. Приложение A.).

AH может использоваться в комбинации с ESP [Ken-ESP] или путем вложения [Ken-Arch]. Услуги по защите могут обеспечиваться между парой взаимодействующих хостов, парой защитных шлюзов, а также между защитным шлюзом и хостом. ESP может использоваться для обеспечения такой же защиты от повторного использования пакетов и аналогичной защиты целостности, а также обеспечивает дополнительную защиту конфиденциальности (шифрование). Основным различием между защитой целостности, обеспечиваемой ESP и AH является расширение покрытия. В частности, ESP не защищает никаких полей заголовка IP, пока эти поля не инкапсулируются ESP (например, за счет использования туннеля). Более детальное описание использования AH и ESP в разных сетевых средах приводится в документе, посвященном архитектуре защиты [Ken-Arch].

В разделе 7 кратко рассмотрены отличия этого документа от RFC 2402 [RFC2402].

# 2. Формат заголовка аутентификации

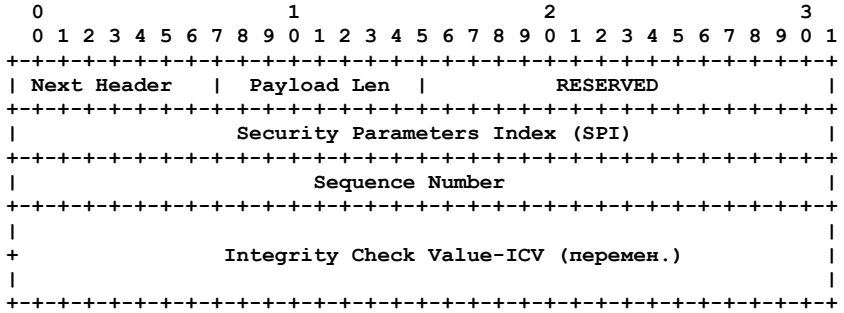



*Рисунок 1. Формат заголовка AH.*

В протокольный заголовок (IPv4, IPv6 или расширение IPv6), непосредственно предшествующий заголовку AH, **следует** помещать значение 51 в поле Protocol (IPv4) или Next Header (IPv6, включая расширение) [DH98]. Рисунок 1 показывает формат заголовка AH.

В приведенной ниже таблице приведены поля заголовка AH, показанные на рисунке, и другие поля, используемые при проверке целостности, а также указано, какие поля покрываются ICV и что передается.

| | Число битов | Когда требуется[3] | Покрывается | Передается |
|---|---|---|---|---|
| IP Header | переменное | M | [4] | да |
| Next Header | 1 | M | да | да |

[1] Encapsulating Security Payload - инкапсуляция защищенных данных.
[2] Security Association.
[3] M = mandatory - обязательно.
[4] Информация о покрываемых полях заголовка IP приведена в параграфе 3.3.3.

| | | | | |
|---|---|---|---|---|
| Payload Len | 1 | M | да | да |
| RESERVED | 2 | M | да | да |
| SPI | 4 | M | да | да |
| Seq# (младшие 32 бита) | 4 | M | да | да |
| ICV | переменное | M | да[1] | да |
| IP datagram[2] | переменное | M | да | да |
| Seq# (старшие 32 бита) | 4 | При поддержке ESN[3] | да | нет |
| ICV Padding | переменное | Если нужно | | нет |

Описание полей заголовка AH приводится в последующих параграфах. Все описанные здесь поля являются обязательными, т. е., всегда должны присутствовать в заголовке AH и учитываются при расчете значения для контроля целостности (ICV[4]) (см. параграфы 2.6 и 3.3.3).

Примечание. Предполагается, что все криптографические алгоритмы IPsec используют на входе канонический сетевой порядок байтов (см. Приложение к RFC 791 [RFC791]) и на выходе дают результат также в каноническом сетевом порядке байтов.

При передаче пакетов IP также используется сетевой порядок байтов.

AH не содержит номера версии, следовательно для обеспечения совместимости со старыми версиями информация о версии аутентификационного заголовка **должна** передаваться с помощью механизмов сигнализации между партнерами IPsec (например, IKE [IKEv2] или настройка по отдельному каналу).

## 2.1. Next Header - следующий заголовок

Восьмибитовое поле Next Header показывает тип информации, расположенной после аутентификационного заголовка AH. Значения этого поля выбираются из списка номеров протоколов IP[5], представленного на сайте IANA[6]. Например, значение 4 показывает протокол IPv4, значение 41 - IPv6, а 6 - протокол TCP.

## 2.2. Payload Length - размер данных

Это 8-битовое поле указывает размер заголовка AH в 32-битовых словах (4-байтовых блоках) минус 2. Например, если алгоритм проверки целостности использует 96-битовое аутентификационное значение, поле длины будет иметь значение 4 (3 слова полей фиксированной длины и 3 слова для ICV минус 2). Для IPv6 общий размер заголовка должен быть кратным 8 октетам (отметим, что хотя IPv6 [DH98] характеризует AH как внешний заголовок, его размер измеряется в 32-битовых словах, а не 64-битовых, как в заголовках расширения IPv6). Комментарии по учету байтов заполнения приведены в параграфах 2.6 и 3.3.3.2.1.

## 2.3. Reserved - резерв

Это 16-битовое поле зарезервировано для использования в будущем. Отправитель **должен** устанавливать для этого поля нулевое значение, а получателю **следует** игнорировать значение этого поля. Отметим, что значение этого поля (0) учитывается при вычислении ICV, но игнорируется получателем.

## 2.4. Security Parameters Index (SPI) - список параметров защиты

SPI представляет собой произвольное 32-битовое значение, используемое получателем для аутентификации SA, с которой связан входящий пакет. Для индивидуальных SA, значение SPI может само по себе аутентифицировать SA или использоваться в комбинации с типом протокола IPsec (в данном случае AH). Поскольку для индивидуальных SA значение SPI генерируется получателем, решение вопроса о достаточности этого значения для аутентификации SA или необходимости использования в комбинации с типом протокола IPsec определяется локальными условиями. Поле SPI является обязательным и упомянутый выше механизм отображения входящего трафика на индивидуальные SA **должен** поддерживаться всеми реализациями AH.

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 защищенных 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 защищенных связей (SAD[7]) [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]Обнуляется перед расчетом ICV и результат расчета помещается в это поле.
[2]В туннельном режиме - дейтаграмма IP, в транспортном - следующий заголовок и данные.
[3]Extended Sequence Number - расширенный порядковый номер.
[4]Integrity Check Value - значение проверки целостности.
[5]Реестр IP Protocol Numbers.
[6]В настоящее время этот список доступен по ссылке http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml. *Прим. перев.*
[7]Security Association Database.

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[1].

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[2] [RFC3547]). Обычно группы SSM[3] [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[4] в период, когда реализация IPsec запрашивает новую SA для объекта управления ключами, но данная SA еще не организована.

## 2.5. Sequence Number - порядковый номер

Это 32-битовое поле, трактуемое, как целое число без знака, содержит значение счетчика пакетов, которое увеличивается на 1 для каждого переданного пакета (счетчик 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. AH не предоставляет возможностей синхронизации порядковых номеров между множеством отправителей или осмысленного счетчика пакетов на стороне получателя и не обеспечивает окна в контексте множества отправителей. Таким образом, для SA с множеством отправителей функции предотвращения повторного использования пакетов AH 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 Number осуществляется по усмотрению получателя, но все реализации AH **должны** обеспечивать возможность обработки, описанной в параграфах 3.3.2. 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.

Счетчики 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 3.3.2). Если предотвращение повторного использования пакетов включено (используется по умолчанию), передаваемые порядковые номера никогда не должны повторяться. Таким образом, счетчики пакетов на стороне отправителя и получателя **должны** сбрасываться (путем создания новой SA и нового ключа) до передачи пакета с порядковым номером 2^32 в каждой SA.

### *2.5.1. 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[5] **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 B: Расширенные порядковые номера (64 бита) ). В заголовке AH каждого пакета 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, но не передаются.

## 2.6. Integrity Check Value (ICV) - контроль целостности

Это поле переменной длины содержит значение контрольной суммы ICV[6] для данного пакета. Размер поля должен быть кратным 32 битам как для IPv4, так и для IPv6. Обработка ICV рассматривается в параграфах 3.3.3. Расчет ICV и 3.4.4. Проверка ICV. Это поле может включать явное заполнение для обеспечения кратности размера заголовка AH в целом 32 (IPv4) или 64 (IPv6) битам. Заполнение **должны** поддерживать все реализации и размер заполнения **должен** быть минимально достаточным для выравнивания заголовков в соответствии с требованиями IPv4/IPv6. Подробное описание расчета размера области заполнения приведено в параграфе 3.3.3.2. Заполнение и расширенные

[1]Ternary Content-Addressable Memory - ассоциативная память.
[2]Group Domain of Interpretation.
[3]Source-Specific Multicast.
[4]No Security Association Exists.
[5]Extended Sequence Number - расширенный порядковый номер.
[6]Integrity Check Value - значение для проверки целостности.

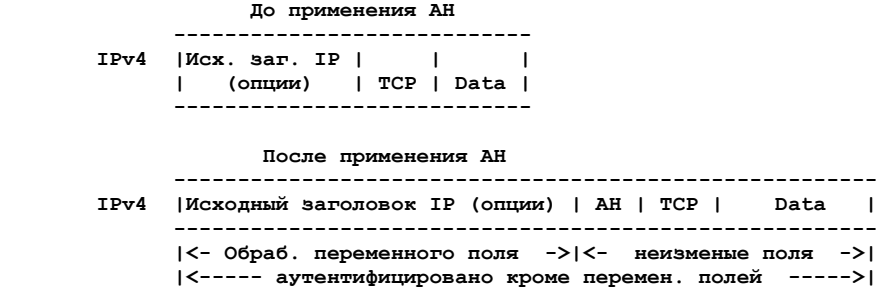
порядковые номера. Спецификация алгоритма контроля целостности **должна** включать размер ICV, а также правила сравнения и этапы обработки при проверке целостности.

# *3. Обработка аутентификационного заголовка AH*

## 3.1. Местоположение AH

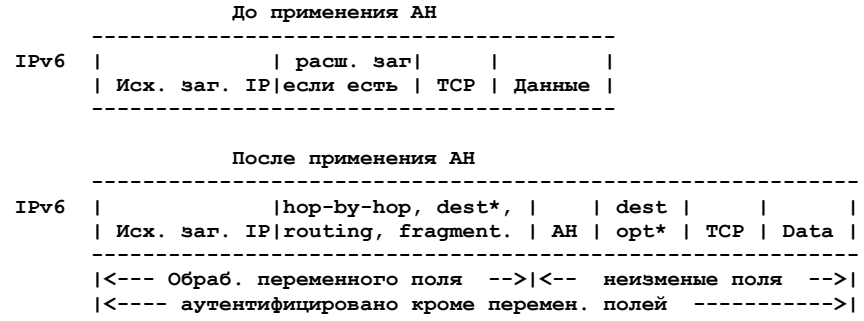
AH может работать в двух режимах - транспортном и туннельном. Более подробное описание этих режимов и рекомендации по выбору приводится в документе, посвященном архитектуре защиты.

### *3.1.1. Транспортный режим*




В транспортном режиме[1] AH помещается между заголовком IP и заголовком протокола следующего уровня (например, TCP, UDP, ICMP и т. п.) или перед другими заголовками IPsec, если они имеются. В контексте IPv4 это говорит о размещении AH после заголовка IP (и всех опций этого заголовка), но перед заголовком протокола следующего уровня. На рисунке справа показан заголовок типичного пакета IPv4 до защиты и положение заголовка AH при работе в транспортном режиме.

В контексте IPv6 заголовок AH представляется, как передаваемые «насквозь» данные и, следовательно, ему надлежит размещаться после заголовков расширения (hop-by-hop, routing, fragmentation). Опции получателя в расширенном заголовке могут располагаться перед заголовком AH, после него или по обе стороны, в зависимости от желаемой семантики. На приведенном ниже рисунке показано размещение AH для транспортного режима в типичном пакете IPv6.

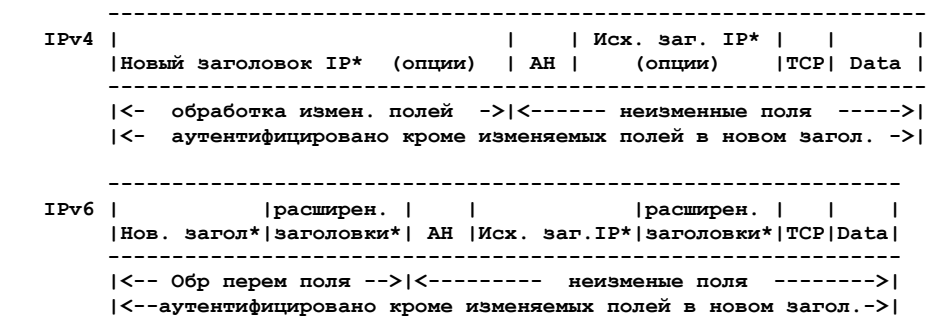



* при наличии может располагаться перед AH, после AH или в обоих местах

ESP и AH могут использоваться совместно в разных режимах. В документе, описывающем архитектуру IPsec, кратко рассматриваются методы настройки защищенных связей при использовании обоих протоколов.

Отметим, что в транспортном режиме для реализаций bump-in-the-stack и bump-in-the-wire, как определено в архитектуре защиты, входящие и исходящие фрагменты IP могут потребовать от реализации IPsec выполнения дополнительных операций фрагментации/сборки дейтаграмм IP для выполнения требования данной спецификации и обеспечения прозрачной поддержки IPsec. Особое внимание следует обращать на выполнение таких операций при использовании множества интерфейсов.

### *3.1.2. Туннельный режим*




* при использовании создается внешний заголовок IP и меняется внутренний, как описано в документе, посвященном архитектуре защиты. Расширенные заголовки могут отсутствовать.

[1]Отметим, что термин транспортный не следует трактовать как на ограничесние использования протоколов только транспортными протоколами TCP и UDP.

В туннельном режиме «внутренний» заголовок IP содержит исходные адреса получателя и отправителя, а «внешний» заголовок IP - адреса «партнеров IPsec (например, защитных шлюзов). Во внутреннем и внешнем заголовках могут использоваться разные версии IP (например, IPv6 через туннель IPv4 или IPv4 через туннель IPv6). В туннельном режиме заголовок AH защищает внутренний пакет целиком (включая его заголовок). Положение AH в туннельном режиме относительно внешнего заголовка IP совпадает с положением AH в транспортном режиме. На рисунке справа показано размещение AH в типичных пакетах IPv4 и IPv6 для туннельного режима.

## 3.2. Контроль целостности

Алгоритм контроля целостности, используемый для расчета ICV, задается SA. Для связи «точка-точка» к числу подходящих алгоритмов контроля целостности относится MAC[1] с использованием симметричных алгоритмов шифрования (например, AES [AES]) или необратимые хэш-функции типа MD5, SHA-1, SHA-256 и т. п.). Для групповых приложений разработан большой набор криптографических стратегий и продолжаются исследования в этой области.

## 3.3. Обработка исходящих пакетов

В транспортном режиме отправитель помещает заголовок AH после заголовка IP и перед заголовком протокола следующего уровня, как описано выше. В туннельном режиме внешний и внутренний заголовок IP и расширения могут взаимодействовать различными способами. Создание внешнего заголовка IP в процессе инкапсуляции описан в документе, посвященном архитектуре защиты.

### *3.3.1. Нахождение SA*

AH применяется к исходящим пакетам только после того, как реализация IPsec определит, что пакет связан с SA, которая вызывается для обработки AH. Процесс определения необходимости обработки IPsec для исходящего трафика описан в документе по архитектуре защиты.

### *3.3.2. Генерация порядковых номеров*

Счетчик отправителя инициализуируется нулевым значением при организации SA. Отправитель инкрементирует счетчик 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 отсчета) счетчика 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 еще 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 AH говорит отправителю о необходимости организации новой SA, когда значение Sequence Number (или ESN) достигает максимума и должно вернуться к нулю.

Если функция предотвращения повторов отключена (как описано выше) отправителю не нужно заботиться о мониторинге переполнения (сброса в 0) счетчика порядковых номеров (например, в случае управления ключами вручную, как описано в разделе 5). Однако отправитель будет по-прежнему инкрементировать значение счетчика и после максимального значения счетчик 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 B), 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 в контрольной сумме ICV.

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 расчете ICV), что вступает в противоречие с отказом от предотвращения повторов для SA.

### *3.3.3. Расчет ICV*

При расчете ICV в AH учитываются следующие поля:

- Поля заголовка IP или расширения перед заголовком AH, которые не изменяются при передаче или предсказуемы на момент прибытия в конечную точку SA;
- заголовок AH (поля Next Header, Payload Len, Reserved, SPI, Sequence Number - младшие 32 бита), поле ICV, значение которого на момент расчета принимается нулевым, и явные байты заполнения (если они есть);
- все данные после заголовка AH предполагаются неизменными при передаче и учитываются в расчете;
- старшие биты ESN (если расширенная нумерация используется) и все байты неявного заполнения, требуемые алгоритмом контроля целостности.

#### 3.3.3.1. Обработка изменяемых полей

Если поле меняется в процессе передачи, при расчете ICV значение этого поля принимается нулевым. Если поле изменчиво, но его значение на уровне получателя IPsec предсказуемо, это значение может использоваться при расчете ICV. Значение поля Integrity Check Value при расчете также принимается нулевым. Отметим, что

[1]Message Authentication Code - код аутентфикации сообщения.

использование нулевого значения поля вместо пропуска этого поля при расчете сохраняет выравнивание для расчета ICV. Кроме того, заполнение неучитываемых полей нулями предотвращает их изменение в процессе передачи, хотя содержимое этих полей и не покрывается явно ICV.

При создании нового расширения заголовка или опции IPv4 они определяются в соответствующем RFC и в сецификацию (раздел «Вопросы безопасности») **следует** включать инструкции по учету полей при расчете ICV для AH. Если реализация IP (v4 или v6) встречается с нераспознанным расширением заголовка, она должна отбросить такой пакет и передать отправителю сообщение ICMP. IPsec просто не увидит такого пакета. Если реализация IPsec сталкивается с нераспознанной опцией IPv4, эту опцию следует целиком обнулить, используя второй байт опции в качестве ее размера. Опции IPv6 (в заголовках Destination Extension или заголовке Hop-by-Hop Extension) содержат флаг изменчивости, который определяет порядок обработки такой опции.

### 3.3.3.1.1. Расчет ICV для IPv4

#### 3.3.3.1.1.1. Поля основного заголовка

Базовые поля заголовка IPv4 характеризуются следующим образом:

***Неизменные***

Version

Internet Header Length

Total Length

Identification

Protocol (здесь должно быть значение для AH)

Source Address

Destination Address (без строгой или нестрогой маршрутизации, заданной отправителем)

***Предсказуемо изменяемые***

Destination Address (со строгой или нестрогой маршрутизацией, заданной отправителем)

***Изменяемые (0 перед расчетом ICV)***

Differentiated Services Code Point (DSCP) (6 битов, см. RFC 2474 [NBBB98])

Explicit Congestion Notification (ECN) (2 бита, см. RFC 3168 [RFB01])

Flags

Fragment Offset

Time to Live (TTL)

Header Checksum

DSCP - маршрутизаторы могут менять значение поля DS для обеспечения желаемого сервиса (локального или сквозного), поэтому значение поля в момент получения пакета отправителю предсказать невозможно.

ECN - это поле будет меняться, если на пути встретится перегруженный маршрутизатор, следовательно значение поля в момент получения пакета не может быть определено отправителем.

Flags - это поле исключается из расчетов, поскольку промежуточные маршрутизаторы могут устанавливать флаг DF, даже в тех случаях, когда отправитель не установил его.

Fragment Offset - поскольку AH применяется только к нефрагментированным пакетам IP, значение этого поля всегда должно быть нулевым, поэтому оно исключено из расчета (несмотря на предсказуемость).

TTL - это значение маршрутизаторы меняют в процессе штатной обработки пакетов, поэтому отправитель не может предсказать значение поля в момент приема пакета.

Header Checksum - это поле меняется при изменении любого флага, поэтому отправитель не может предсказать значение поля на момент доставки пакета.

#### 3.3.3.1.1.2. Опции

Для IPv4 (в отличие от IPv6) не существует механизма маркировки опций, изменяющихся при передаче. По этой причине опции IPv4 перечислены в Приложении A и явно классифицированы, как неизменные, изменяющиеся предсказуемо и переменчивые. Для IPv4 опция рассматривается как неделимый объект, поэтому, несмотря на неизменность типа и размера некоторых опций при передаче, изменение значения опции делает все поле данной опции изменяемым и опция целиком учитывается как нулевые значения при расчете ICV.

### 3.3.3.1.2. Расчет ICV для IPv6

#### 3.3.3.1.2.1. Поля основного заголовка

Поля заголовков IPv6 классифицируются следующим образом:

***Неизменные***

Version

Payload Length

Next Header

Source Address

Destination Address (без заголовка Routing Extension)

***Предсказуемо изменяемые***

Destination Address (с заголовком Routing Extension)

***Изменяемые (0 перед расчетом ICV)***

DSCP (6 битов, см. RFC2474 [NBBB98])

ECN (2 бита, см. RFC3168 [RFB01])

Flow Label[1]

Hop Limit

#### 3.3.3.1.2.2. Расширенные заголовки с опциями

Опции IPv6 в расширенных заголовках Hop-by-Hop и Destination содержат бит, указывающий, что опция может измениться (непредсказуемо) в процессе передачи. Для любой опции, которая может измениться на маршруте, все поле Option Data должно трактоваться как нулевые октеты при вычислении и проверке ICV. Поля Option Type и Opt Data Len включаются в расчет ICV. Все опции, для которых упомянутый бит показывает неизменность, включаются в расчет ICV. Дополнительную информацию о заголовках IPv6 можно найти в спецификации протокола [DH98].

#### 3.3.3.1.2.3. Расширенные заголовки без опций

Расширенные заголовки IPv6, не содержащие опций, явно перечислены в Приложении A и классифицированы, как неизменные, изменяемые предсказуемо или изменяемые.

### 3.3.3.2. Заполнение и расширенные порядковые номера

#### 3.3.3.2.1. Заполнение ICV

Как указано в параграфе 2.6, поле ICV может включать явное заполнение, если это требуется для выравнивания заголовка AH по границе 32 (IPv4) или 64 (IPv6) бита. Если заполнение требуется, его размер определяют два фактора:

- размер ICV;
- версия протокола IP (v4 или v6)

Например, если выбранный алгоритм дает 96 контрольную сумму, заполнения не требуется для обеих версий IP. Однако, если другой алгоритм расчета ICV дает сумму иного размера, может потребоваться заполнение в зависимости от размера результата и версии протокола IP. Содержимое поля заполнения выбирается по усмотрению отправителя (заполнение произвольно, но использования случайных значений в целях защиты не требуется). Эти байты заполнения включаются в расчет ICV, учитываются, как часть Payload Length, и передаются в конце поля ICV, чтобы позволить получателю повторно выполнить расчет ICV для проверки. Заполнение, превышающее по размеру количество, требуемое для выравнивания заголовков Ipv4/IPv6, не допускается.

#### 3.3.3.2.2. Неявное заполнение и ESN

Если для SA выбрано использование ESN, старшие 32 бита расширенного порядкового номера должны включаться в расчет ICV. При расчете эти биты (неявно) добавляются непосредственно после данных и перед любым неявным заполнением в пакете.

Для некоторых алгоритмов контроля целостности строка байтов, по отношению к которой выполняются операции расчета ICV, должна иметь размер, кратный заданному алгоритмом размеру блока. Если размера пакета IP (включая AH и 32 старших бита ESN при использовании расширенной нумерации) не соответствует этим требованиям, в конец пакета перед расчетом ICV **должны** добавляться байты неявного заполнения. Октеты заполнения **должны** иметь нулевое значение. Для решения вопроса об использовании такого неявного заполнения **необходимо** обратиться к документу, определяющему алгоритм контроля целостности. Если в документе нет ответа на данный вопрос, по умолчанию предполагается необходимость использования неявного заполнения (для выравнивания размера пакета в соответствии с размером блока, требуемого алгоритмом). Если байты заполнения требуются, но алгоритм не задает значения этих байтов, должны использоваться нулевые значения байтов заполнения.

## *3.3.4. Фрагментация*

Если требуется фрагментация IP, она выполняется после обработки AH в реализации IPsec. Таким образом, в транспортном режиме AH применяется только к целым дейтаграммам IP[2] (не фрагментам). Пакет IPv4, к которому применили AH, может быть фрагментирован маршрутизаторами на пути и в таком случае фрагменты должны быть собраны до обработки AH на приемной стороне (этого не возникает для IPv6, где фрагментация по инициативе маршрутизаторов невозможна). В туннельном режиме AH применяется к пакетам IP, содержимое которых может

[1]Метки потока, описанные в Ahv1, были изменяемыми и в RFC 2460 [DH98] сохранили потенциальную изменчивость. Для совместимости с существующими реализациями AH метки потоков не вкалючаются в расчет ICV для AHv2.

[2]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.

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 AH для таких фрагментов в туннельном режиме.

Фрагментация, выполняемая реализацией IPsec или маршрутаторами на пути доставки между партнерами IPsec, существенно снижает производительность. Более того, необходимость сборки фрагментов на приемной стороне до выполнения операций AH, порождает возможность организации атак на отказ служб. Таким образом, реализация AH **может** выбрать отказ от поддержки фрагментации и маркировать передаваемые пакеты флагом DF[2] для облегчения определения PMTU[3]. В любом случае, реализация AH **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*

При наличии нескольких заголовков/расширений IPsec при обработке каждого из них остальные игнорируются (не обнуляются и не используются).

### *3.4.1. Сборка фрагментов*

Если нужна сборка фрагментов[4], 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 аутентификационный заголовок IP (AH), получатель определяет подходящую (одностороннюю) SA путем просмотра SAD. Для индивидуальных SA определение основано на значении SPI, в дополнение к которому может использоваться поле протокола, как описано в параграфе 2.4. 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 2.4 (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 задает алгоритм(ы), используемый для расчета ICV и показывает, нужно ли проверять значения ICV.

Если для пакета не найдено защищенной 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 AH **должны** поддерживать предотвращение повторного использования пакетов[5], хотя использование этой функции может быть включено или отключено получателем на уровне SA. Функции предотвращения повторного использования применимы как к индивидуальным, так и к групповым SA. Однако данный стандарт не задает 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 путем 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 3.3.2. Генерация порядковых номеров), получателю **следует** уведомить отправителя об отказе от поддержки предотвращения повторного использования на этапе организации SA с помощью протокола организации SA (например, IKE).

Если получатель включил предотвращение повторного использования для SA, он **должен** установить значение счетчика 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 AH по отношению к данному пакету сразу после проверки принадлежности пакета к SA для ускорения отбрасывания дубликатов.

Дубликаты отбрасываются с помощью «скользящего» окна приема. 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 счетчика) 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 интервал приема для отдельной SA до $2^{32}$-1 пакетов. Если

[2]Не фрагментировать. *Прим. перев.*
[3]Path MTU - размер максимального передаваемого блока для пути.
[4]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.
[5]Anti-replay service.

интервал становится больше, **могут** использоваться эвристические проверки для ресинхронизации порядковых номеров на приемной стороне, как описано в Приложении B.

Если полученный пакет попадает в окно и не является дубликатом, получатель выполняет проверку ICV. При отрицательном результате проверки ICV 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 приема обновляется только при положительном результате проверки ICV.

**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 высоскоскоростных сред размер окна приема следует увеличивать. Минимальные и рекомендуемые размеры окна для высокоскоростных (например, мультимегабитных) устройств данный стандарт не задает.

### *3.4.4. Проверка ICV*

Получатель рассчитывает ICV для соответствующих полей пакета, используя заданный алгоритм контроля целостности и сравнивает полученный результат со значением поля ICV в принятом пакете. Детальное описание расчета приведено ниже.

Если рассчитанное значение ICV совпадает с полученным в пакете, дейтаграмма считается корректной. Если значения не совпадают, получатель **должен** отбросить полученную дейтаграмму IP, как некорректную с записью информации об этом факте в журнал аудита. В запись **следует** включать значение SPI, дату и время, адреса отправителя и получателя, а также Flow ID для IPv6.

Примечание для разработчиков.

Разработчики могут использовать любую последовательность действий, которая дает такой же результат, как перечисленные здесь операции.

Сначала значение ICV из принятого пакета сохраняется и заменяется нулем. Значения поля заполнения ICV при этом не изменяются. Далее обнуляются все остальные поля, которые могли измениться в процессе передачи пакета (поля, которые следует обнулять при расчете ICV, перечислены в параграфе 3.3.3.1. Обработка изменяемых полей). Если для данной SA выбрано использование ESN, добавляются старшие 32 бита ESN в конце пакета. Проверяется общий размер пакета (как описано выше) и при необходимости в конец пакета (после старших битов ESN, если они используются) добавляются нулевые байты неявного заполнения с учетом требований алгоритма контроля целостности.

Здесь описан лишь общий ход проверки, который на практике может быть существенно другим. Например, при использовании цифровой подписи или однонаправленной хэш-функции для расчета ICV соответствующий процесс может быть более сложным.

## *4. Аудит*

Не все системы, поддерживающие AH, реализуют аудит. Однако, если AH встраивается в систему, поддерживающую аудит, реализация AH **должна** поддерживать аудит и также **должна** позволять администратору системы включать и отключать аудит для AH. 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. 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 своем соответствии или совместимости с данной спецификацией, **должны** полностью реализовать синтаксис и обработку AH, 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 расчета ICV, корректная работа системы предотвращения повторного использования пакетов требует аккуратной поддержки состояния счетчика на передающей стороне при замене ключа, поскольку в этом случае невозможно восстановить работу после переполнения счетчика. Таким образом, совместимым со спецификацией реализациям **не следует** предоставлять такой сервис для SA с распространением ключей вручную.

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

## *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 2402*

В этом документе имеется ряд перечисленных ниже отличий от RFC 2402 [RFC2402].

- 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 защищенных связей с множеством отправителей.
- Спецификации обязательных алгоритмов вынесены в отдельный документ [Eas04].

# *8. Благодарности*

Автор выражает свою благодарность Ran Atkinson, за его критически важную роль на начальном этапе создания IPsec и всем авторам первой серии стандартов IPsec - RFC 1825-1827. Отдельная благодарность Karen Seo за помощь в редактировании этой и предыдущей версии данной спецификации. Автор также благодарит членов рабочих групп IPsec и MSEC, которые внесли свой вклад в разработку спецификации протокола.

# *9. Литература*

## 9.1. 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, D., "Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 4305, December 2005.

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

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

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

## 9.2. Дополнительная литература

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

[HC03] Holbrook, H. and B. Cain, "Source Specific Multicast for IP", Work in Progress[1], November 3, 2002.

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

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

[NBBB98] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

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

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

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

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

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

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

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

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

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

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

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

# *Приложение A: Изменяемые опции и расширения заголовков IP*

## A1. Опции IPv4

В таблице показаны опции IPv4 с классификацией их по «изменяемости». Если в колонке «Документ» указаны две ссылки, второй документ имеет более высокий приоритет. Таблица основана на информации, представленной в RFC 1700, "ASSIGNED NUMBERS"[2], (октябрь 1994).

| Копия | Класс | Номер опции | Имя | Документ |
|---|---|---|---|---|
| ***Неизменные*** - учитываются в ICV | | | | |
| 0 | 0 | 0 | End of Options List[3] – конец списка опций | [RFC791] |

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

[2]В настоящее время документ «Assigned Numbers» утратил силу. Реестры выделенных значений публикуются на сайте http://www.iana.org/numbers/. *Прим. перев.*

[3]Опции End of Options List **следует** повторять при необходимости для выравнивания заголовка IP по 4-байтовой границе, чтобы предотвратить появление байтов, которые могли бы использоваться для организации скрытых каналов.

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | No Operation - нет операции | [RFC791] |
| 1 | 0 | 2 | Security[3] - защита | [RFC1108][4] |
| 1 | 0 | 5 | Extended Security[3] - расширенная защита | [RFC1108][4] |
| 1 | 0 | 6 | Commercial Security[3] - коммерческая защита | [RFC2113] |
| 1 | 0 | 20 | Router Alert - сигнал маршрутизатора | [RFC1770] |
| 1 | 0 | 21 | Sender Directed Multi-Destination Delivery - направленная отправителем доставка по множеству адресов | |
| ***Изменяемые*** - должны иметь нулевое значение | | | | |
| 1 | 0 | 3 | Loose Source Route - нежестко заданный отправителем маршрут | [RFC791] |
| 0 | 2 | 4 | Time Stamp - временная метка | [RFC791] |
| 0 | 0 | 7 | Record Route[5] - запись маршрута | [RFC791] |
| 1 | 0 | 9 | Strict Source Route - жестко заданный отправителем маршрут | [RFC791] |
| 0 | 2 | 18 | Traceroute - трассировка пути | [RFC1393] |
| ***Экспериментальные и переопределенные*** - должны иметь нулевое значение | | | | |
| 1 | 0 | 8 | Stream ID - идентификатор потока | [RFC791], [RFC1122] |
| 0 | 0 | 11 | MTU Probe – проба MTU | [RFC1063], [RFC1191] |
| 0 | 0 | 12 | MTU Reply – отклик MTU | [RFC1063], [RFC1191] |
| 1 | 0 | 17 | Extended Internet Protocol - расширенный протокол IP | [RFC1385], [DH98] |
| 0 | 0 | 10 | Experimental Measurement – экспериментальные измерения | |
| 1 | 2 | 13 | Experimental Flow Control - экспериментальное управление потоком данных | |
| 1 | 0 | 14 | Experimental Access Ctl - экспериментальный контроль доступа | |
| 0 | 0 | 15 | ??? | |
| 1 | 0 | 16 | IMI Traffic Descriptor - дескриптор трафика IMI | |
| 1 | 0 | 19 | Address Extension - расширение адреса | |

## A2. Заголовки расширения IPv6

В таблице показаны расширения заголовков IPv6 и дана их классификация в плане «изменчивости».

| Название опции или расширения | Ссылка |
|---|---|
| ***Предсказуемо изменяемые*** – включаются в расчет ICV | |
| Routing (Type 0) | [DH98] |
| ***Биты, показывающие изменяется ли опция*** (изменения при передаче непредсказуемы) | |
| Опция Hop-by-Hop | [DH98] |
| Опция Destination | [DH98] |
| ***Неприменимо*** | |
| Fragmentation | [DH98] |

Опции IPv6 в расширенных заголовках Hop-by-Hop и Destination содержат бит, который показывает возможность (непредсказуемого) изменения опции в процессе доставки пакета. Для любой опции, содержимое которое может меняться на пути, все поле Option Data должно трактоваться, как нулевые октеты при расчете и проверке ICV. Поля Option Type и Opt Data Len включаются в расчет ICV. Все опции, для которых флаг показывает неизменность, включаются в расчет ICV. Дополнительная информация по опциям IPv6 приведена в [DH98].

Routing (Type 0) - этот заголовок IPv6 будет приводить к реорганизации адресных полей в пакете на пути доставки. Однако содержимое пакета на момент доставки известно отправителю и всем промежуточным узлам. Следовательно, это поле включается в расчет ICV, поскольку его изменения предсказуемы. Отправитель должен перед расчетом ICV включить в это поле то значение, которое увидит получатель.

---

[3]Добавление или удаление защитных меток (например, Basic Security Option - BSO, Extended Security Option - ESO или Commercial Internet Protocol Security Option -CIPSO) системами на пути доставки пакета вступает в конфликт с данной классификацией, считающей эти опции IP неизменными, и, следовательно, несовместимо с использованием IPsec.

[4]Устарел, но продолжает использоваться.

[5]Использование опции Router Alert потенциально несовместимо с IPsec. Хотя эта опция является неизменной, ее использование ведет к тому, что все маршрутизаторы на пути будут «обрабатывать» пакет и, следовательно, могут менять его. Это может происходить поэтапно на пути от одного маршрутизатора к другому. До обработки приложениями, которым эта опция адресуется (например, протокол RSVP или IGMP) пакет следует подвергнуть обработке AH. Однако эта обработка требует, чтобы каждый маршрутизатор на пути был членом групповой SA, определяемой SPI. Это может вызывать проблемы с нестрого заданной отправителем маршрутизацией и требует методов поддержки группового трафика, которые в настоящее время недоступны.

Fragmentation - фрагментация происходит после выходной обработки IPsec (параграф 3.3), а сборка - перед входной обработкой IPsec (параграф 3.4). По этой причине расширенный заголовок Fragmentation (если он имеется) не виден для IPsec.

Отметим, что на приемной стороне реализация IP может сохранить расширенный заголовок Fragmentation после сборки фрагментов. В таком случае реализация AH при получении пакета **должна «**удалить» (или пропустить) этот заголовок до обработки ICV и заменить значение предыдущего поля Next Header на значение Next Header из заголовка Fragmentation.

Отметим, что на стороне отправителя реализация IP может передавать IPsec пакет с расширенным заголовком Fragmentation, где Offset = 0 (первый фрагмент) и флаг More Fragments = 0 (последний фрагмент). В таких случаях до обработки ICV реализация AH **должна** «удалить» (или пропустить) этот заголовок и заменить значение предыдущего Next Header на значение Next Header из заголовка Fragmentation.

# *Приложение B: Расширенные порядковые номера (64 бита)*

## B1. 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 повтрно используемых пакетов, а также механизм определения старших битов порядкового номера, используемых для отбрасывания пакетов и расчета ICV. Описан также механизм обработки случаев потери синхронизации для старших (не передаваемых в пакетах) битов порядкового номера.

## B2. Окно Anti-Replay

Получатель будет поддерживать окно предотвращения повторного использования пакетов размером W[1]. 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 приведенной 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 |
| 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.

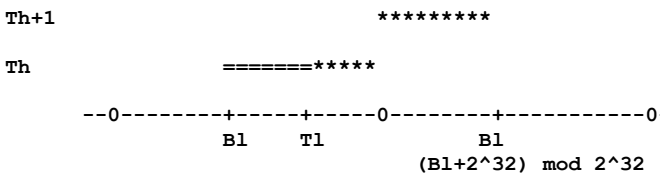



*Рисунок 2. Случай A.*

- Случай A: Tl >= (W - 1) все окно находится в одном подпространстве порядковых номеров (Рисунок 2)
- Случай B: Tl < (W - 1) окно захватывает части двух смежных подпространств порядковых номеров (Рисунок 3).

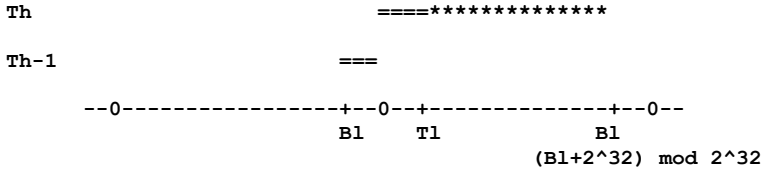



*Рисунок 3. Случай B.*

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. Звездочки **** обозначают грядущие номера, т. е., номера номера, превышающие максимальный аутентифицируемый в данный момент номер (ThTl).

[1]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.

## B2.1. 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 приема 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 нгомера.

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 B2.2.
- Случай B: Если Seql >= Bl ( где Bl = Tl - W + 1) **ИЛИ** Seql <= Tl, проверяется соответствую щий 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 B2.2.

## B2.2. Определение старших битов (Seqh) порядкового номера

```
+ Для случая A (Рисунок 2):
  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 3):
  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 передается только значение Seql, получатель должен отслеживать подпространство порядковых номеров для каждого пакета (т. е., определять значение Seqh). Приведенные справа уравнения определяют выбор Seqh в «нормальных» условиях. В параграфе B3 рассматривается определение старших битов номера в условиях экстремальных потерь пакетов.

## B2.3. Пример псевдокода

Приведенный 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
```

## B3. 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 приемной стороне является предпочтительной; кроме того, такое решение обеспечивает совместимость с ранними версиями.

### *B3.1. 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. Прием корректного пакета для данной SA ведет к сбросу счетчика 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.

### *B3.2. Процесс ресинхронизации*

Когда значение счетчика 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 задается локальным параметром. Поскольку значение Seqh неявно помещается после данных AH (или ESP), может оказаться возможной оптимизация процедуры восстановления за счет выполнения процедуры контроля целостности пакета с использованием нарастающих значений Seqh для расчета ICV. При успешной аутентификации пакета с помощью описанной процедуры значение счетчика некорректных пакетов сбрасывается и устанавливается значение T, определенное по прошедшему проверку пакету.

Это решение требуется поддерживать только на приемной 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 создает дополнительной возможности организации атак на службы путем 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**

IETF не принимает какой-либо позиции в отношении действительности или объема каких-либо прав интеллектуальной собственности (Intellectual Property Rights или IPR) или иных прав, которые, как может быть заявлено, относятся к реализации или использованию описанной в этом документе технологии, или степени, в которой любая лицензия, по которой права могут или не могут быть доступны, не заявляется также применение каких-либо усилий для определения таких прав. Сведения о процедурах IETF в отношении прав в документах RFC можно найти в BCP 78 и BCP 79.

Копии раскрытия IPR, предоставленные секретариату IETF, и любые гарантии доступности лицензий, а также результаты попыток получить общую лицензию или право на использование таких прав собственности разработчиками или пользователями этой спецификации, можно получить из сетевого репозитория IETF IPR по ссылке http://www.ietf.org/ipr.

IETF предлагает любой заинтересованной стороне обратить внимание на авторские права, патенты или использование патентов, а также иные права собственности, которые могут потребоваться для реализации этого стандарта. Информацию следует направлять в IETF по адресу ietf-ipr@ietf.org.

**Подтверждение**

Финансирование функций RFC Editor обеспечено Internet Society.