Протоколы автоматического установления контекстов безопасности и управления ключами в Интернет[1]

К.В. Ребриков, В.З. Шнитман

"Мы не думаем, что кто-либо способен
изучить IPsec по его документации".
"…мы не уверены в том, что полностью понимаем систему,
хотя проанализировали ее досконально".
Нильс Фергюсон, Брюс Шнайер
"Оценка IPsec с точки зрения криптографии" [9]

Аннотация. В данной работе рассматриваются протоколы автоматического установления контекстов безопасности в Интернет. Приводится подробное описание работы протокола первого поколения ISAKMP/IKE. Рассмотрены его основные недостатки, а также способы их преодоления в следующей версии протокола – IKEv2.

1. Введение

Семейство протоколов безопасности Интернет (далее IPsec) обеспечивает конфиденциальность, целостность данных (на уровне пакетов), контроль доступа и аутентификацию источника данных для IP-дейтаграмм. Семейство протоколов IPsec является очень сложным, состоящим из нескольких вспомогательных протоколов. Изучение IPsec затрагивает всё многообразие проблем работы на сетевом уровне стека TCP/IP.

Для предоставления услуг безопасности между взаимодействующими узлами устанавливается соединение. Такие узлы обладают общим состоянием, называемым контекстом безопасности (Security Association), которое, в частности, определяет предоставляемые услуги, используемые криптографические алгоритмы и ключи. На каждом из узлов имеется своя копия переменных, образующих контекст безопасности. Установление этого

[1] Работа выполнена при финансовой поддержке РФФИ в рамках проекта № 04-07-90308 "Верификация функций безопасности и мобильности протоколов IP".

общего состояния вручную достаточно обременительно. В общем случае необходим протокол для автоматического установления контекстов безопасности. В семействе IPsec этой цели служит протокол обмена ключевой информацией IKE (Internet Key Exchange).

Задачей IKE является защищённое согласование параметров и предоставление аутентифицированного ключевого материала для контекстов безопасности или, другими словами, автоматическое установление контекстов безопасности между двумя взаимодействующими узлами. IKE – это громоздкий протокол сам по себе. Официальная спецификация протокола (RFC 2407, 2408, 2409 [1, 2, 3]), являясь довольно запутанной, практически не позволяет разобраться во всех деталях его работы. Разумеется, спецификация является истиной в последней инстанции, но в качестве источника, чтобы познакомиться с протоколом или понять его, она вряд ли подходит.

Авторы данной статьи ставили перед собой задачу написания максимально понятного и вместе с тем по возможности полного обзора протокола IKE, его свойств, возможностей, особенностей работы. Предполагается знакомство читателя с семейством IPSec. Для облегчения одновременной работы с англоязычной литературой (оригинальные спецификации, статьи, документация к реализациям IKE), как правило, рядом с терминами IKE приводятся их англоязычные аналоги. В приложении 1 приводится краткий словарь терминов, при переводе которых возможна неоднозначность. Краткое изложение некоторых понятий из области криптографии, необходимых для понимания работы протокола, вынесено в отдельное приложение 2.

Изложение ведется методом последовательного приближения. В разделе 2 рассмотрены вопросы взаимосвязи IKE с его предшественниками, вводятся основные понятия и принятые обозначения, а также кратко описаны режимы первой и второй фазы работы протокола. Следует отметить, что данный раздел является законченным рассмотрением протокола, несмотря на то, что в принятом приближении проигнорирован целый ряд особенностей его работы. Раздел 3 является следующим приближением – рассмотрены вопросы согласования криптографических параметров и формирования криптографического материала, а также все возможные варианты обменов. При этом информация о (весьма нетривиальном) строении пакетов IKE и их элементов вынесена в специальное приложение 3.

После публикации в 1998 году спецификаций семейства протоколов IPsec они были подвергнуты всестороннему анализу и стали повсеместно использоваться. Выявленные в процессе критического анализа недостатки способствовали лучшему пониманию их особенностей и послужили причиной продолжения исследований в этой области. Поэтому в раздел 4 данной статьи включены материалы двух авторитетных обзоров, связанных с анализом недостатков IKE, а в раздел 5 – вопросы преодоления этих недостатков в следующей версии протокола (IKEv2).

2. Обзор протокола IKE

2.1. Взаимосвязь протоколов Oakley, SKEME, ISAKMP и IKE

Протокол IKE создан на базе нескольких протоколов. Основой IKE является протокол ISAKMP, в который внедрены некоторые решения из двух ранее разработанных протоколов управления ключами: Oakley и SKEME. Кроме того, в IKE определяются два собственных типа обмена ключевой информацией.

Протокол ISAKMP был разработан в Национальном агентстве безопасности США. Спецификация ISAKMP (RFC 2408 "Internet Security Association and Key Management Protocol" [2]) определяет общую структуру протоколов, которые используются для установления контекстов безопасности и выполнения других функций управления ключами. Этот протокол, задавая каркас для создания контекстов безопасности, определяя структуру пакетов, формат и семантику полей, общие правила обработки пакетов, не конкретизирует ни конкретные криптографические величины, применяемые при установлении контекста безопасности, ни параметры обмена. Кроме того, в ISAKMP определяется избыточное (с точки зрения IKE) количество различных типов обменов и блоков данных.

Некоторые синтаксические детали работы ISAKMP, например, структура полей, имеющих отношение к конкретному протоколу, контексты безопасности которого согласует ISAKMP, определены в других документах, задающих так называемый домен интерпретации (DOI – Domain of Interpretation), и стандартах обмена ключами. Протоколы безопасности, относящиеся к одному домену интерпретации, выбирают криптографические преобразования из общего пространства имён, а также используют общие идентификаторы протокола обмена ключами. Они одинаково интерпретируют содержание данных, относящихся к этому домену интерпретации. Домен интерпретации ISAKMP для семейства протоколов IPsec определен в спецификации RFC 2407 "The Internet IP Security Domain of Interpretation for ISAKMP" [1].

Oakley (RFC 2412 "The Oakley Key Determination Protocol"[4]) – это протокол свободного формата, позволяющий каждой из сторон изменять состояние протокола со своей скоростью. Из этого протокола была заимствована идея различных режимов – последовательностей сообщений, обеспечивающих аутентифицированный обмен ключевой информацией. Протокол Oakley не определял, какой именно информацией необходимо обмениваться в каждом из сообщений. Режимы (modes) использовались в качестве примеров того, как этот протокол может быть использован для достижения безопасного обмена ключами. В спецификации IKE реализованы некоторые режимы Oakley, которые часто называются также обменами (exchanges).

Протокол SKEME (A Versatile Secure Key Exchange Mechanism for Internet [5]) описывает механизм обмена ключами, предоставляющий услуги анонимности,

возможности отказа от обязательств и быстрого обмена ключами. Этот протокол определяет защищённый обмен ключами, при котором участники для аутентификации друг друга используют шифрование с открытым ключом. Каждый из участников зашифровывает открытым ключом партнера случайное число, и оба этих случайных числа (после расшифровывания) участвуют в формировании основного ключа. Кроме того, для обновления существующего ключа в протоколе SKEME предусмотрена возможность проведения дополнительного обмена Диффи-Хеллмана для обеспечения совершенной прогрессирующей секретности (PFS – Perfect Forward Secrecy) либо использования другого быстрого обмена, который не требует операций с открытыми ключами. Протокол IKE для одного из своих методов аутентификации (аутентификации с шифрованием открытым ключом) заимствует этот метод прямо из спецификации SKEME, а также заимствует оттуда идею быстрого обновления ключа без обеспечения PFS.

В отличие от Oakley и SKEME, из которых заимствованы отдельные идеи, ISAKMP является основой протокола IKE. Более того, IKE принято рассматривать не как независимый протокол, а как расширение ISAKMP. Это не совсем верно, IKE не только конкретизирует каркас, заданный ISAKMP, но в ряде случаев несколько изменяет схемы ISAKMP. В качестве наиболее серьёзного отличия можно привести пример с фазами работы: согласно ISAKMP, в ходе и первой, и второй фаз работы применяются одни и те же обмены, а в IKE вводятся разные обмены для первой и второй фаз (подробнее об этом далее). Тем не менее, по сравнению с тем, что IKE заимствовал из ISAKMP, эти изменения пренебрежимо малы. Более того, сама спецификация IKE (RFC 2409 [3]) не является независимым документом, а основана на спецификации ISAKMP (RFC 2408 [2]).

Нередко употребляется обозначение ISAKMP/IKE (в данной статье оно будет использоваться как синоним IKE). Что именно имеется в виду под ISAKMP, IKE и ISAKMP/IKE в каком-то конкретном документе, иной раз невозможно понять без учёта контекста. Например, одну из реализаций IKE под платформу BSD, isakmpd (недвусмысленное название, isakmp daemon), авторы называют "ISAKMP/Oakley (a.k.a. IKE) implementation".

Технические детали реализации протокола (такие как механизм сцепления блоков данных – структурных элементов пакета IKE, битовая длина конкретных полей блоков и заголовков), не очень важные для понимания принципов его работы, относятся к компетенции именно ISAKMP. Поэтому традиция оригинальных спецификаций (продолженная во многих обзорах и статьях) вести изложение IKE в восходящем порядке – от ISAKMP (RFC 2408) к IKE (RFC 2409), рассматривая IKE как расширение ISAKMP, в данной работе нарушена, и в первых двух разделах статьи рассматривается именно итоговый протокол IKE, а описание форматов сообщений и блоков данных вынесено в отдельное приложение 3. Вместе с тем, протокол IKE неразрывно

связан со своей основой, протоколом ISAKMP, поэтому ссылок на ISAKMP избежать не удастся.

2.2. Основные понятия

При описании процедуры взаимодействия по протоколу ISAKMP/IKE, прежде всего, следует определить такие понятия, как сообщения, включающие заголовок и блоки данных, а также обмены или режимы и фазы. Термины «контекст безопасности» (security association), «предложение» (proposal), «преобразование» (transform), «атрибут» (attribute), «домен интерпретации» (domain of interpretation), «защитный набор» (protection suite) относятся к процедуре согласования информации об услугах безопасности и строению вовлечённых в эту процедуру блоков данных.

Сообщения (messages) – это сами пакеты IKE. Эти слова будут далее использоваться как синонимы. В спецификации RFC 2408 предполагается, что сообщения IKE могут передаваться в пакетах любого из протоколов транспортного уровня. Однако реализации IKE обязаны иметь возможность принимать и посылать сообщения IKE, используя протокол UDP. Для протокола ISAKMP/IKE регистрационным центром Internet IANA был зарезервирован порт UDP 500.

Каждое сообщение IKE состоит из **заголовка** (header) фиксированной структуры и переменного числа блоков данных, которые и составляют тело сообщения.

Блок данных (payload) – это элемент сообщения, имеющий собственную структуру, зависящую от типа блока. В общем случае блоки данных состоят из набора неделимых элементарных полей и не могут вкладываться друг в друга, однако они нередко зависят друг от друга. В IKE применяется техника формирования цепочки блоков данных, которая и связывает заголовок с блоками данных в единое сообщение. Общая картина может показаться довольно тривиальной, однако из-за сложных зависимостей между блоками данных Security Association (Контекст Безопасности), Proposal (Предложение) и Transform (Преобразование) строение сообщения имеет весьма непростой характер.

Контекст безопасности (security association) любого протокола безопасности – это набор параметров, полностью определяющих услуги и механизмы, предоставляемые этим протоколом для защиты трафика. Такими параметрами могут быть идентификаторы алгоритмов, режимы, криптографические ключи и т.д. Задачей IKE является согласование контекстов безопасности для семейства протоколов IPsec, однако предусмотрено его использование и для согласования контекстов безопасности других протоколов.

Домен интерпретации (Domain of Interpretation), доопределяя структуру блоков данных и задавая правила именования информации, определяющей безопасность (политики безопасности, криптографические режимы и алгоритмы), связывает IKE с протоколом, защита которого обеспечивается.

Защитный набор (protection suite) – это список средств защиты трафика, которые могут предоставляться различными протоколами безопасности. Например, защитный набор может включать алгоритм шифрования DES для протокола инкапсулирующей защиты данных ESP [8] и алгоритм вычисления управляемой ключом хэш-функции MD5 для протокола аутентифицирующего заголовка AH [7]. Следует рассматривать и анализировать защиту, предоставляемую защитным набором в целом, поскольку средства защиты различных протоколов безопасности могут сложным образом влиять друг на друга.

Теперь рассмотрим логически более крупные объекты взаимодействия. Сообщения группируются в **режимы** (modes), которые также иногда называются **обменами** (exchanges, терминология ISAKMP), – определённые последовательности сообщений, обеспечивающие аутентифицированный обмен ключевой и управляющей информацией, задающие роли сторон, порядок следования, а также типы конкретных блоков данных в конкретных сообщениях. Режим может состоять из 6 сообщений, из 4 сообщений, из 3 сообщений, из одного сообщения.

В работе IKE можно выделить два основных этапа, называемых **фазами** (phases). В первой фазе выполняется взаимная аутентификация участников, и устанавливаются сеансовые ключи. Предполагается, что каждый из участников первой фазы IKE имеет свой идентификатор, известный другой стороне, а также связанный с этим идентификатором секрет, который может быть проверен другой стороной. В качестве такого секрета могут выступать заранее распределённые симметричные секретные ключи или секретный ключ пары ключей в асимметричной системе (системе с открытым ключом), на основе которых и выполняется взаимная аутентификация.

Первая фаза является вспомогательной по отношению ко второй. В её ходе IKE создаёт контекст безопасности IKE (IKE SA), который определяет, как именно будет обеспечиваться защита последующего трафика, в частности, обменов второй фазы. Таким образом, "полезная работа" IKE осуществляется в ходе второй фазы, когда и согласуются, модифицируются и удаляются контексты безопасности (которых может быть несколько) для протокола, использующего IKE. Для семейства IPsec такими контекстами безопасности являются ESP SA и AH SA (общим названием этих контекстов безопасности является IPsec SA).

Несмотря на очевидные недостатки такого решения (накладные расходы на проведения первой фазы, усложнение протокола), в большинстве случаев это оправдано.

Во-первых, участники взаимодействия могут уменьшить стоимость первой фазы, выполняя для каждой такой фазы обмены второй фазы несколько раз. Это позволяет согласовать несколько контекстов безопасности, не начиная каждый раз всё соединение заново. Таким образом, одной первой фазе может соответствовать несколько вторых фаз, и в каждой второй фазе может быть

согласовано несколько IPsec SA. Количество вторых фаз, которые можно осуществить на основе одной заранее проведённой первой фазы ограничивается лишь соображениями необходимости обновления ключей.

Во-вторых, в ходе первой фазы согласовываются параметры безопасности второй фазы. Таким образом, в зависимости от наших потребностей мы можем выбрать различные степени защиты для второй фазы.

В-третьих, наличие IKE SA позволяет значительно снизить накладные расходы на работу IKE в целом. Образуется управляющий контекст безопасности, или "защищённый канал", без которого участникам пришлось бы проходить через полную процедуру взаимной аутентификации для каждого сообщения об ошибке, информационного сообщения или сообщения об удалении контекста безопасности.

Следует ещё раз подчеркнуть принципиальную разницу между первой и второй фазами. Услуги безопасности в разных фазах могут применяться по-разному. Например, в ходе разных фаз могут аутентифицироваться различные сущности. В ходе первой фазы друг друга аутентифицируют оконечные узлы IKE-соединения, тогда как в ходе второй фазы аутентификация осуществляется для пользователей или прикладных программ.

Следует также отличать контексты безопасности IKE (IKE SA) от контекстов безопасности протоколов, применяющих IKE, в частности, от IPsec SA. Общая идея одна и та же: контекст безопасности – это абстракция, воплощающая в себе политику безопасности и ключ. Однако реализованы эти контексты безопасности по-разному. И дело отнюдь не только в их принципиальной неравноправности, о которой шла речь выше. Например, контексты безопасности IKE SA – двунаправленные, в отличие от однонаправленных контекстов безопасности IPsec SA.

2.3. Организация сообщений IKE и обозначения

Сообщения протокола IKE имеют достаточно сложную структуру. Для понимания того, как работает протокол, знать все детали не обязательно. В этом подразделе приводится краткая информация по организации сообщений, вводятся условные обозначения, применяемые далее при описании работы различных режимов. Подробная информация о формате сообщений и блоков данных ISAKMP/IKE дана в приложении 3.

Сообщение состоит из заголовка фиксированной структуры и произвольного числа блоков данных. Блок данных – это структурный элемент сообщения IKE. Он состоит из стандартного заголовка, общего для всех блоков данных, и содержимого, индивидуального для каждого типа блока данных. Некоторые поля многих блоков и, соответственно, сами блоки имеют переменную длину. Некоторые блоки данных синтаксически и семантически являются зависимыми (второстепенными) блоками данных.

При описании организации сообщений в различных режимах будут применяться следующие обозначения:

HDR – Заголовок IKE

CKY_x – Идентифицирующая цепочка (cookie) "x". В заголовке сообщения IKE имеются два поля: идентифицирующая цепочка инициатора – CKY_I и идентифицирующая цепочка ответчика – CKY_R.

SA – Контекст безопасности. В описываемых далее схемах обменов аббревиатура "SA" обозначает не один блок данных Контекст Безопасности, а целый набор блоков данных, состоящий из блока Контекст Безопасности (SA payload), одного или нескольких соответствующих ему блоков Предложение (P – proposal payload) и одного или нескольких блоков Преобразование (T - transform payload), соответствующих каждому из этих блоков Предложение.

XX_b – Содержимое блока данных XX (исключая стандартный заголовок блоков данных).

SAx_b – Содержимое блока данных Контекст Безопасности, включая содержимое зависимых от него блоков Преобразование и Предложение (за исключением заголовков).

KE – Блок данных Обмен Ключами (Key Exchange payload)

IDx – Блок данных Идентификация (Identification payload) для "x". "x" может принимать значения "i" или "r" для инициатора и ответчика соответственно.

HASH – Блок данных Хэш (Hash payload).

SIG – Блок данных Подпись (Signature payload).

AUTH – Указывает на общий механизм аутентификации, HASH или SIG.

CERT – Блок данных Сертификат (Certificate payload).

NONCE – Блок данных Одноразовый Номер (Nonce payload).

Nx – Обозначение блока данных Одноразовый Номер в тех случаях, когда необходимо подчеркнуть роль участника взаимодействия, "x" может принимать значения "i" или "r" для инициатора и ответчика соответственно.

[...] – Необязательный блок данных.

(A)b – Блок данных A, зашифрованный ключом b (может применяться симметричное шифрование либо шифрование с открытым ключом).

***** – Указывает на то, что сообщение зашифровано. Шифрование должно применяться ко всем блокам данных сообщения, начиная с первого блока данных, следующего за заголовком. Сам заголовок не зашифровывается.

=> – Указывает на направление сообщения "от инициатора к ответчику".

<= – Указывает на направление сообщения "от ответчика к инициатору".

2.4. Агрессивный и основной режимы первой фазы

В спецификации IKE определяются два типа обменов первой фазы, которые называются режимами. В **агрессивном режиме** (Aggressive Mode) взаимная аутентификация и установка сеансовых ключей выполняется с помощью трех сообщений. В **основном режиме** (Main Mode) для этой цели используются шесть сообщений. За счет трех дополнительных сообщений в этом режиме участникам обмена предоставляются более широкие функциональные возможности, в частности, обеспечивается возможность скрыть от подслушивания идентификаторы оконечных точек, а также бóльшая гибкость при согласовании криптографических алгоритмов.

На первом шаге любого обмена выполняется обмен **идентифицирующими цепочками** CKY_I и CKY_R, которые создают инициатор и ответчик соответственно. Каждая идентифицирующая цепочка (cookie) представляет собой 8-байтовое псевдослучайное число и является уникальной для данного удаленного партнера, а также для конкретного обмена, в котором она определяется. Идентифицирующие цепочки, как следует из их названия, предназначены для идентификации конкретного обмена (а следовательно, и контекста безопасности IKE SA), а также для обеспечения защиты от некоторых видов атак на доступность (DOS – denial of service attack). Они позволяют задержать выполнение участниками интенсивных вычислений (например, операций возведения в степень для обмена Диффи-Хеллмана) до тех пор, пока не завершится проверка правильности возвращенной идентифицирующей цепочки. Таким образом, злоумышленник, генерирующий множество фальшивых сообщений IKE с подделанными адресами возврата, не может заставить атакуемый узел выполнять никакую существенную работу, поскольку не будет получено второе сообщение, которое содержит уникальную для данного (фальшивого) адреса идентифицирующую цепочку.

Обычно для генерации идентифицирующих цепочек используется метод, предложенный в протоколе обмена ключами Photuris [6]. Идентифицирующая цепочка является результатом вычисления хэш-функции, аргументами которой являются уникальный идентификатор партнера (например, его IP-адрес, порт и протокол), а также секрет, известный только генерирующей стороне, и временная метка. Таким образом, каждая идентифицирующая цепочка привязывается к удаленному партнеру. Идентифицирующие цепочки помещаются в заголовок сообщений IKE (см. приложение 3). Такой механизм обеспечивает простейшую проверку того, что ответное сообщение партнёра – это ответ именно на наше сообщение: в заголовке ответного сообщения должна стоять идентифицирующая цепочка, которую мы только что послали.

Общая схема взаимодействия в основном режиме имеет следующий вид (заметим, что на схеме указаны лишь необходимые блоки данных; для разных

методов аутентификации структура сообщений может несколько отличаться):

```
 №     Инициатор                 Ответчик
(1)    HDR, SA          =>
(2)                     <=       HDR, SA
(3)    HDR, KE, NONCE   =>
(4)                     <=       HDR, KE, NONCE
(5)    HDR*, IDi, AUTH  =>
(6)                     <=       HDR*, IDr, AUTH
```

В первом сообщении инициатор посылает свою идентифицирующую цепочку (в заголовке сообщения) и предлагает адекватную с его точки зрения защиту трафика в блоке данных Контекст Безопасности, а также связанных с ним блоках Предложение и Преобразование (все эти блоки на схеме обозначаются как SA). Во втором сообщении ответчик посылает свою идентифицирующую цепочку и сообщает о своём выборе одного из защитных наборов в блоках Контекст Безопасности, Предложение, Преобразование. Заметим, что во всех последующих сообщениях идентифицирующие цепочки инициатора и ответчика помещаются в соответствующие фиксированные поля заголовков, обеспечивая идентификацию конкретного обмена.

В третьем и четвёртом сообщениях инициатор и ответчик обмениваются ключевой информацией (в блоках KE передаются открытые значения Диффи-Хеллмана) и случайными одноразовыми номерами (Nonce), защищающими от атаки повторного воспроизведения сообщений. В зависимости от выбранного метода аутентификации некоторые из передаваемых блоков данных могут зашифровываться.

В пятом и шестом сообщениях инициатор и ответчик обмениваются своими идентификаторами (в блоках IDi и IDr соответственно) и значениями согласованных аутентифицирующих функций (AUTH). Таким образом каждая из сторон подтверждает, что она знает соответствующий секрет (например, секретный ключ подписи или заранее распределенный секретный ключ). Эти данные передаются под защитой общего секрета, полученного в результате обмена Диффи-Хеллмана в третьем и четвертом сообщениях. Шифрованию подвергается всё сообщение (точнее, содержимое всех блоков данных). В зависимости от выбранного метода аутентификации вместо изображённого на схеме абстрактного блока данных AUTH посылается блок данных HASH или блок данных SIG. При аутентификации на основе шифрования открытыми ключами блоки IDi и IDr передаются в третьем и четвертом сообщениях соответственно.

Для проведения первой фазы может использоваться также и агрессивный режим. Схема взаимодействия в агрессивном режиме имеет следующий вид:

```
 №     Инициатор                 Ответчик
(1)    HDR,SA,KE,NONCE,IDi  =>
(2)                         <=   HDR,SA,KE,NONCE,IDr,AUTH
(3)    HDR,AUTH             =>
```

В агрессивном режиме имеются только три сообщения. В первых двух сообщениях, помимо согласования защитного набора (блоки SA), выполняется обмен Диффи-Хеллмана (блоки KE), обеспечивающий установление сеансовых ключей, а также обмен одноразовыми номерами (блоки NONCE) и идентификаторами участников (блоки IDi, IDr). Во втором и третьем сообщениях путем передачи соответствующего аутентифицирующего кода AUTH каждая сторона подтверждает, что она знает как секретное значение Диффи-Хеллмана, так и секрет, связанный с соответствующим идентификатором.

Заметим, что в основном режиме инициатор имеет возможность перечислить в порядке приоритета все криптографические алгоритмы, которые он поддерживает, а ответчик делает выбор одного защитного набора. В агрессивном режиме возможности инициатора по согласованию параметров контекста безопасности IKE SA ограничены, поскольку он уже не может предложить разные группы Диффи-Хеллмана для разных защитных наборов.

2.5. Режимы второй фазы

После завершения первой фазы и установления между инициатором и ответчиком контекста безопасности IKE SA любой из участников обмена может стать инициатором второй фазы. Во второй фазе могут выполняться обмены быстрого режима, режима новой группы, а также информационного режима.

Быстрый режим (Quick Mode) используется для установления контекста безопасности IPsec (IPsec SA, т.е. ESP SA и/или AH SA). Процесс установления IPsec SA включает согласование криптографических параметров, выполнение при необходимости дополнительного обмена Диффи-Хеллмана (если требуется совершенная прогрессирующая секретность), а также согласование параметров трафика, который будет посылаться под защитой этого контекста безопасности. Схема этого обмена имеет следующий вид:

```
 №   Инициатор                        Ответчик
(1)  HDR*, AUTH(1), SA,
       Ni[, KE ] [, IDi, IDr ]    =>
(2)                                 <= HDR*, AUTH(2), SA,
                                        Nr[, KE ] [, IDi, IDr ]
(3)  HDR*, AUTH(3)                  =>
```

Сообщения быстрого режима защищены уже установленным контекстом безопасности IKE SA: все блоки данных зашифровываются с помощью алгоритма, согласованного в ходе первой фазы. В первом сообщении инициатор предлагает защитные наборы, описывающие IPsec SA, передаёт свой одноразовый номер Ni, блок данных с аутентифицирующим кодом AUTH(1) и, возможно, блоки KE и/или IDi, IDr. Ответчик сообщает выбранный защитный набор, предоставляет аутентифицирующий код

сообщения AUTH(2) и свой одноразовый номер Nr, а также, возможно, блоки KE и/или IDi, IDr. В третьем сообщении, завершающем обмен, инициатор передаёт свой блок данных с аутентифицирующим кодом AUTH(3). Заметим, что правила вычисления кодов аутентификации обеспечивают не только защиту целостности сообщений и аутентификацию их источника, но также подтверждают (благодаря использованию одноразовых номеров), что партнер взаимодействия действительно существует и является активным участником обмена (подробнее об этом см. в п. 3.5.3).

IPsec позволяет каждому участнику взаимодействия установить определенные ограничения на трафик, который пересылается под защитой конкретного контекста безопасности второй фазы. Эти ограничения отражают степень детализации политики участников и называются селекторами трафика (traffic selectors). В качестве селекторов трафика могут использоваться IP-адреса (одиночный IP-адрес, диапазон IP-адресов, или адрес подсети, задаваемый IP-адресом и маской), тип протокола (поле заголовка IP, в котором указывается номер протокола UDP, TCP и т.д.) или номера портов TCP/UDP.

Информация о селекторах, отражающая IPsec-политику участника взаимодействия IKE, может передаваться инициатором в быстром режиме в необязательных блоках данных идентификации первого сообщения (IDi, IDr). Эта информация относится ко всем контекстам безопасности, которые согласуются в рамках одного обмена в быстром режиме. Ответчик может использовать её для проверки соответствия его собственной политике и либо принять эти селекторы в точности в том виде, в котором они были указаны, либо отвергнуть их.

Схемы обменов в режиме Новой Группы (New Group Mode), который служит для введения нестандартных параметров обмена Диффи-Хеллмана, и в Информационном режиме (Informational Exchange), использующемся для информирования об ошибках и обслуживания уже существующих контекстов безопасности, будут рассмотрены в подразделах 3.7 и 3.6 соответственно.

3. Особенности работы протокола IKE

В разделе 2 был представлен краткий обзор протокола IKE, в котором некоторые особенности его работы были сознательно опущены. Рассмотрим протокол IKE в следующем приближении. Что конкретно согласуется в ходе обменов? Как именно на основании согласовываемых данных создаются криптографические параметры? Как конкретно выглядят схемы обменов в различных режимах при разных методах аутентификации? Чем отличается функциональность протокола для разных методов аутентификации?

3.1. Согласование криптографических параметров

Параметры контекстов безопасности, устанавливаемых с помощью протокола IKE, определяются политиками безопасности участников взаимодействия; эти

политики могут быть достаточно сложными. В ходе переговоров стороны должны выработать и согласовать общую политику безопасности. Необходимые криптографические параметры для первой и второй фаз IKE несколько различаются. Поэтому рассмотрим их по отдельности.

Как отмечалось в разд. 2, в первой фазе обмена происходит согласование параметров контекста безопасности IKE (IKE SA). Эти параметры (алгоритм шифрования, алгоритм вычисления хэш-функции, метод аутентификации и группа Диффи-Хеллмана) составляют защитный набор. Таким образом, в первом сообщении первой фазы IKE инициатор предлагает множество приемлемых для него защитных наборов, а ответчик осуществляет выбор одного из них. Заметим, что при этом режим работы (основной или агрессивный) не согласуется. Инициатор принимает решение, какой из режимов выполнять, и указывает свой выбор. Для этого в заголовке сообщений имеется специальное поле типа обмена - Exchange Type (см. приложение 3). Если, например, инициатор выбирает агрессивный режим, то все предлагаемые защитные наборы должны иметь одну и ту же группу Диффи-Хеллмана.

Поскольку в первой фазе согласуются параметры только одного контекста безопасности (IKE SA), первое сообщение содержит только один блок данных Контекст Безопасности (SA) и только один связанный с ним блок данных Предложение (P). Каждый из указанных выше обязательных параметров защитного набора содержится в отдельном блоке данных Преобразование (T), фактически вложенном в блок данных P.

В качестве примеров алгоритмов шифрования в спецификации IKE приведены алгоритмы DES-CBC, IDEA-CBC, Blowfish-CBC, RC5-R16-B64-CBC, 3DES-CBC и CAST-CBC, но обязательным для реализации объявлен только первый алгоритм. Примерами алгоритмов хэш-функций являются алгоритмы MD5, SHA и Tiger; при этом обязательными для реализации объявлены первые два алгоритма. Спецификация определяет четыре метода выполнения взаимной аутентификации сторон: на основе заранее распределенных ключей, на основе цифровой подписи (в криптосистемах RSA и DSS), на основе шифрования с открытым ключом в системе RSA с использованием старого стандартного протокола и на основе шифрования с открытым ключом в системе RSA с использованием модифицированного протокола. Метод аутентификации на основе заранее распределенных ключей является обязательным для реализации.

Группа Диффи-Хеллмана определяет параметры ключевого материала для обмена Диффи-Хеллмана. В спецификации IKE определяются четыре группы Диффи-Хеллмана и присваиваются им значения, которые и подлежат согласованию. В двух группах используется традиционное возведение в степень по модулю простого числа с конкретными значениями g и p (MODP), в двух других группах – эллиптические кривые (ECP и EC2N). Группы имеют следующие номера:

1. Группа MODP с 768-битным модулем
2. Группа MODP с 1024-битным модулем
3. Группа ECP с 155-битным модулем
4. Группа EC2N с 185-битным модулем

Спецификация IKE допускает использование и других групп Диффи-Хеллмана, при этом участники взаимодействия могут определить свою собственную группу с помощью обмена новой группой (New Group Exchange). Из стандартных групп только группа 2 является обязательной для реализации. Считается, что группы 1 и 3 предоставляют приблизительно одинаковую степень безопасности. Группы 2 и 4 также обеспечивают сходную степень безопасности. Основное отличие между сходными группами заключается в вычислительной сложности.

В дополнение к этим обязательным параметрам существуют необязательные параметры, которые могут быть согласованы как часть защитного набора. Важнейшим необязательным параметром является время жизни (lifetime). Когда время жизни близится к исчерпанию, контекст безопасности должен быть закрыт и, в случае необходимости, должен быть установлен новый контекст безопасности. В спецификации определяются два способа измерения времени жизни контекста безопасности: по объему переданных под его защитой данных и по продолжительности его существования. Время жизни считается исчерпанным, когда превышено любое из установленных ограничений. Заметим, что чем больше время жизни контекста безопасности, тем выше угроза рассекречивания ключей, чем оно меньше, тем выше накладные расходы на переустановку контекстов безопасности.

Следует отметить, что во всех обменах в основном режиме могут быть согласованы все обязательные параметры защитного набора. Однако ни один вариант агрессивного режима не допускает согласования группы Диффи-Хеллмана, поскольку инициатор уже выбрал одну из них и включил свое открытое значение Диффи-Хеллмана в блок данных KE первого сообщения. При этом для правильной интерпретации ответчиком номер соответствующей группы должен быть указан в блоке данных T этого сообщения. Таким образом, агрессивный режим накладывает определенные ограничения на возможности согласования, хотя в нем могут согласовываться некоторые криптографические алгоритмы. Не могут согласовываться только те алгоритмы, которые инициатор использовал в первом сообщении. Например, в обоих вариантах шифрования с открытым ключом инициатор может послать хэш-значение сертификата ответчика (позволяя последнему определить, какой из его ключей необходим для расшифровывания информации, которую ему посылает инициатор), так что инициатор должен сам выбрать алгоритм вычисления хэш-функции. А в агрессивном режиме с аутентификацией на основе шифрования с открытым ключом с использованием модифицированного протокола инициатор дополнительно использует шифрование секретным ключом и поэтому должен сам выбрать алгоритм

шифрования открытым ключом. Он не может предлагать для согласования что-либо отличное от того, что он уже использовал.

Во второй фазе IKE происходит согласование параметров контекстов безопасности IPsec (AH SA и ESP SA). Поскольку в качестве защитных средств здесь могут предлагаться только AH, только ESP, AH+ESP или любое из указанных средств плюс протокол IP-компрессии, то один блок данных SA первого сообщения в быстром режиме может включать несколько блоков данных Предложение (P). В каждом предложении могут указываться несколько необходимых преобразований (блоков данных T, связанных с соответствующим блоком данных P).

Каждому блоку данных Предложение инициатором присваивается определенный номер (в поле Proposal Number, см. приложение 3). Эти номера используются для выражения логических операторов OR (ИЛИ) или AND (И). Совпадающие номера предложений означают, что соответствующие блоки данных P составляют одно предложение, т.е. реализуется функция AND (логического И), а отличающиеся номера блоков данных Предложение определяют разные предложения, т.е. реализуется функция OR (логического ИЛИ). Например, если политика IPsec требует "AH AND ESP", то блоки данных Предложение для каждого отдельного предложения (по одному блоку P для каждого протокола) будут иметь одинаковые номера. Если политика IPsec требует "AH OR ESP", то блоки данных Предложение для каждого отдельного предложения (по одному блоку P для каждого протокола) будут иметь разные номера. Таким способом можно создавать достаточно сложные предложения. В приложении 3 приведены наглядные примеры составления сложных предложений контекстов безопасности.

Следует отметить, что если во второй фазе предлагается протокол AH, то в каждый предлагаемый защитный набор входят только алгоритм вычисления кода аутентификации (защиты целостности) и необязательный параметр – время жизни. Каждый защитный набор для протокола ESP включает алгоритм шифрования, алгоритм вычисления кода аутентификации и необязательный параметр – время жизни. Для протокола IP-компрессии должен указываться соответствующий алгоритм.

3.2. Формирование ключевого материала

Описанные в предыдущем разделе согласуемые параметры контекста безопасности IKE SA не зашифровываются (их согласование происходит в первых сообщениях любого обмена в основном или агрессивном режимах) и передаются в открытом виде. Однако заметим, что каждая из сторон поддерживает также некоторую секретную информацию, не видимую тому, кто читает (или перехватывает) сообщения. Эта секретная информация представляет собой аутентифицируемые ключи, применяемые для защиты сообщений IKE, а также для создания ключей для других сервисов безопасности.

С целью защиты целостности (аутентификации) и шифрования последних сообщений первой фазы и всех сообщений второй фазы, а также для создания источника ключевого материала, который должен подмешиваться к информации в обменах второй фазы при формировании уникальных ключей соответствующих контекстов безопасности, в первой фазе IKE устанавливаются три вида ключей, получившие названия SKEYID_a, SKEYID_e и SKEYID_d. Ключ SKEYID_a используется для обеспечения целостности данных и аутентификации источника сообщений IKE; ключ SKEYID_e – для шифрования сообщений IKE; и ключ SKEYID_d – для генерации ключевого материала для контекстов безопасности IPsec SA, которые устанавливаются во второй фазе. Эти три ключа генерируются на основе секрета, получившего название SKEYID, значение которого зависит от согласованного метода аутентификации. По существу, сгенерированные ключи представляют собой значения, полученные в результате применения хэш-функции к используемым в обмене значениям Диффи-Хеллмана, одноразовым номерам, идентифицирующим цепочкам, а в случае заранее распределенного секрета, и к этому секрету. Таким образом, они формируются из секретного материала, известного только активным участникам обмена.

В спецификации IKE под термином "псевдослучайная функция" (prf – pseudo random function) понимается зависимая от ключа хэш-функция prf(key, msg), у которой имеются два аргумента – ключ и данные. Функция prf используется как для создания новых ключей, так и для аутентификации. Алгоритм вычисления хэш-функции согласуется как часть защитного набора; поэтому, как правило, в качестве функции prf используется версия HMAC согласованной хэш-функции. В приложении 2 приводится краткая справка о криптографических хэш-функциях, MAC, HMAC, цифровых подписях.

При генерации SKEYID используются идентифицирующая цепочка инициатора (CKY_I), его одноразовый номер (Ni), а также идентифицирующая цепочка ответчика (CKY_R) и его одноразовый номер (Nr). Мы уже отмечали, что обмен идентифицирующими цепочками и одноразовыми номерами защищает от атак на доступность, которые связаны с навязыванием интенсивных вычислений, присущих криптографии с открытым ключом, и обеспечивает защиту от воспроизведения сообщений. Таким образом, каждый из участников обмена демонстрирует своему партнеру, что он действительно существует и обладает идентифицирующей цепочкой и одноразовым номером партнера, а также доказывает партнеру, что тот имеет дело не с сохранёнными и воспроизведёнными сообщениями одного из прошлых обменов.

Результатом обмена Диффи-Хеллмана является общий секрет Диффи-Хеллмана g^{xy}, известный обоим участникам. Для удобства чтения мы будем опускать "mod p" и предполагать, что возведение в степень, например, вычисление g^{xy} выполняется по модулю p.

Как уже отмечалось, в основном или агрессивном режиме обмена возможны четыре метода аутентификации: цифровая подпись, два вида аутентификации с шифрованием открытым ключом и заранее распределенный ключ. IKE определяет SKEYID для каждого метода аутентификации следующим образом (символ "|" обозначает конкатенацию):

- Для аутентификации заранее распределённым ключом:
  SKEYID = prf(заранее распределённый ключ, Ni_b | Nr_b);
- Для аутентификации цифровыми подписями:
  SKEYID = prf(Ni_b | Nr_b , g^{xy});
- Для аутентификации с шифрованием открытым ключом:
  SKEYID = prf(hash(Ni_b | Nr_b), CKY_I | CKY_R).

На основе SKEYID вычисляются производные ключи:

SKEYID_d = prf(SKEYID , g^{xy} | CKY_I | CKY_R | 0)
SKEYID_a = prf(SKEYID , SKEYID_d | g^{xy} | CKY_I | CKY_R | 1)
SKEYID_e = prf(SKEYID , SKEYID_a | g^{xy} | CKY_I | CKY_R | 2)

Детали генерации этих секретных ключей, например, вопросы выравнивания результатов различной длины при использовании различных функций prf, вопросы выбора вектора инициализации для некоторых алгоритмов шифрования приводятся в приложении B к RFC 2408 [2]. Ключ, используемый для шифрования, формируется на основе SKEYID_e в соответствии с согласованным алгоритмом. Детали этого процесса приведены там же.

Обмены первой фазы аутентифицируются каждой из сторон путём вычисления значений хэш-функции. И сторона, аутентифицирующая сообщение, и получатель сообщения применяют хэш-функцию к одним и тем же элементами сообщения, используя общий ключ. Вычисление соответствующего значения хэш-функции аутентифицирует сообщение, поскольку считается, что, во-первых, невозможно определить аргументы хэш-функции по ее значению, во-вторых, ключ должен быть известен лишь участникам обмена, и, в-третьих, одни и те же аргументы хэш-функции всегда порождают один и тот же результат. Для аутентификации любого обмена инициатор генерирует величину HASH_I, а ответчик генерирует величину HASH_R, которые вычисляются следующим образом:

HASH_I = prf(SKEYID, g^{xi} | g^{xr} | CKY_I | CKY_R | SAi_b | IDi_b)
HASH_R = prf(SKEYID, g^{xr} | g^{xi} | CKY_R | CKY_I | SAi_b | IDr_b)

Здесь g^{xi} и g^{xr} – открытые значения Диффи-Хеллмана для инициатора и ответчика соответственно. SAi_b – блок данных Контекст Безопасности со всеми зависимыми блоками данных Предложение и Преобразование, предлагаемый инициатором. IDi_b и IDr_b – блоки идентификационных данных инициатора и ответчика соответственно.

Для аутентификации цифровыми подписями значения HASH_I и HASH_R подписываются и передаются в блоках данных Подпись (SIG), а для

аутентификации с шифрованием открытыми ключами или заранее распределенными ключами величины HASH_I и HASH_R передаются в блоках данных HASH и непосредственно аутентифицируют обмен.

Использование полного блока данных Контекст Безопасности (т.е. блока SA со всеми зависимыми блоками Предложение и Преобразование) в качестве исходных данных при вычислении значения хэш-функции важно, поскольку это позволяет предотвратить атаку типа "человек посередине", когда злоумышленник модифицирует защитный набор, выбирая самый слабый из предложенных наборов. Пусть, например, оба участника в качестве алгоритма шифрования хотят использовать 3DES, однако готовы принять и DES, если этого требует партнер. Если бы при аутентификации не вычислялось значение хэш-функции от полного блока Контекст Безопасности, то злоумышленник имел бы возможность заменить запрос "3DES OR DES" на "DES", что заставило бы стороны применять более слабую защиту.

Метод аутентификации влияет на содержимое и использование сообщений в ходе первой фазы IKE, но не на их назначение. Как уже отмечалось, всего существует четыре метода аутентификации, которые будут более подробно описаны в следующих разделах. Наличие различных методов аутентификации вынуждает рассматривать все варианты каждого из режимов, то есть имеются четыре вида основного режима и четыре вида агрессивного режима.

Формирование ключевого материала для контекстов безопасности IPsec SA и аутентификация сообщений второй фазы будут рассмотрены в разд. 3.5.

3.3. Основной режим

В основном режиме (Main Mode) для создания IKE SA используются шесть сообщений, которые составляют три шага его выполнения. На первом шаге происходит согласование контекста безопасности, на втором – обмен открытыми значениями Диффи-Хеллмана и одноразовыми номерами, на третьем – взаимная аутентификация. Основной режим обеспечивает сокрытие идентификаторов участников и полное использование возможностей ISAKMP по согласованию параметров контекстов безопасности.

3.3.1. Аутентификация заранее распределёнными ключами

При использовании аутентификации заранее распределёнными ключами основной режим принимает следующий вид:

№	Инициатор		Ответчик
(1)	HDR, SA	=>	
(2)		<=	HDR, SA
(3)	HDR, KE, Ni	=>	
(4)		<=	HDR, KE, Nr
(5)	HDR*, ID-i, HASH_I	=>	
(6)		<=	HDR*, ID-r, HASH_R

В сообщениях (1),(2) участники согласовывают параметры IKE SA и

обмениваются идентифицирующими цепочками, которые размещаются в соответствующих полях заголовков. В сообщениях (3),(4) происходит обмен открытыми значениями Диффи-Хеллмана (вычисленными в группе, которая согласована как часть защитного набора в сообщениях (1),(2)) и одноразовыми номерами. По окончанию шагов (1)-(4) участники могут вычислить общий секрет Диффи-Хеллмана, общий секрет SKEYID и создать производные ключи. В сообщениях (5),(6), которые зашифровываются ключом SKEYID_e, значение которого обусловлено сообщениями (3) и (4), участники идентифицируют себя и обмениваются блоками данных HASH_I и HASH_R для проведения взаимной аутентификации. Таким образом, каждая из сторон подтверждает, что она знает заранее распределенный секретный ключ, связанный с ее идентификатором.

Следует отметить, что поскольку инициатор посылает свой идентификатор в сообщении (5), зашифрованном ключом SKEYID_e, который является функцией общего заранее распределенного секретного ключа, ответчик не может расшифровать это сообщение без знания этого общего секрета, и для поиска соответствующего ключа он должен знать идентификатор инициатора. По существу, это означает, что идентификатором инициатора может быть только IP-адрес, что является ограничением основного режима с аутентификацией заранее распределёнными ключами. В большинстве случаев проблем не возникает, однако в ряде ситуаций, например, когда инициатор обмена без заранее известного IP-адреса (мобильный пользователь) соединяется с сервером, такой метод порождения контекста безопасности неприемлем: ответчик не может хранить заранее распределённые ключи для априори неизвестного IP-адреса. Решением этой проблемы может быть как использование аутентификации подписями на основе открытых ключей, так и применение агрессивного режима вместо основного. Заметим, что в разд. 4 рассматриваются недостатки протокола ISAKMP/IKE и некоторые дополнительные детали обменов первой фазы.

3.3.2. Аутентификация подписями

В основном режиме может быть применена аутентификация подписями на основе открытых ключей в криптосистемах RSA и DSS. В приложении 2 содержится криптографическая справка, в которой даны определения и основные свойства хэш-функций, MAC, HMAC и цифровых подписей. С использованием аутентификации подписями основной режим принимает следующий вид:

```
 №   Инициатор                      Ответчик
(1)  HDR, SA                  =>
(2)                           <=   HDR, SA
(3)  HDR, KE, Ni, [CR]
(4)                           <=   HDR, KE, Nr, [CR]
(5)  HDR*, IDi, [CERT,] SIG_I =>
(6)                           <=   HDR*, IDr, [CERT,] SIG_R
```


Внешне этот обмен выглядит похожим на обмен с использованием заранее распределённых ключей. Основное отличие заключается в том, что теперь аутентификация обеспечивается цифровой подписью, а не просто значением хэш-функции. Блоки данных SIG_I, SIG_R содержат результат применения согласованного алгоритма цифровой подписи к вычисленным значениям HASH_I и HASH_R соответственно. Более подробная информация, касающаяся особенностей вычисления значений подписей в криптосистемах RSA и DSS, приводится в спецификации RFC 2409 [3].

Открытые ключи обычно извлекаются из сертификатов. С помощью необязательных блоков данных CERT (сертификат) и CR (запрос сертификата) IKE позволяет передавать сертификаты, а также запрашивать сертификат у участника, с которым ведутся переговоры.

3.3.3. Аутентификация на основе шифрования открытыми ключами - стандартный метод

Если для аутентификации вместо подписей применяется шифрование с открытыми ключами, то возникает одно важное дополнительное свойство. При использовании подписей или заранее распределенных ключей одна из сторон должна первой открыть свой идентификатор другой стороне. Если первым открывает свой идентификатор ответчик, то злоумышленник может инициировать соединение IPsec с использованием его IP-адреса, чтобы выяснить, кто там находится. Если первым открывает свой идентификатор инициатор, то активный злоумышленник может имитировать IP-адрес ответчика, чтобы увидеть, кто с ним может создать соединение. При шифровании с открытым ключом можно сделать так, чтобы обе стороны открывали свои идентификаторы только тем, перед кем они намереваются аутентифицировать себя, путем шифрования открытым ключом другой стороны своих идентификаторов и любой другой идентифицирующей информации (например, их сертификатов). Но это может быть сделано, только если, по крайней мере, одна из сторон уже знает открытый ключ шифрования другой стороны.

Существуют два метода аутентификации с использованием шифрования открытыми ключами – стандартный (standard public key encryption) и модифицированный (revised public key encryption). Модифицированный метод возник в связи с нареканиями в адрес стандартного, при использовании которого необходимо выполнять четыре отдельные вычислительно сложные операции шифрования/расшифровывания в асимметричной системе. Стандартный режим оставлен в спецификации IKE для сохранения совместимости с уже существующими реализациями.

Заметим, что оба этих метода не гарантируют соблюдение участниками своих обязательств, то есть каждый из участников может впоследствии заявить, что он не принимал участия в обмене, и отказаться от своих обязательств, даже если все сообщения обмена были сохранены.

Схема обмена сообщениями в стандартном методе имеет следующий вид:

```
 №   Инициатор                        Ответчик
(1)  HDR, SA                    =>
(2)                             <=  HDR, SA
(3)  HDR, KE, [HASH(1),]
       (IDi_b)PubKey_r,
       (Ni_b)PubKey_r           =>
(4)                             <=  HDR, KE, [HASH,]
                                    (IDr_b)PubKey_i,
                                    (Nr_b)PubKey_i
(5)  HDR*, HASH_I               =>
(6)                             <=      HDR*, HASH_R
```

В сообщении (3) имеются два поля (IDi_b и Ni_b – идентификатор и одноразовый номер инициатора), которые инициатор должен по отдельности зашифровать открытым ключом ответчика, а последний должен их расшифровать своим секретным ключом. Аналогичные операции должны быть выполнены и для сообщения (4): ответчик должен по отдельности зашифровать два поля (IDr_b и Nr_b – идентификатор и одноразовый номер ответчика) открытым ключом инициатора, который должен их расшифровать своим секретным ключом. А такая работа требует достаточно интенсивных вычислений. Заметим, что зашифровываются только тела соответствующих блоков данных, а заголовки блоков данных остаются открытыми.

Чтобы выполнить шифрование открытым ключом, инициатор уже должен иметь открытый ключ ответчика. Заметим, что, поскольку открытые ключи обычно получаются из соответствующих сертификатов, в случае, когда ответчик имеет несколько открытых ключей, в третьем сообщении пересылается хэш-значение сертификата (блок данных HASH(1)), на основании которого инициатор осуществлял шифрование служебной информации. Таким способом ответчик может определить, какой из своих секретных ключей он должен использовать для расшифровывания зашифрованных блоков данных.

Следует также отметить, что при использовании этого метода аутентификации ни одна из сторон не может запросить сертификат у противоположной стороны. Обмен сертификатами нарушил бы сокрытие идентификаторов участников. Таким образом, в данном случае необходим какой-то другой (за рамками IKE) способ получения сертификатов. Поскольку инициатор знает, с кем он начинает диалог, то обычно для него проблема получения сертификата ответчика не стоит.

В этом методе для аутентификации используются одноразовые номера, зашифрованные открытыми ключами противоположных сторон. Обмен аутентифицирует способность каждой из сторон восстановить хэш-значение (доказывающую, что другая сторона расшифровала соответствующий одноразовый номер).

3.3.4. Аутентификация на основе шифрования открытыми ключами - модифицированный метод

В модифицированном методе аутентификации с использованием шифрования открытыми ключами исправляются некоторые недостатки стандартного метода. При применении этого метода в асимметричной системе выполняются только две вычислительно сложные операции шифрования/расшифровывания вместо четырех, а также имеется возможность передачи инициатором своего сертификата. Однако по-прежнему инициатору необходимо использовать какой-то внешний (по отношению к IKE) способ получения сертификата ответчика.

Схема обмена сообщениями в модифицированном методе имеет следующий вид:

```
 №   Инициатор                        Ответчик
(1)  HDR, SA                    =>
(2)                             <=  HDR, SA
(3)  HDR, (Ni_b)PubKey_r,
       (KE_b)ke_i, (IDi_b)ke_i
       [,(CERT_b)ke_i]          =>
(4)                             <=  HDR, (Ni_b)PubKey_e,
                                      (KE_b)ke_r, (IDr_b)ke_r
(5)   HDR*, HASH_I              =>
(6)                             <=  HDR*, HASH_R
```

В этом методе одноразовый номер по-прежнему зашифровывается открытым ключом другой стороны, однако идентификаторы (и сертификат, если он посылается) зашифровываются с помощью согласованного в первых двух сообщениях алгоритма симметричного шифрования ключами ke_i и ke_r, которые генерируются с использованием соответствующих одноразовых номеров. Кроме того, с помощью тех же самых симметричных ключей зашифровываются и блоки данных KE (обмена ключами). Это обеспечивает дополнительную защиту обмена Диффи-Хеллмана.

Опуская детали, вычисление симметричных ключей с использованием расшифрованных одноразовых номеров можно представить следующим образом:

```
ke_i = prf(Ni_b, CKY_I)

ke_r = prf(Nr_b, CKY_R)
```

Конкретные особенности вычисления этих ключей, связанные с длиной исходных данных и результата для псевдослучайной функции и их выравнивания, подробно рассмотрены в RFC 2409 [3].

3.4. Агрессивный режим

Основные отличия агрессивного режима (Aggressive Mode) от основного заключаются во вдвое меньшем числе пересылаемых сообщений, а также в

некоторых изменениях свойств: агрессивный режим предоставляет меньше возможностей по согласованию, а также не обеспечивает сокрытия идентификаторов участников (при некоторых методах аутентификации).

В агрессивном режиме уже в первом сообщении инициатор предлагает список защитных наборов, своё открытое значение Диффи-Хеллмана, свой одноразовый номер и свой идентификатор. Ответчик в своем первом сообщении указывает выбранный защитный набор, своё открытое значение Диффи-Хеллмана, свой одноразовый номер, свой идентификатор, а также блок данных с аутентифицирующей информацией. Для аутентификации заранее распределёнными ключами и для обоих методов аутентификации на основе шифрования открытым ключом аутентифицирующая информация передается в блоке данных HASH, для аутентификации на основе подписей – в блоке данных SIG. Третьим сообщением инициатор отправляет свой блок данных с аутентифицирующей информацией.

Возможности агрессивного режима изначально ограничены. Предоставляя своё открытое значение Диффи-Хеллмана в первом же сообщении, инициатор лишается возможности предложить разные группы Диффи-Хеллмана для разных защитных наборов. Если инициатор хочет использовать аутентификацию на основе шифрования открытым ключом, то это может быть его единственное предложение – для такого обмена одноразовый номер (блок данных Ni_b) должен быть зашифрован. Кроме того, если инициатор хочет использовать модифицированный метод аутентификации на основе шифрования открытыми ключами, то он вынужден ограничиться предложением единственного защитного набора с единственным выбором алгоритма шифрования или аутентификации.

Тем не менее, в определённых ситуациях агрессивный режим оказывается единственным возможным для реализации первой фазы. Рассмотрим, например, использование аутентификации на основе заранее распределённого ключа. Как было показано в предыдущем подразделе, при таком выборе метода аутентификации в основном режиме инициатор может идентифицировать себя лишь IP-адресом. В агрессивном режиме такого ограничения нет.

Следует также отметить, что при аутентификации на основе шифрования открытым ключом (в отличие от других методов аутентификации) агрессивный режим обеспечивает сокрытие идентификаторов участников.

В ситуациях, когда инициатору заранее известна политика ответчика (например, когда удалённый пользователь, соединяется со своим офисом), расширенные возможности основного режима по согласованию параметров устанавливаемого контекста безопасности оказываются избыточными, и агрессивный режим является предпочтительным из-за меньших накладных расходов.

Ниже представлены четыре схемы обмена в агрессивном режиме при разных методах аутентификации, которые, с нашей точки зрения, не требуют дополнительных пояснений.

3.4.1. Аутентификация заранее распределёнными ключами

```
 №   Инициатор                  Ответчик
(1)  HDR, SA, KE, Ni, ID_i =>
(2)                         <=  HDR, SA, KE, Nr, ID_r, HASH_R
(3)  HDR, HASH_I            =>
```

3.4.2. Аутентификация подписями

```
 №   Инициатор                  Ответчик
(1)  HDR, SA, KE, Ni, ID_i =>
(2)                         <=  HDR, SA, KE, Nr, ID_r, [CERT,]
                                                          SIG_R
(3)  HDR, [CERT,] SIG_I     =>
```

3.4.3. Стандартный метод аутентификации с шифрованием открытыми ключами

```
 №   Инициатор                  Ответчик
(1)  HDR, SA, [HASH,] KE,
       (IDi_b)PubKey_r,
       (Ni_b)PubKey_r       =>
(2)                         <=  HDR, SA, [HASH,] KE,
                                  (IDr_b)PubKey_i,
                                  (Nr_b)PubKey_i, HASH_R
(3)  HDR, HASH_I            =>
```

3.4.4. Модифицированный метод аутентификации с шифрованием открытыми ключами

```
 №   Инициатор                  Ответчик
(1)  HDR, SA, [HASH,]
     (Ni_b)PubKey_r,
     (KE_b)key_i,
     (IDi_b)key_i
     [, (CERTi_b)key_i)     =>
(2)                         <=  HDR, SA, [HASH,]
                                (Nr_b)PubKey_i,
                                (KE_b)key_r,
                                (IDr_b)key_r,
(3)  HDR, HASH_I            =>
```

3.5. Быстрый Режим

3.5.1. Схема обмена сообщениями в быстром режиме

Быстрый режим (Quick Mode) используется в ходе второй фазы работы ISAKMP/IKE для согласования и установки контекстов безопасности IPsec SA (AH SA и/или ESP SA). При этом согласуются криптографические параметры устанавливаемых контекстов безопасности и для каждого направления ключи, которые должны использоваться созданными IPsec SA, а также индексы

параметров безопасности (SPI), с помощью которых эти контексты безопасности второй фазы будут идентифицироваться. Напомним, что контексты безопасности AH SA и ESP SA являются однонаправленными. Схема обмена сообщениями в быстром режиме имеет следующий вид:

```
 №   Инициатор                      Ответчик
(1)  HDR*, HASH(1), SA,
      Ni[, KE ] [, IDi, IDr ]  =>
(2)                            <=  HDR*, HASH(2), SA,
                                    Nr[, KE ] [, IDi, IDr ]
(3)  HDR*, HASH(3)             =>
```

Как и в агрессивном режиме обмена, в быстром режиме используются три сообщения. В первом сообщении инициатор в блоке данных SA предлагает защитные наборы для IPsec SA, посылает свой одноразовый номер (Ni), предоставляет для аутентификации блок данных HASH(1), а также (в качестве необязательных элементов сообщения) блок данных обмена ключами (KE), и блоки данных идентификации (IDi, IDr). В ответе также должны присутствовать блоки SA, HASH(2), Nr, и могут присутствовать блоки KE, IDi и IDr. Третье сообщение быстрого режима инициатор посылает ответчику. В нём содержится блок данных HASH(3). Функции и применение всех этих блоков рассмотрены ниже. Заметим, что в быстром режиме для организации ответчиком подтверждения о завершении создания на его стороне контекста безопасности может использоваться дополнительное четвёртое сообщение (см. п. 3.5.6).

В результате представленного выше обмена будут созданы два контекста безопасности второй фазы (по одному в каждом направлении).

В ходе одного быстрого обмена могут быть согласованы сразу несколько контекстов безопасности и соответствующие ключи. В спецификации IKE [3] в качестве примера представлена схема такого обмена:

```
 №   Инициатор                      Ответчик
(1)  HDR*, HASH(1), SA0, SA1,
      Ni[, KE ] [, IDi, IDr ]  =>
(2)                            <=  HDR*, HASH(2), SA0, SA1,
                                    Nr[, KE ] [, IDi, IDr ]
(3)  HDR*, HASH(3)             =>
```

В данном примере передаются и согласуются два контекста безопасности (SA0 и SA1). В результате обмена будут созданы четыре контекста безопасности.

В некотором смысле обмен в быстром режиме не является полноценным (независимым) обменом – в нем (точно так же, как и в информационном обмене) для шифрования и аутентификации сообщений используется контекст безопасности IKE SA, созданный в первой фазе. Следует обратить внимание на эту особенность протокола IKE. В исходном протоколе ISAKMP, являющемся основой IKE, существовал набор равноправных обменов, каждый

из которых можно было употребить в любой из фаз. В протоколе IKE каждый из режимов может употребляться только в определённой фазе, и режимы второй фазы используют данные первой фазы.

Контекст безопасности IKE SA является двунаправленным, начать быстрый режим может любой из участников, вне зависимости от того, кто из них начал обмен первой фазы. Если в быстром режиме (во второй фазе) происходит смена инициатора (по сравнению с первой фазой), меняются лишь роли сторон, но не информация о ролях. То есть идентифицирующие цепочки инициатора и ответчика в заголовках сообщений второй фазы следуют в том же порядке, что и в первой фазе, вне зависимости от новых ролей участников, продолжая идентифицировать соответствующий IKE SA.

Все предложения, сделанные в ходе быстрого режима должны быть логически связанными и последовательными. Например, если посылается блок данных KE (обмен ключами), то атрибуты, описывающие группу Диффи-Хеллмана, должны быть включены в каждое преобразование каждого предложения каждого согласуемого контекста безопасности. Аналогично, если используется необязательная идентифицирующая информация (как в приведенном выше примере согласования двух контекстов безопасности в ходе одного быстрого режима), она должна быть применима к каждому из согласуемых контекстов безопасности.

3.5.2. Поле Message ID заголовка ISAKMP/IKE

На базе одного режима первой фазы (основного или агрессивного) может быть основано несколько обменов в быстром режиме второй фазы. Более того, под защитой одного контекста безопасности IKE SA одновременно может выполняться несколько обменов в быстром режиме. Для осуществления такого одновременного взаимодействия между одними и теми же участниками необходимо отличать сообщения, соответствующие одной первой фазе, но относящиеся к разным экземплярам обмена в быстром режиме. Для этой цели применяется поле Message ID (идентификатор сообщения) заголовка ISAKMP (см. приложение 3) – для каждого из обменов используется свой уникальный идентификатор сообщения. Кроме того, это поле применяется для создания уникального для каждого из экземпляров обмена вектора инициализации, используемого при шифровании и расшифровывании сообщений. Заметим также, что поле Message ID участвует в формировании кодов аутентификации сообщений быстрого режима, передаваемых в блоках данных HASH.

3.5.3. Шифрование и аутентификация сообщений быстрого режима

При шифровании в быстром режиме используется ключ, полученный из значения SKEYID_e первой фазы в соответствии с согласованным алгоритмом шифрования. Шифрование этим ключом обеспечивает конфиденциальность. В отличие от обменов первой фазы, все сообщения быстрого режима являются

зашифрованными (зашифровываются все блоки данных, а заголовок сообщения остаётся незашифрованным).

Аутентификация сообщений быстрого режима проводится с помощью псевдослучайной функции prf, применявшейся в первой фазе (IKE SA). В качестве ключа в функции prf используется значение SKEYID_a.

Для первого сообщения вторым аргументом функции prf является конкатенация идентификатора сообщения (Message ID) и оставшейся части этого сообщения:

HASH(1) = prf(SKEYID_a, M-ID | SA | Ni [| KE] [| IDi | IDr)

Значение блока данных HASH(1) обеспечивает целостность данных и аутентификацию источника первого сообщения.

Аналогично строится код аутентификации второго сообщения:

HASH(2) = prf(SKEYID_a, M-ID | Ni_b | SA | Nr [| KE] [| IDi | IDr)

Однако следует отметить, что при вычислении значения этого блока данных, помимо идентификатора сообщения, ответчик использует одноразовый номер инициатора (Ni_b), полученный в первом сообщении. Таким образом, значение блока данных HASH(2) обеспечивает целостность данных и аутентификацию источника второго сообщения, а также служит для инициатора доказательством того, что ответчик действительно в данный момент существует и является активным участником обмена.

Ответчику также необходимо доказательство того, что инициатор в данный момент действительно существует и является активным участником обмена – сообщение инициатора на самом деле могло бы быть сохранённым сообщением одного из старых диалогов между данными инициатором и ответчиком, которое злоумышленник может многократно воспроизводить, чтобы ответчик каждый раз создавал новый контекст безопасности, исчерпывая свои ресурсы. Именно для предотвращения такого рода атак в быстрый режим добавлено третье сообщение. В нём инициатор посылает аутентифицирующий блок данных HASH(3), при вычислении значения которого используются одноразовые номера (Ni_b и Nr_b) и идентификатор сообщения (Message ID):

HASH(3) = prf(SKEYID_a, 0 | M-ID | Ni_b | Nr_b)

Следует отметить, что в спецификации IKE в сообщениях быстрого режима устанавливается жесткий порядок только для блоков данных HASH и SA, которые должны следовать непосредственно за заголовком, а также для необязательных блоков данных ID. Поэтому, если порядок блоков данных в сообщениях быстрого режима отличается от приведенного выше на схеме обмена, или если к сообщению были добавлены какие-либо дополнительные необязательные блоки данных, например, блок данных Уведомление, то выражения для вычисления значений HASH(1) и HASH(2) должны быть соответствующим образом скорректированы.

3.5.4. Формирование ключевого материала и обеспечение совершенной прогрессирующей секретности

В быстром режиме ключевой материал для IPsec SA создаётся с использованием значения SKEYID_d, вычисленного в первой фазе. Ключ SKEYID_d используется в качестве аргумента псевдослучайной функции вместе с одноразовыми номерами, переданными в ходе обмена, а также со значениями полей "protocol" и "SPI" из предложений контекстов безопасности IPsec SA (см. приложение 3). Это гарантирует уникальность ключевого материала для разных контекстов безопасности, в том числе и для контекстов безопасности IPsec SA, отличающихся только направлением. Напомним, что в результате одного согласования IPsec SA получаются два контекста безопасности - входящий и исходящий. Им назначаются разные значения индексов параметров безопасности (SPI): одно выбирается инициатором, другое – ответчиком. Поэтому для разных контекстов безопасности будет создан разный ключевой материал, даже если эти контексты безопасности отличаются только направлением между одними и теми же участниками.

Поскольку все ключи получаются из одного источника, они оказываются зависимыми друг от друга. Если злоумышленникам удастся определить значение SKEYID_d, то все ключи будут скомпрометированы. Таким образом, базовый быстрый режим не обладает свойством совершенной прогрессирующей секретности (PFS – perfect forward secrecy).

Тем не менее, существует возможность обеспечить совершенную прогрессирующую секретность в быстром режиме. Для этого участники с помощью блоков данных KE (обмен ключами) проводят дополнительный обмен Диффи-Хеллмана, результат которого используется при создании ключевого материала для IPsec SA. При этом в быстром режиме согласование группы Диффи-Хеллмана невозможно по тем же причинам, что и в агрессивном режиме: и открытое значение Диффи-Хеллмана (в блоке данных KE), и защитный набор (в составе блока SA) предлагаются инициатором в одном и том же первом сообщении, когда заранее неизвестно, какой защитный набор будет выбран ответчиком.

Если необходимости в совершенной прогрессирующей секретности нет, и обмен блоками данных KE не выполнялся, новый ключевой материал определяется следующим образом:

KEYMAT = prf(SKEYID_d, protocol | SPI | Ni_b | Nr_b).

Если же для обеспечения совершенной прогрессирующей секретности стороны осуществили дополнительный обмен Диффи-Хеллмана в быстром режиме, то новый ключевой материал определяется так:

KEYMAT = prf(SKEYID_d, $g(qm)^{xy}$ | protocol | SPI | Ni_b | Nr_b),

где $g(qm)^{xy}$ – общий секрет, полученный в результате дополнительного обмена Диффи-Хеллмана. Детали таких вычислений приведены в спецификации RFC 2409 [3].

Этот ключевой материал используется для формирования ключей соответствующего сервиса безопасности (AH или ESP). Правила формирования ключей из ключевого материала определены в соответствующих спецификациях (RFC 2402 [7] и RFC 2406 [8]).

Заметим, что после вычисления ключевого материала само значение общего секрета Диффи-Хеллмана ($g(qm)^{xy}$) из текущего состояния должно безусловно стираться. Значения SKEYID_e и SKEYID_a (полученные в процессе согласования первой фазы) продолжают защищать и аутентифицировать IKE SA, а значение SKEYID_d продолжает использоваться для формирования ключевого материала для других контекстов безопасности второй фазы.

3.5.5. Обмен селекторами трафика

Как уже отмечалось в разд. 2, селекторы трафика позволяют участникам взаимодействия установить определенные ограничения на трафик, который пересылается под защитой конкретного контекста безопасности второй фазы. Если, например, контекст безопасности устанавливается для обеспечения защиты взаимодействия между узлами по протоколу Telnet, то он может использоваться только для этого вида трафика. Весь другой трафик между этими узлами защищаться данным контекстом безопасности не будет.

Информация о селекторах может передаваться инициатором в первом сообщении быстрого режима вместе с предложениями соответствующих защитных наборов для данного контекста безопасности в необязательных блоках данных Идентификация (IDi, IDr). Формат этих блоков определяется спецификацией RFC 2409 [3]. Заметим, что допустимые типы идентификаторов для селекторов трафика определяются в спецификации RFC 2407 "IPsec DOI" [1]. Этот документ как раз и предназначен для описания возможных значений конкретных параметров обмена второй фазы. В частности, в качестве селекторов трафика могут использоваться IP-адреса (одиночный IP-адрес, диапазон IP-адресов или адрес подсети, задаваемый IP-адресом и маской), тип протокола (поле заголовка IP, в котором указывается номер протокола UDP, TCP и т.д.), номера портов TCP/UDP.

В случае успешной проверки на соответствие своей собственной политике ответчик сохраняет эту информацию в своей базе контекстов безопасности SAD вместе с выбранными им параметрами защитного набора и использует ее в дальнейшем для контроля трафика.

3.5.6. Возможность дополнительного подтверждения в быстром режиме

Следует рассмотреть следующий нюанс. После получения второго сообщения от ответчика инициатор имеет всю информацию, достаточную для создания контекста безопасности. Однако, отослав сообщение (2), ответчик не может создавать контекст безопасности, поскольку до того, как будет получено сообщение (3), у него нет уверенности в том, что инициатор действительно существует и является активным участником обмена. Поэтому существует вероятность того, что инициатор начнёт посылать пакеты (под защитой только что согласованного контекста безопасности IPsec) до того, как ответчик на своей стороне создаст соответствующий контекст безопасности. Такая ситуация может возникнуть, например, когда пакеты IPsec по тем или иным причинам "обогнали" третье сообщение быстрого обмена. Таким образом, пакеты IPsec, пришедшие до того, как ответчик создал контекст безопасности, могут быть отброшены. Если защищался TCP-трафик, это не страшно – пакет TCPsyn будет посылаться заново после истечения тайм-аута. Но если защите подвергался UDP-трафик, то часть его будет потеряна.

Чтобы предотвратить такой сценарий, в IKE используется специальный бит подтверждения (commit bit) в поле Flags заголовка сообщения ISAKMP (см. приложение 3); это позволяет продлить быстрый режим на одно сообщение. Если любой из участников обмена установил бит подтверждения, ответчик обязан послать заключительное (четвёртое) сообщение, включающее аутентифицирующий блок данных HASH, за которым следует блок данных Уведомление, содержащий сообщение "connected". Инициатор не добавляет контекст безопасности в свою базу данных контекстов безопасности (SAD) и, очевидно, не начинает передачу по нему защищенного трафика до тех пор, пока не получит сообщения "connected".

3.6. Информационный обмен

Основной, агрессивный и быстрый режимы предназначены для согласования и, в конечном итоге, создания контекстов безопасности. В отличие от них, информационный обмен (Informational Exchange) служит, в основном, для обслуживания уже существующих контекстов безопасности. Этот обмен позволяет сторонам посылать друг другу уведомления о состоянии и об ошибках. Поскольку сообщения ISAKMP/IKE пересылаются внутри UDP-дейтаграмм, и нет гарантии того, что сообщение информационного обмена благополучно дошло, а подтверждение этим обменом не предусмотрено, информационный обмен изначально ненадёжен. В силу этих обстоятельств сообщения информационного обмена являются необязательными; в принципе реализации могут их и не посылать, хотя такое поведение не приветствуется.

Как правило, информационный обмен используется для информирования партнера о том, что какой-то из контекстов безопасности (IKE SA или IPsec SA) был удалён; соответственно, необходимо удалить его и на другом конце.

Ниже представлена схема информационного обмена:

```
№   Инициатор                  Ответчик

(1)   HDR*, HASH(1), N/D     =>
```

Как и для быстрого режима, сообщения информационного обмена имеют уникальный идентификатор сообщения (Message ID) и создают уникальный вектор инициализации для шифрования (обмену первой фазы может соответствовать много быстрых режимов или информационных обменов). Код аутентификации сообщения передается в блоке данных HASH:

```
HASH(1) = prf(SKEYID_a, M-ID | N/D)
```

Здесь N/D – это либо блок данных Notify (Уведомление), либо блок данных Delete (Удаление) (см. приложение 3).

Если возникает необходимость провести информационный обмен в тот момент, когда контекст безопасности IKE SA ещё не создан, то обмен выполняется в открытом виде без сопровождающего блока данных HASH.

3.7. Режим новой группы

Режим новой группы (New Group Mode), как и быстрый режим, является в IKE новым (по сравнению с ISAKMP). Режим новой группы не относится ни к одной из фаз. Задачей служебного обмена в режиме новой группы является согласование между участниками взаимодействия частной группы Диффи-Хеллмана (отличной от четырех стандартных групп, определённых в спецификации IKE). Поскольку происходит обмен информацией, критической с точки зрения безопасности, он должен быть защищен с помощью контекста безопасности IKE SA. Поэтому режим новой группы может выполняться только после завершения первой фазы. Так же, как и быстрый режим с информационным обменом, данный режим защищается с помощью шифрования с использованием ключа SKEYID_e и аутентифицируется значением хэш-функции, управляемой ключом SKEYID_a.

В режиме новой группы выполняется обмен типа запрос-ответ, в котором инициатор посылает блок данных SA (контекст безопасности), содержащий характеристики предлагаемой группы. Если ответчик согласен использовать данную группу, он отвечает сообщением с теми же блоками данных. Если группа неприемлема, то ответчик должен ответить блоком данных Уведомление, содержащим сообщение "attributes-not-supported".

Ниже приведена схема режима новой группы:

```
№   Инициатор                    Ответчик
(1)  HDR*, HASH(1), SA       =>
(2)                          <=  HDR*, HASH(2), SA
```


Значения HASH(1) и HASH(2) вычисляются следующим образом:

```
HASH(1) = prf(SKEYID_a, M-ID | SA),
HASH(2) = prf(SKEYID_a, M-ID | SA).
```

4. Недостатки протокола ISAKMP/IKE

После публикации в 1999 году спецификации IPsec, она была подвергнута всестороннему анализу. В этом разделе будут коротко рассмотрены два наиболее важных критических обзора [9,10]. Эти работы цитируются чаще всего. В них указаны недостатки протокола ISAKMP/IKE, проанализированы основные его проблемы, намечены способы решения этих проблем в ходе развития протокола. Рассматриваемые работы сыграли важную роль в разработке следующей версии протокола IKE - IKEv2.

4.1. Обзор Шнайера и Фергюсона

В 1999 году известные специалисты по криптографической защите данных Брюс Шнайер (Bruce Schneier) и Нильс Фергюсон (Niels Ferguson) опубликовали критический обзор протокола IPsec "A Cryptographic Evaluation of IPsec" [9]. Хотя предметом их исследования был протокол IPsec в целом, значительная часть обзора относится к протоколам ISAKMP и IKE.

Главным недостатком IPsec вообще и ISAKMP с IKE, в частности, авторы считают чрезмерную сложность, причина которой состоит в избыточной гибкости протокола и излишнем количестве всевозможных особенностей. Они утверждают, что сложность – это главный враг безопасности. Сложные системы содержат больше ошибок; кроме того, чем сложнее система, тем более замысловатые ошибки она содержит. При этом количество ошибок и трудоёмкость экспертизы системы растут значительно быстрее, чем ее сложность. Шнайер и Фергюсон полагают, что семейство протоколов IPsec настолько сложно, что его уязвимости невозможно надёжно проанализировать современными методами. Бóльшая часть упрёков в чрезмерной сложности относится к лишним, с точки зрения авторов обзора, протоколу AH и транспортному режиму IPsec; однако Шнайер и Фергюсон также утверждают, что избыточно сложны и ISAKMP и IKE. С их точки зрения:

- Обмен с аутентификацией без шифрования (Authentication Only Exchange) протокола ISAKMP не имеет практического применения и лишён смысла.
- Сомнительна попытка ограничить атаки на доступность с помощью идентифицирующих цепочек. Сложность протокола увеличивается, а механизм идентифицирующих цепочек предоставляет лишь частичное решение проблемы. Защита от сложных разновидностей атаки не имеет смысла, если отсутствует защита от ее простых разновидностей.

- Обмен с сокрытием идентификаторов участников (Identity Protection Exchange) протокола ISAKMP может быть осуществлён за четыре сообщения вместо шести.
- Основной режим протокола IKE (реализация обмена с сокрытием идентификаторов участников ISAKMP) также может поддерживаться с помощью меньшего количества сообщений.
- Стандартный метод аутентификации с открытыми ключами является частным случаем модифицированного метода, и поэтому для упрощения протокола IKE без него можно обойтись.

Шнайер и Фергюсон подвергают самой безжалостной критике документацию IPsec. Эпиграфом к данной статье являются их слова о том, что неразумно считать, что кто-либо способен изучить IPsec на основании только документации. По мнению авторов обзора, набор спецификаций семейства протоколов IPsec очень труден для понимания; часто отсутствует введение или обзор спецификации, из-за чего читателю приходится в уме сопоставлять все разделы документации, чтобы составить общее впечатление о предмете чтения. Более того, часто в спецификациях содержатся различные ошибки, противоречия, тогда как необходимые пояснения нередко опущены.

В частности, в документации IPsec не определяется, каких целей должен достигать протокол. Хотя некоторые из задач, стоящих перед IPsec, более или менее очевидны, читателю часто приходится по спецификации самому определять задачи, стоявшие перед разработчиками стандарта. Кроме того, ни в одной из спецификаций, входящих в состав IPsec, не предоставляется какое-либо обоснование выбора тех или иных решений. Отсутствие информации о целях, которые разработчики протокола хотели достигнуть, и мотивировки выбранных решений является серьёзным недостатком документации IPsec.

По мнению авторов критики:

- Документация ISAKMP находится в особенно неудовлетворительном состоянии. Текст сложен для понимания и часто содержит повторы.
- Ряд необходимых деталей не специфицирован. Например, спецификация RFC 2408 не отвечает на вопрос, в каком из блоков данных Transform (Преобразование) должны находиться атрибуты, относящиеся к контексту безопасности в целом (одному блоку Security Association может соответствовать несколько предложений, каждому из которых может соответствовать несколько преобразований).
- В документации в явном виде не упоминается неравноправие блоков данных. Из перечисления всех блоков и описания их свойств неявно следует, что блоки Предложение и Преобразование являются нетипичными блоками данных, фактически подблоками данных. Они не встречаются в сообщении сами по себе. Эти блоки интерпретируются как содержимое блока Контекст Безопасности,

синтаксически вынесенное в отдельные блоки данных. Наличие поля Next Payload (следующий блок данных), в котором иногда содержится тип следующего, а иногда совсем не следующего блока данных, только усложняет понимание структуры пакета ISAKMP/IKE.
- Согласно спецификации RFC 2408, блок данных HASH в соответствии со своим названием хранит результат применения согласованной хэш-функции к некоторым полям сообщения ISAKMP и/или данным состояния протокола (ключам, временным данным и т.п.). В спецификации утверждается, что этот блок данных, в частности, используется для аутентификации сторон, участвующих в обмене. Авторы обзора отмечают, что само по себе значение хэш-функции не может аутентифицировать кого-либо. Скорее всего, под значением, передаваемым в блоке данных HASH, разработчики ISAKMP подразумевали код аутентификации сообщения (MAC – Message Authentication Code), который, действительно, используется и для проверки целостности, и для аутентификации сообщений. В документации IPsec вообще путаются понятия кода аутентификации сообщения и профиля или дайджеста сообщения (каноническое определение этих понятий дано в приложении 2).
- В спецификации IKE RFC 2409 также неправильно употребляется термин HASH. Величины HASH, используемые при аутентификации, на самом деле представляют собой коды аутентификации сообщений (MAC).
- Авторы считают, что разбиение спецификации IKE на отдельные документы: RFC 2408 (ISAKMP), RFC 2407 (DOI) и RFC 2409 (IKE) привело к неудовлетворительному результату. Итоговая спецификация содержит множество ссылок из одного документа в другой и сложна для понимания.

Шнайер и Фергюсон отмечают ряд недостатков в строении пакета ISAKMP. Стандартный заголовок каждого блока данных содержит поле Next Payload. Поскольку в заголовке пакета содержится общая длина пакета, программе синтаксического анализа известно, имеются ли ещё блоки данных. Было бы проще, если бы в заголовке каждого блока данных хранился его собственный тип. В поле Next Payload блока Контекст Безопасности не могут указываться блоки Предложение или Преобразование, поэтому вообще отсутствует указание о типе первого подблока. Предполагается, что реализации знают, что первым подблоком будет блок Предложение. Но тогда возникает вопрос, зачем вообще нужно поле Next Payload?

Протокол IKE позволяет согласовать базовые криптографические функции, в частности, хэш-функцию и псевдослучайную функцию prf. Однако для этих функций спецификация не задаёт никаких ограничений, не сообщается, какие свойства ожидаются от этих функций. В стандартном определении хэш-функции требуется, чтобы она была устойчивой к коллизиям. То есть предполагается, что невозможно найти два аргумента m1 и m2, таких, что

H(m1) = H(m2), где H – рассматриваемая хэш-функция. Авторы обзора приводят пример, демонстрирующий, что для практических хэш-функций этого требования может быть недостаточно. В спецификации должны быть явным образом перечислены все необходимые требования к используемым хэш-функциям.

Кроме того, в спецификации ISAKMP/IKE не определяются конкретные функции prf, требуется управляемая ключом псевдослучайная хэш-функция (keyed pseudorandom function). По умолчанию в качестве функции prf предполагается использование согласованной хэш-функции (алгоритма вычисления дайджеста сообщения) в конструкции алгоритма вычисления кода аутентификации сообщения (MAC) – такой алгоритм получил название HMAC (keyed-hash message authentication code). В статье Шнайера и Фергюсона показано, что из некоторых хэш-функций, используемых при вычислении HMAC, может получиться плохая функция prf, которая в ряде ситуаций позволит злоумышленнику определить SKEYID. Использование в алгоритме HMAC произвольной хэш-функции, пусть и устойчивой к коллизиям, является уязвимостью протокола. Такие детали, которые разработчиками стандарта подразумевались, должны быть явно отражены в спецификации.

В рассматриваемой статье указывается также на ряд уязвимостей протоколов ISAKMP и IKE, предложены различные виды атак на эти протоколы. В качестве примера можно привести атаку отражения (reflection attack) на протокол IKE. Вычисления значений HASH_I и HASH_R симметричны относительно перемены данных инициатора и ответчика местами. Это позволяет злоумышленнику объявить себя ответчиком – CKY_I выбирается в качестве CKY_R, g^{xi} – в качестве g^{xr}, идентификатор инициатора используется ответчиком. Тогда значение HASH_R будет равно значению HASH_I, и поэтому ответчик будет посылать обратно значение, присланное инициатором.

Таким образом, общее впечатление авторов от IPsec является двойственным: признавая, что IPsec – это, возможно, лучший из доступных протоколов безопасности (на момент написания статьи), Шнайер и Фергюсон разочарованы несоответствием результата усилиям, затраченным на разработку протокола.

4.2. Статья Перлман и Кауфмана

В опубликованной в 2001 году статье Радиа Перлман (Radia Perlman) и Чарли Кауфмана (Charlie Kaufman) "Analysis of IPsec Key Exchange Standard" [10] описываются назначение и история протокола IKE, содержится анализ этого протокола. Эта статья заслуживает внимания сама по себе. Однако то, что Чарли Кауфман стал разработчиком опубликованной в декабре 2005 года спецификации следующей версии протокола (IKEv2) [11], заставляет обратить на статью особое внимание. Словно отвечая на упрёк, содержащийся в обзоре

Шнайера и Фергюсона, авторы статьи рассматривают значение протокола IPsec, сравнивают его с протоколом SSL (TLS).

Анализу интересующего нас протокола IKE посвящена большая часть статьи. Перлман и Кауфман также утверждают, что IKE – это невероятно сложный протокол. По их мнению, это является следствием неправильной политики разработчиков и специфики разработки стандарта комитетом, включающим большое число участников. Шнайер и Фергюсон утверждали буквально то же самое. По мнению авторов статьи, именно в силу необычайной сложности протокола, а также из-за запутанности документации, практически не было значительных обзоров IKE. Рассмотрим последовательно основные замечания авторов статьи.

4.2.1. Наличие второй фазы IKE

Перлман и Кауфман подвергают сомнению необходимость реализации второй фазы. После вынужденно трудоёмкой (включающей вычисления с открытыми ключами) первой фазы проводится вторая фаза, которая может быть осуществлена значительно проще при использовании сеансового ключа, созданного под защитой первой фазы. Такое разделение имеет смысл, только если одному обмену первой фазы будет соответствовать несколько обменов второй фазы, что на практике встречается достаточно редко.

Аргументы авторов статьи сводятся к следующему:

- Полезным свойством криптографического протокола является возможность периодического обновления ключей. Обновление ключей в ходе второй фазы обходится дешевле, чем обновление их в ходе обмена, который основывается на криптографии с открытым ключом, вынесенной в первую фазу. Однако если мы хотим, чтобы в ходе обновления ключей соблюдалась совершенная прогрессирующая секретность, то трудоёмкость второй фазы становится незначительно меньше трудоёмкости первой фазы. А если мы просто обновляем ключи для того, чтобы ограничить объём данных, переданных под защитой одного ключа, или для того, чтобы использовать новый ключ после переполнения счётчика защиты от повторного воспроизведения сообщений, то протокол, разработанный специально для обновления ключей, может быть проще и дешевле.
- Можно создать несколько соединений, обладающих различными свойствами безопасности. Например, одно соединение будет предоставлять только защиту целостности, другое – шифрование коротким ключом, третье – шифрование длинным надёжным ключом. Необходимость поддержки таких возможностей весьма сомнительна. Несколько соединений с разными свойствами безопасности между одними и теми же участниками могут быть реализованы с помощью отдельных не связанных между собой (общей первой фазой) контекстов безопасности.

- Между двумя узлами может потребоваться установление нескольких соединений из-за того, что эти соединения относятся к разным взаимодействующим приложениям, и каждое из приложений хочет использовать свой ключ, взаимодействуя с IPsec. Этот случай является достаточно редким. Установление контекстов безопасности, не связанных общей первой фазой, позволяет решить и эту проблему.

4.2.2. Аутентификация открытыми ключами в первой фазе работы IKE

Описывая работу первой фазы протокола IKE, авторы отмечают, что из-за наличия двух режимов и четырёх методов аутентификации существует восемь различных способов осуществить первую фазу. Перлман и Кауфман перечисляют причины выделения аутентификации заранее распределёнными ключами и аутентификации цифровой подписью, но подвергают сомнению необходимость наличия разновидностей основного и агрессивного обмена, основанных на аутентификации открытыми ключами.

4.2.3. Идентифицирующие цепочки

Перлман и Кауфман подвергают критике внедрённый в IKE механизм идентифицирующих цепочек. Они описывают реализацию идентифицирующих цепочек в протоколах Photuris и Oakley и задаются вопросом, почему механизм идентифицирующих цепочек протокола IKE, разработанного позже, уступает более ранним аналогам.

Основная задача идентифицирующих цепочек состоит в предотвращении атак на доступность (DOS – denial of service). У узла, на котором запущен IKE (назовём его сервером), имеются ограниченные память и вычислительные мощности. Злоумышленник инициирует большое количество соединений со случайными IP-адресами отправителя или с разных IP-адресов, стараясь вызвать исчерпание ресурсов сервера. В протоколе Photuris сервер не будет создавать состояния или производить какие-либо значительные вычисления до тех пор, пока не получит от инициатора обмена значение идентифицирующей цепочки (cookie) – комбинацию IP-адреса инициатора и секрета, известного серверу (см. подраздел 2.4).

Механизм идентифицирующих цепочек обеспечивает защиту от некоторых разновидностей атак на доступность. Для поддержки этого механизма в протоколе Photuris требуется посылка двух дополнительных сообщений в начале обмена. В протоколе Oakley идентифицирующие цепочки не обязательны. Обмен может проводиться без них.

Несмотря на то, что IKE был разработан гораздо позже протоколов Photuris и Oakley, в нём сервер вынужден создавать состояние сразу же при получении сообщения от инициатора обмена, когда инициатор ещё не продемонстрировал знание идентифицирующей цепочки, сформированной для него сервером. Перлман и Кауфман утверждают, что механизм Photuris и

Oakley мог бы быть внедрён в IKE, даже без необходимости тратить два лишних сообщения на согласование идентифицирующих цепочек.

4.2.4. Сокрытие идентификаторов участников

В основном режиме IKE реализуется обмен с сокрытием идентификаторов участников протокола ISAKMP. В какой мере в этом режиме выполняется свойство сокрытия участников? В статье подробно анализируется эта проблема и предлагаются способы значительного усиления свойства сокрытия идентификаторов участников (hiding endpoints identities или identity protection) в первой фазе работы протокола IKE. Авторы рассматривают пассивные и активные атаки на IKE.

В общем случае агрессивный режим IKE не обеспечивает защиты идентификаторов участников. Однако для агрессивного режима с аутентификацией подписями незначительное изменение протокола могло бы обеспечить защиту идентификаторов обеих сторон от пассивных атак, а идентификатора инициатора даже от активных атак. С этой целью идентификаторы сторон могли бы быть перемещены из сообщений 1 и 2 в сообщения 2 и 3 соответственно и зашифрованы с помощью секретного ключа Диффи-Хеллмана (см. схему агрессивного режима с аутентификацией подписями в подразделе 3.4).

В основном режиме с аутентификацией подписями число сообщений может быть сокращёно с шести до пяти (см. схему оригинального обмена в подразделе 3.3). Эта разновидность основного режима защищает идентификаторы обоих участников от пассивных атак, а также защищает идентификатор ответчика от активной атаки со стороны инициатора. При этом атакующий злоумышленник, изображающий из себя настоящего ответчика, может провести обмен Диффи-Хеллмана с инициатором и получить идентификатор инициатора в пятом сообщении. Авторы статьи считают, что защищать идентификатор инициатора важнее, чем защищать идентификатор ответчика: для многих ситуаций ответчик будет иметь постоянный IP-адрес, так что информация о нём и так доступна. Поэтому они предлагают изменить данную разновидность обмена, поместив информацию из шестого сообщения в четвёртое. В результате обмен сокращается на одно сообщение, а вместо идентификатора ответчика защищается идентификатор инициатора.

Обмены первой фазы с аутентификацией открытыми ключами предоставляют защиту как против пассивных атак по раскрытию идентификаторов участников, так и против активных атак. Однако, как указывалось выше, Перлман и Кауфман рекомендуют вообще отказаться от этих типов обмена.

Основной режим с аутентификацией заранее распределёнными ключами не предоставляет никакой защиты против раскрытия идентификаторов сторон (см. полную схему оригинального обмена в подразделе 3.3). Может показаться, что идентификаторы обоих участников защищены, поскольку они зашифрованы. Однако после отправления четвёртого сообщения ответчик ещё

не знает, с кем он ведёт диалог, а ключ, которым зашифрованы сообщения 5 и 6, является производным, в частности, от заранее распределённого ключа. Поэтому ответчик не в состоянии расшифровать пятое сообщение (в котором будет содержаться идентификатор инициатора), если он заранее не знает этот идентификатор. Это является причиной того, что в соответствии со спецификацией IKE в качестве идентификаторов в этом типе обмена могут быть использованы только IP-адреса. Но тогда и пассивный атакующий злоумышленник знает эти идентификаторы.

В спецификации утверждается, что такой обмен может быть полезен в тех сценариях, в которых мобильный пользователь связывается с корпоративным сервером. Авторы критики указывают, что в этом случае подобный обмен был бы бесполезен, поскольку IP-адрес инициатора может быть переменным. Ими предлагаются способы улучшения рассматриваемого обмена.

4.2.5. Согласование параметров контекстов безопасности

Авторы статьи подвергают также сомнению оптимальность существующего механизма согласования параметров контекстов безопасности. Протокол IKE позволяет осуществить максимально гибкое согласование, когда инициатор предлагает список конкретных защитных наборов, свойства которых его удовлетворяют. При этом полное перечисление всех возможных защитных наборов может быть очень длинным. Альтернативой могло бы быть указание отдельных свойств (метод шифрования, метод аутентификации, хэш-функция), которые удовлетворяют сторону обмена. Защитные наборы получались бы как множества всех комбинаций этих отдельных свойств. Альтернативный способ был бы более экономным, хотя и не предоставлял бы всей мощи способа, реализованного в IKEv1.

4.2.6. Другие замечания

Авторы статьи указывают, что востребованной является возможность односторонней аутентификации (когда только у одной из сторон есть криптографический идентификатор). Они полагают, что такая возможность была бы полезной для IKE. Перлман и Кауфман указывают также на опасность криптографически слабых заранее распределённых ключей (weak pre-shared secret keys) и предлагают способы модификации IKE, позволяющие сделать протокол защищённым от атаки по словарю (dictionary attack).

5. Протокол IKEv2

5.1. Переход от ISAKMP/IKE к IKEv2

После утверждения спецификаций RFC2407, RFC2408 и RFC2409 протокола ISAKMP/IKE в ноябре 1998 года протокол IKE стал повсеместно использоваться. Трудности реализации, связанные, прежде всего, с его чрезмерной сложностью, и недостатки, выявленные в процессе критического

анализа, послужили причиной продолжения исследований для создания более эффективных средств автоматического управления контекстами безопасности и ключами.

В техническом комитете проектирования Интернет (Internet Engineering Task Force, IETF) началось обсуждение нового протокола. Были сформулированы требования к такому протоколу, условно названному "сын IKE" (SOI – son-of-IKE). Ниже перечислены некоторые из этих требований (в соответствии с черновым вариантом спецификации draft-ietf-ipsec-sonofike-rqts-00):

- упрощение протокола;
- сведение документации по протоколу в единый документ;
- расширяемость; поддержка новых блоков данных, полей, обменов и т.п., возможность распознавания новых возможностей;
- обработка ситуаций, когда один из участников обмена теряет информацию о состоянии из-за перезагрузки или зависания;
- взаимодействие с устройствами трансляции сетевых адресов (NAT);
- сходимость протокола; специфицирование и документирование всех особенностей взаимодействия, в частности, обработки ошибок, служебных сообщений, механизма обновление ключей, средств обнаружения потери связи с партнёром;
- устойчивость к атакам на доступность (DOS attacks); реализация бесконтекстных идентифицирующих цепочек (stateless cookies), недоверие к не аутентифицированным сообщениям;
- устойчивость к атакам повторного воспроизведения сообщений (replay attacks) и атакам, направленным на деградацию производительности узлов сети (downgrade attacks).

Основное внимание IETF уделил проектам JFK (just fast keying) и IKEv2, хотя имелись и другие предложения, например, протокол IKE-SIGMA. В течение длительного времени оба направления развивались параллельно, у каждого из них было много сторонников, и рабочая группа IPSEC не могла определиться с выбором.

В 2002 году продолжалось активное обсуждение обоих вариантов протокола (в частности, велась рассылка материалов рабочей группы, серии черновиков спецификаций draft-ietf-ipsec-jfk- и draft-ietf-ipsec-ikev2- для JFK и IKEv2 соответственно, а также черновиков спецификаций draft-ietf-ipsec-soi-features). Некоторые изменения исходного протокола были приняты обеими сторонами. Важнейшими из таких изменений стали следующие:

- оптимизация обмена – возможность проведения согласования контекстов безопасности за четыре сообщения;
- значительное общее упрощение протокола по сравнению с исходным протоколом ISAKMP/IKE;
- сведение спецификации в единый документ.

Если попытаться сформулировать различия между протоколами несколькими словами, то можно сказать, что JFK – это совершенно новый протокол, не связанный с ISAKMP/IKE, в котором авторы пытались добиться максимального упрощения протокола, нередко за счёт ограничения функциональности. Протокол IKEv2 скорее представляет собой развитие или модификацию протокола ISAKMP/IKEv1. В отличие от авторов JFK, разработчики IKEv2 пытались найти баланс между простотой, гибкостью и большей функциональностью. Рассмотрим более подробно некоторые различия между этими протоколами.

5.1.1. Количество фаз

В IKEv2 воспроизводится идея ISAKMP/IKE о двухфазной организации протокола. В первой фазе создается управляющий контекст или "защищенный канал" для обмена сообщениями второй фазы, который используется для создания новых контекстов безопасности IPsec SA, удаления старых или обновления ключей для них, а также для передачи управляющих или информационных сообщений. Одному IKE SA может соответствовать несколько IPsec SA.

В протоколе JFK вторая фаза отсутствует. Если нужно создать новый IPsec SA, предлагается использовать JFK "с чистого листа", необходимость обновлять ключи подвергается сомнению (зачем обновлять, если можно создать новый контекст безопасности), передача защищенных уведомлений не поддерживается, а обнаружение нарушения соединения между сторонами осуществляется путем замера времени поступления пакетов.

5.1.2. Управление трафиком

В протоколе IKEv2 в качестве способа обновления ключей, удаления контекстов безопасности, обнаружения обрыва соединения предлагается использовать сообщения, передаваемые в ходе второй фазы. В протоколе JFK для каждой из этих целей используются специальные методы. Например, в JFK удаление IPsec SA осуществляется с помощью обмена для создания нового IPsec SA с нулевым временем жизни. Обнаружение обрыва соединения предлагается проводить с использованием внешних механизмов (таких как ICMP echo).

5.1.3. Согласование параметров контекста безопасности

Авторы IKEv2 предлагают использовать традиционный метод: инициатор предлагает на выбор список алгоритмов, а ответчик указывает устраивающий его защитный набор. То есть инициатор предлагает альтернативы, а ответчик выбирает. Авторы JFK пошли по другому пути. В этом протоколе инициатор предлагает конкретный защитный набор, а ответчик либо с ним соглашается, либо от него отказывается.

5.1.4. Методы аутентификации

В протоколе JFK поддерживается единственный метод аутентификации – аутентификация открытыми ключами, то есть аутентификация на основе сертификатов. В протоколе IKEv2 дополнительно поддерживается аутентификация заранее распределёнными ключами.

5.1.5. Защита от атак на доступность

В качестве механизма защиты от некоторых видов атак на доступность, реализующих затопление целевого узла запросами инициализации сеанса IKE с поддельных IP-адресов, в обоих протоколах предлагается использовать идентифицирующие цепочки (cookie). Различие заключается в том, что в JFK для проведения обмена всегда используются четыре сообщения, и отсутствует стремление к сохранению информации о состоянии.

В IKEv2 для снижения эффективности подобного рода атак предлагается другое решение. В обычных условиях ответчик начинает создавать на своей стороне состояние сразу после получения первого сообщения от инициатора сеанса, при этом весь обмен реализуется за четыре сообщения. Однако при обнаружении большого количества наполовину открытых контекстов безопасности IKE (что является признаком указанной выше атаки) ответчик должен отбрасывать все начальные сообщения IKE, если они не содержат специального блока данных Уведомление типа COOKIE. При этом в качестве ответа инициатору ответчик должен послать незащищенное сообщение IKE и включить в него блок данных Уведомление типа COOKIE с данными идентифицирующей цепочки, которые должны быть возвращены. В свою очередь, инициаторы, получающие такие ответы, должны повторить свои начальные сообщения, включив в них блоки данных Уведомление типа COOKIE, содержащие предоставленные ответчиком данные соответствующих идентифицирующих цепочек. Заметим, что в этом случае каждый обмен удлиняется на два сообщения. При этом предложенный способ генерации ответчиком идентифицирующих цепочек не требует сохранения какого-либо состояния для их последующего распознавания при получении повторных начальных сообщений и обеспечивает невозможность их подделки злоумышленником.

5.1.6. Организация пакетов

Авторы протокола IKEv2 предлагают использовать синтаксис, схожий с ISAKMP/IKE. Добавляются новые блоки данных, в некоторой степени модифицируется синтаксис передачи зашифрованных данных, меняются некоторые блоки данных. В протоколе JFK сообщения организуются совершенно иначе. У блоков данных имеется значительно более простая структура; многие поля из ISAKMP/IKE, IKEv2, такие как номер версии в заголовке, не используются.

В конце концов, предпочтение было отдано IKEv2. Следует отметить, что протокол, предложенный Чарли Кауфманом, изначально был назван IKEv2, но только впоследствии он фактически стал второй версией IKE, оправдав своё название. Доработки IKEv2 продолжались еще в течение более двух лет. В декабре 2005 года серия черновиков стандартов, включая черновик спецификации IKEv2, получила статус спецификаций RFC 4301-4308, заменив предыдущее поколение спецификаций IPsec (RFC 2401-2409).

5.2. Обзор протокола IKEv2

Рассмотрим общую схему работы протокола IKEv2. Заметим, что в данной статье не ставилась цель предоставить полный и всесторонний обзор протокола IKEv2. Во-первых, IKEv2 – это новый протокол, его реализации не распространены, и рассмотрение IKEv2 служит здесь лишь цели краткого сравнения с IKE. Во-вторых, в отличие от ISAKMP/IKE, спецификация IKEv2 является хорошим самостоятельным источником информации по протоколу. Материалы данного подраздела носят обзорный характер, описывая протокол IKEv2 "в первом приближении".

5.2.1. Терминология и основные понятия

Вторая версия протокола IKE не является совместимой с первой. Тем не менее, преемственность между двумя версиями прослеживается как на уровне архитектуры протокола, так и в попытке IKEv2 сохранить структуру сообщений, а также номера и числовые значения различных полей ("magic numbers") первой версии.

В новой версии протокола произошли изменения определений некоторых терминов. В частности, в новой спецификации защитный набор (protection suite) получил название криптографического набора (cryptographic suite). Контекст безопасности IKE SA обозначается как IKE_SA, а контексты безопасности IPsec SA (т.е. ESP SA или AH SA) называются дочерними контекстами безопасности и имеют общее обозначение CHILD_SA. Идентифицирующие цепочки в заголовках пакетов IKEv2 получили название индексов параметров безопасности (SPI – Security Parameter Indexes).

В спецификации IKEv2 RFC4306 [11] любое взаимодействие участников называется "обменом" (exchange) – на каждое сообщение-запрос должен поступить ответ. К сожалению, за последовательностью обмена сообщениями, проходящей для достижения определённой цели, то есть обменом в терминологии ISAKMP (exchange) или режимом в терминологии IKE (mode) также сохранилось название обмен (exchange, например, "Initial Exchange").

Работа протокола IKEv2 делится на две фазы. Первая фаза называется Начальным Обменом (Initial Exchange). Как правило, Начальный Обмен состоит из четырех сообщений. Первая пара сообщений (обмен) называется IKE_SA_INIT. В ходе этого обмена стороны согласуют параметры контекста безопасности IKE_SA, посылают одноразовые номера и проводят обмен

Диффи-Хеллмана. Вторая пара сообщений, составляющих Начальный Обмен, называется IKE_AUTH. В ходе обмена IKE_AUTH стороны передают свои идентификаторы, проводят взаимную аутентификацию. В результате его проведения стороны создают контекст безопасности IKE_SA и (первый) дочерний контекст безопасности CHILD_SA. В исключительных случаях Начальный Обмен может выполняться с помощью большего числа сообщений. В любом случае обмен сообщениями IKE_AUTH начинается только после завершения всех обменов IKE_SA_INIT, а обмен сообщениями второй фазы проходит только после завершения всех обменов IKE_AUTH (и, соответственно, Начального Обмена).

Во многих сценариях взаимодействия сторонам вполне достаточно одного контекста безопасности IPsec SA, и, таким образом, работа протокола IKEv2 исчерпывается Начальным Обменом. В случае необходимости в ходе второй фазы могут быть созданы дополнительные дочерние контексты безопасности CHILD_SA. Ко второй фазе относятся обмен CREATE_CHILD_SA, создающий дополнительный контекст безопасности CHILD_SA, и Информационный обмен (INFORMATIONAL Exchange), позволяющий осуществлять управление ранее созданными IPsec SA. Каждый из этих обменов состоит из двух сообщений.

В IKEv2 обязателен ответ на запрос. Надёжность обеспечивает инициатор. Если ответ не получен в течение определённого времени, инициатор запроса должен либо послать запрос заново, либо разорвать соединение.

5.2.2. Начальный обмен

В этом пункте в схемах обмена сообщениями будут использоваться следующие обозначения (структура блоков данных протокола IKEv2 подробно описана в разд. 3 RFC4306 [11]):

AUTH – блок данных Аутентификация
CERT – блок данных Сертификат
CERTREQ – блок данных Запрос Сертификата
CP – блок данных Настройка
D – блок данных Удаление
E – блок данных Зашифрованный
N – блок данных Уведомление
V – блок данных Код Разработчика
IDi, IDr – блоки данных Идентификация инициатора и ответчика соответственно
KEi, KEr – блоки данных Обмен Ключами инициатора и ответчика соответственно
Ni, Nr – блоки данных Одноразовый Номер инициатора и ответчика соответственно
TSi, TSr – блоки данных Селектор Трафика инициатора и ответчика соответственно

[...]– блок данных является необязательным

SK{...} – означает, что блоки данных, находящиеся внутри фигурных скобок, были зашифрованы, и для этих блоков была обеспечена защита целостности (об используемых при этом ключах речь идёт ниже)

HDR – заголовок IKEv2

В первой паре сообщений IKE_SA_INIT согласуются криптографические алгоритмы, осуществляется обмен одноразовыми номерами и проводится обмен Диффи-Хеллмана.

Схема обмена IKE_SA_INIT имеет следующий вид:

```
 N  Инициатор                Ответчик
(1) HDR, SAi1, KEi, Ni    =>
(2)                       <= HDR, SAr1, KEr, Nr, [CERTREQ]
```

Заголовок сообщения HDR содержит индексы параметров безопасности (SPI), номера версий и различные флаги. Блок данных SAi1 описывает криптографические наборы, предлагаемые инициатором для IKE_SA. В блоках данных KEi и Ni пересылаются открытое значение Диффи-Хеллмана и одноразовый номер инициатора. Ответчик выбирает криптографический набор из множества, предложенного инициатором, и сообщает о своём выборе в блоке SAr1, завершает обмен Диффи-Хеллмана, пересылая свое открытое значение в блоке KEr, и посылает свой одноразовый номер Nr.

После получения инициатором второго сообщения каждый из участников, используя результат обмена Диффи-Хеллмана, может найти величину SKEYSEED, на основе которой вычисляются все ключи, используемые в IKE_SA. Для всех последующих сообщений будет обеспечиваться защита целостности и шифрование всего содержимого, кроме заголовка. Для этого используются ключи SK_a (аутентификация) и SK_e (шифрование) соответственно, получаемые из SKEYSEED. Для каждого направления вычисляются свои ключи. Кроме того, вычисляется величина SK_d, на основе которой впоследствии будет генерироваться ключевой материал для CHILD_SA. Именно ключи SK_e и SK_a используются для шифрования и обеспечения целостности данных в обозначении SK{...}.

Схема обмена IKE_AUTH имеет следующий вид:

```
 N  Инициатор                          Ответчик
(3) HDR, SK{ IDi, [CERT,] [CERTREQ,]
     [IDr,] AUTH, SAi2, TSi, TSr }  =>
(4)                                 <= HDR, SK{ IDr, [CERT,]
                                        AUTH, SAr2, TSi, TSr }
```

Инициатор удостоверяет свою личность с помощью блока IDi, доказывает знание секрета, относящегося к IDi, и защищает целостность всего своего сообщения блоком AUTH. Он также, возможно, посылает свой сертификат в блоке CERT и запрашивает сертификат собеседника, указывая в блоке CERTREQ список сертификационных центров, которым инициатор доверяет.

Если в сообщение были включены сертификаты, то первый из них должен содержать открытый ключ для проверки AUTH. Блок данных IDr позволяет указать, с кем из представителей ответчика хочет установить связь инициатор (на одном узле может находиться несколько участников IPsec-взаимодействия). Блок SAi2 позволяет согласовать параметры дочернего контекста безопасности CHILD_SA, а блоки данных TSi и TSr (селекторы трафика) служат для согласования параметров трафика, который будет передаваться под защитой этого контекста безопасности.

Ответчик удостоверяет свою личность блоком IDr, посылает один или несколько сертификатов, в которых инициатор найдёт открытый ключ ответчика для проверки AUTH. Ответчик аутентифицирует сообщение и обеспечивает защиту его целостности блоком AUTH, после чего завершает согласование CHILD_SA, сообщая о выбранном криптографическом наборе в блоке данных SAr2.

5.2.3. Обмен CREATE_CHILD_SA

Этот обмен, состоящий из одной пары сообщений запрос/ответ, может начать любой из участников диалога после завершения Начального Обмена, вне зависимости от того, кто инициировал Начальный Обмен. Обмен CREATE_CHILD_SA проходит под защитой контекста безопасности IKE_SA (его параметры были согласованны в обмене IKE_SA_INIT). В этих сообщениях используется синтаксис зашифрованного блока данных (E). Термином инициатор теперь обозначается именно та из сторон, которая начала обмен второй фазы.

Запрос CREATE_CHILD_SA может содержать необязательный блок данных KE, если требуется дополнительный обмен Диффи-Хеллмана для обеспечения совершенной прогрессирующей секретности. Ключевой материал для CHILD_SA основывается на величине SK_d, вычисленной в ходе согласования IKE_SA, одноразовых номеров, обмен которыми произошёл в процессе выполнения CREATE_CHILD_SA, и секретного значения Диффи-Хеллмана (если дополнительный обмен Диффи-Хеллмана имел место).

Схема обмена CREATE_CHILD_SA имеет следующий вид:

```
Инициатор                          Ответчик

HDR, SK {[N], SA, Ni, [KEi],
      [TSi, TSr]}           =>
                             <=  HDR, SK {SA, Nr, [KEr],
                                    [TSi, TSr]}
```

Инициатор посылает предложения криптографических наборов для CHILD_SA в блоке SA, одноразовый номер в блоке Ni, возможно, посылает свое открытое значение Диффи-Хеллмана в блоке KEi и предлагаемые селекторы трафика TSi, TSr. Если этот обмен CREATE_CHILD_SA является обновлением существующего CHILD_SA, то блок данных уведомления (N)

должен указывать на контекст безопасности, для которого происходит обновление ключей. В противном случае блок N должен отсутствовать. Ответчик отвечает, используя тот же идентификатор сообщения (Message ID), выбранным криптографическим набором в блоке SA, своим одноразовым номером в блоке Nr, своим открытым значением Диффи-Хеллмана в блоке KEr, если осуществляется необязательный дополнительный обмен Диффи-Хеллмана. Селекторы трафика, который будет проходить под защитой создаваемого контекста безопасности, указываются в блоках TS, значение которых может представлять подмножество того, что предложил инициатор обмена CREATE_CHILD_SA.

5.2.4. Информационный обмен

Во время работы IKE_SA сторонам, участвующим во взаимодействии по протоколу IKEv2, может понадобиться передача друг другу управляющих сообщений, включающих сообщения об ошибках или уведомления о наступлении определённых событий. Этой цели служит Информационный Обмен (Informational Exchange). Информационный Обмен защищается с помощью IKE_SA и может произойти только после завершения Начального Обмена.

Управляющие сообщения, относящиеся к контексту безопасности IKE_SA, должны посылаться под защитой этого IKE_SA. Управляющие сообщения, относящиеся к дочернему контексту безопасности CHILD_SA, должны посылаться под защитой IKE_SA, создавшего этот CHILD_SA, или предка этого CHILD_SA (в случае обновления ключей).

Сообщения Информационного Обмена могут содержать один или несколько блоков Уведомление, Удаление или Настройка или не содержать их. Получатель запроса Информационного Обмена обязан послать какой-то ответ (иначе инициатор обмена сочтет, что сообщение не дошло, и пошлет его заново). Этот ответ может быть пустым сообщением, не содержащим блоков данных. Сообщение инициатора Информационного Обмена также может быть пустым; это способ проверки наличия соединения между участниками для одной из сторон. Контексты безопасности IPsec всегда существуют парами, по одному SA на каждое направление. При закрытии IPsec SA экземпляры этого SA на обоих концах должны быть удалены (закрыты). В случае вложенных контекстов безопасности все контексты безопасности должны уничтожаться вместе. Для удаления контекста безопасности посылается сообщение с одним или несколькими блоками данных Удаление, в которых перечисляются индексы параметров безопасности (SPI) удаляемых контекстов безопасности. Получатель такого сообщения обязан удалить указанные контексты безопасности.

Схема информационного обмена (Informational Exchange) имеет следующий вид:

```
Инициатор                                   Ответчик

HDR,SK { [N,] [D,] [CP,] ...} =>
```



```
                                   <= HDR, SK { [N,] [D,] [CP,] ... }
```

5.3. Основные изменения

Рассмотрим список основных изменений, которые претерпел протокол автоматического установления контекстов безопасности и управления ключами при переходе от первой версии ISAKMP/IKE ко второй версии IKEv2.

- Возникла единая спецификация протокола. Вместо документов RFC 2407, RFC 2408, RFC 2409, ссылающихся друг на друга, возник один документ RFC 4306. Документация была значительно переработана, кардинально изменился порядок подачи материала.
- Механизм обменов был значительно упрощён. Вместо восьми различных способов реализации первой фазы в протоколе ISAKMP/IKE предложен единственный метод. Сокращено число сообщений, позволяющих осуществить обмен. Ранее для проведения первой фазы требовалось переслать от трех до шести сообщений, для второй фазы – три сообщения. Теперь IKEv2 осуществляет установление IPsec SA за четыре или шесть сообщений.
- Исправлены недостатки системы безопасности протокола ISAKMP/IKE, такие как симметрия хэш-функций, используемых при аутентификации.
- Увеличена сетевая надёжность протокола. Теперь все сообщения должны подтверждаться. Введён механизм проверки связи между участниками взаимодействия. Специфицирована работа в случаях дублирования сообщений, посылки повторных сообщений, утери запроса или ответа. Специфицирован порядок работы в условиях отказов сети и атак на доступность.
- Введен новый блок данных Селектор Трафика. Его наличие позволяет сторонам определять потоки пакетов и обмениваться информацией о своих политиках IPsec. В первой версии протокола механизм селекторов трафика перегружал блок Идентификация и был менее гибким.
- Улучшена защита от атак на доступность.

Заметим также, что в новой версии протокола учтена необходимость поддержки его работы при пересечении устройств трансляции сетевых адресов (NAT), а также добавлена возможность асимметричной ("расширяемой") аутентификации (EAP) [11,12]. Эти вопросы остались за рамками данной статьи.

6. Заключение

Важным элементом семейства протоколов безопасности IPsec являются протоколы автоматического установления контекстов безопасности и

управления ключами. В протоколе ISAKMP введены структурные элементы, задана синтаксическая основа, предоставлены схемы обменов, что сделало его каркасом для протокола IKE. В протоколе обмена ключами в Интернет IKE определены фактическое содержимое обменов, реальные механизмы шифрования и аутентификации (в отличие от общих моделей ISAKMP). По сравнению с протоколом ISAKMP в протоколе IKE были, кроме того, сужены некоторые чрезмерно общие механизмы обмена, например, изменилась вторая фаза обмена.

Протокол IKE стал мощным и гибким инструментом, позволяющим организовывать автоматическое установление контекстов безопасности, управление ими, а также обновление ключей. Многочисленные реализации IKE широко используются на практике. Однако в процессе его внедрения было выявлено множество недостатков, в частности, чрезмерная сложность, плохая спецификация и целый ряд уязвимостей, что вызвало заслуженную критику и стало причиной разработки следующей версии протокола. Протокол IKEv2 был значительно упрощён по сравнению со своим предком. Это упрощение коснулось практически всех сторон протокола, начиная со схем обменов, закачивая спецификацией, однако по своим возможностям протокол IKEv2 практически не уступает протоколу IKE. Время покажет, насколько протокол IKEv2 оправдает возлагаемые на него надежды.

Литература

[1] Piper D., "The Internet IP Security Domain Of Interpretation for ISAKMP", RFC 2407, November 1998. http://RFC.net/rfc2407.html

[2] Maughhan D., Schertler M., Schneider M., and Turner J. "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998. http://RFC.net/rfc2408.html

[3] Harkins D., Carrel D., "The Internet Key Exchange (IKE)", RFC 2409, November 1998. http://RFC.net/rfc2409.html

[4] Orman H., "The Oakley Key Determination Protocol", RFC 2412, November 1998. http://RFC.net/rfc2412.html

[5] Krawczyk H., "SKEME: A Versatile Secure Key Exchange Mechanism for Internet", from IEEE Proceedings of the 1996 Symposium on Network and Distributed Systems Security.

[6] Karn, P., and Simpson, W., "Photuris: Session-Key Management Protocol", RFC 2522, March 1999. http://RFC.net/rfc2522.html

[7] Kent S., and Atkinson R., "IP Authentication Header", RFC 2402, November 1998. http://RFC.net/rfc2402.html

[8] Kent S., and Atkinson R., "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998. http://RFC.net/rfc2406.html

[9] N. Ferguson and B. Schneier, "A Cryptographic Evaluation of IPsec, CounterPane", http://www.counterpane.com/ipsec.html

[10] Perlman Radia, Charlie Kaufman "Analysis of the IPsec Key Exchange Standard". sec.femto.org/wetice-2001/papers/radia-paper.pdf

[11] C. Kaufman "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005. http://RFC.net/rfc4306.html

[12] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and Levkowetz, H., "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004. http://RFC.net/rfc2401.html

[13] NIST, "Secure Hash Standard", FIPS 180-1, National Institute of Standards and Technology, U.S. Department of Commerce, May 1994.

[14] Krawczyk H., Bellare M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997 http://RFC.net/rfc2104.html

[15] W. Diffie, M. Hellman, "New Derections in Cryptography", IEEE Transactions on Information Theory, vol. IT-22, no. 6 (1976), pp. 644-654.

[16] R.L. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, vol. 21, no. 2 (February 1978), pp. 120-126.

[17] Kent, S., and Atkinson, R., "Security Architecture for the Internet Protocol", RFC 2401, November 1998. http://RFC.net/rfc2401.html

[18] N. Doraswamy, D. Harkins, "IPsec. The New Security Standard for Internet, Intranets, and Virtual Private Networks", Prentice Hall, 2003.

[19] Shacham A., Monsour R., Pereira R., and M. Thomas, "IP Payload Compression Protocol (IPComp)", RFC 2393, August 1998. http://RFC.net/rfc2393.html

Приложение 1. Перевод терминов

В этом приложении перечислены некоторые термины и сокращения, используемые в англоязычной документации по ISAKMP/IKE, при переводе которых возможна неоднозначность. Приведён их перевод, употребляемый в данной работе.

Англоязычный термин	**Используемый перевод**
Access Control	Контроль доступа
Authentication Only Exchange	Обмен с аутентификацией без шифрования
Authentication with Public Key Encryption	Аутентификация с шифрованием открытым ключом
Confidentiality	Конфиденциальность/секретность данных
Cookie / Anti-clogging Token	Идентифицирующая цепочка
Data Integrity	Целостность данных
Identity Protection	Сокрытие / защита идентификаторов участников
Identity Protection Exchange	Обмен с сокрытием идентификаторов участников
Key Leakage	Рассекречивание ключей
Keyed Pseudorandom Function	Управляемая ключом псевдослучайная хэш-функция
Main Mode	Основной режим
Nonce	Одноразовый номер
Nonrepudiation	Невозможность отказа участников соединения от факта участия в передаче или приёме сообщений
Notify Payload	Блок данных Уведомление
Payload	Блок данных
Payload Chaining	Механизм образования цепочки блоков данных
PFS (Perfect Forward Secrecy)	Совершенная прогрессирующая секретность
Preshared Keys	Заранее распределённые ключи
Preshared Key Authentication	Аутентификация заранее распределёнными ключами
Proposal Payload / P payload	Блок данных Предложение
Protection Suite	Защитный набор

Англоязычный термин	**Используемый перевод**
Public key signatures authentication	Аутентификация цифровыми подписями в криптосистеме с открытыми ключами
Repudiability	Возможность отказа от обязательств, т.е. возможность отказа участников соединения от факта участия в передаче или приёме сообщений
SA (Security Association)	Контекст безопасности
SPI (Security Parameter Index)	Индекс параметров безопасности
Traffic Selector (TS)	Селектор трафика
Transform Payload / T payload	Блок данных Преобразование

Приложение 2. Криптографическая справка

В данном приложении дано краткое изложение некоторых понятий из области криптографии, необходимых для понимания работы протокола IKE.

Хэш-функция (hash function) – это преобразование, при котором из данных произвольной длины получается некоторое значение (свертка) фиксированной длины, называемое хэш-значением или хэш-кодом. Если на вход хэш-функции подается последовательность битов, составляющих сообщение, то значение хэш-функции называется профилем или дайджестом сообщения (message digest).

Криптографические хэш-функции отличаются необратимостью и отсутствием коллизий. Необратимость - невозможность найти исходные данные (аргументы) по известной хэш-функции и результатам её работы. Поскольку аргументы у хэш-функций произвольной длины, а результат - фиксированной длины, то множество значений у хэш-функций много меньше, чем множество определения. Поэтому потенциально для бесконечно большого числа аргументов значения совпадают. Отсутствие коллизий означает, что для известной хэш-функции и данного аргумента невозможно (вычислительно сложно) найти другой аргумент, дающий то же значение.

Реально применяемые в программировании хэш-функции являются приближениями к идеальной криптографической хэш-функции. Примерами плохой хэш-функции могут служить проверка на чётность, выполнение операций побитового XOR, вычисление контрольной суммы или кода CRC. Примером хорошей хэш-функции является алгоритм SHA [13], получающий на входе не более чем 264 бит данных, и дающий 160-битный результат.

Криптографические хэш-функции разделяются на два класса:

- Хэш-функции без ключа, результат которых называется кодом обнаружения изменений в сообщении MDC (Modification /Manipulation Detect Code).
- Управляемые ключом хэш-функции, результат которых называется кодом аутентификации сообщения MAC (Message Authentication Code).

Общепринятым считается, что алгоритмы вычисления хэш-функций открыты. Поэтому для хэш-функций без ключа профиль сообщения известен любому, кто знает исходное сообщение, а при использовании управляемой ключом хэш-функции для вычисления профиля сообщения необходимо знать секретный ключ. Функции первого типа обеспечивают только целостность сообщения (message integrity). Хэш-функции без ключа являются одним из

составных элементов цифровых подписей. Функции второго типа обеспечивают также аутентификацию сообщений (message authentication).

Точное определение криптографической хэш-функции hash следующее:

1. Для данного значения h должно быть трудно найти такое m, что h = hash(m). Это свойство называется стойкостью к прообразу (premiere resistance); строго говоря, оно отличается от свойства односторонности функции (one-way function).
2. Для данного аргумента m1, должно быть трудно найти другой аргумент m2, такой что hash(m1) = hash(m2). Это свойство называется стойкостью ко второму прообразу (second preimage resistant).
3. Должно быть трудно найти два различных аргумента m1 и m2, таких что hash(m1) = hash(m2).

Для защиты от атаки "по дню рождения" (birthday attack) соблюдение третьего свойства требует, чтобы хэш-значение было как минимум в два раза больше, чем требуется для соблюдения свойства стойкости к прообразу.

Хэш-функция, удовлетворяющая свойствам (1)-(3), всё же может иметь нежелательные свойства. Например, большинство популярных хэш-функций уязвимо к атаке "расширения длины" (length extension attack): при данном профиле сообщения h(m); зная длину профиля сообщения len(m), но не зная сообщение, атакующий может, найдя подходящее m', подсчитать h(m | m'), где | обозначает конкатенацию.

Идеальная хэш-функция является "скучной", у неё нет никаких особенных свойств. Она похожа на случайную функцию, отличие в том, что криптографическая хэш-функция детерминирована и эффективно вычисляется.

HMAC (Keyed-Hash Message Authentication Code) представляет собой алгоритм вычисления кода аутентификации сообщения (MAC) с использованием управляемой секретным ключом криптографической хэш-функции. HMAC, как и другие коды аутентификации сообщений, используется для защиты целостности и аутентификации сообщений. При вычислении HMAC может быть использована любая итеративная хэш-функция, например, MD5 или SHA-1; результат будет называться HMAC-MD5 или HMAC-SHA-1. Алгоритм HMAC является более надёжным, чем MAC, например, HMAC устойчив к вышеупомянутой атаке "расширения длины". Спецификация "HMAC: Keyed-Hashing for Message Authentication" содержится в документе RFC 2104 [14].

Цифровая подпись (digital signature) – это способ аутентификации информации, основанный на криптографии с открытым ключом. Отправитель сообщения для создания подписи использует свой секретный ключ. Получатель сообщения проверяет подпись открытым ключом отправителя. Конкретный метод цифровой подписи состоит из двух алгоритмов: один для подписания сообщений, другой – для проверки подписей. Существует

несколько методов цифровой подписи: RSA, DSS, ElGamal, ГОСТ Р36.10. Как правило, на вход алгоритму, вычисляющему подпись, подаётся не само сообщение, а профиль сообщения.

В отличие от MAC, цифровые подписи обеспечивают не только аутентификацию сообщения (authentication) и гарантию его целостности (integrity). Сторона, подписавшая сообщение, не может впоследствии отказаться от авторства сообщения (nonrepudiation).

Протокол – это спецификация последовательности шагов, предпринимаемых друг за другом в порядке строгой очередности двумя или большим количеством сторон для совместного решения некоторой задачи. Каждый участник протокола должен быть заранее оповещен о шагах, которые ему предстоит предпринять, и протокол должен описывать реакцию участников на любые ситуации, которые могут возникнуть в ходе его реализации. В случае если в основе протокола лежит криптографический алгоритм, протокол называется **криптографическим**.

Обмен Диффи-Хеллмана – криптографический протокол, позволяющий двум участникам, не обладающим знаниями друг о друге, создать общий секретный ключ, используя незащищённое соединение. Этот ключ может быть далее использован для шифрования последующих сообщений, например, в качестве ключа для симметричного алгоритма.

Одним из краеугольных камней современной криптографии стала статья Диффи и Хеллмана 1976 года [15], в которой авторы рассмотрели односторонние хэш-функции, криптографические системы с открытым ключом и механизм цифровых подписей. Кроме того, в исходной работе авторами был предложен протокол, позволяющий участникам, не имеющим никакой общей информации, создать общий секретный ключ. Значение криптографии с открытым ключом и цифровых подписей было осознано лишь через два года Ривестом, Шамиром и Адлеманом в их криптосистеме RSA [16].

Пусть G – конечная циклическая группа, порождённая элементом g. |G| - порядок этой группы. Для согласования общего секретного ключа два участника, Алиса и Боб, предпринимают следующие действия:

1). Они выбирают случайные натуральные числа a и b соответственно из интервала $[0, |G| -1]$ (если порядок группы не известен, они выбирают случайные числа из достаточно большого интервала). Алиса и Боб подсчитывают значения g^a и g^b и посылают эти значения друг другу по связывающему их незащищённому каналу.

2). Алиса вычисляет $(g^b)^a$, а Боб вычисляет $(g^a)^b$. Значение $g^{ab}=(g^b)^a=(g^a)^b$ может быть использовано в качестве общего секретного ключа – либо непосредственно как сеансовый ключ, либо для шифрования сгенерированного сеансового ключа.

Злоумышленник, прослушивающий канал, узнает g^b и g^a, но для взлома протокола ему необходимо решить задачу дискретного логарифмирования (discrete logarithm problem), которая считается вычислительно сложной.

Изначально в протоколе Диффи-Хеллмана отсутствовала аутентификация, и он был уязвим к атаке типа "человек посередине" (man-in-the-middle attack). Поэтому при применении протокола Диффи-Хеллмана используются различные способы аутентификации.

Совершенная прогрессирующая секретность (PFS – Perfect Forward Secrecy) – это свойство криптографического протокола, заключающееся в том, что компрометация ключа приводит к раскрытию только тех данных, которые были зашифрованы с использованием этого ключа. Это свойство рассматривается только по отношению к аутентифицируемым протоколам.

Приложение 3. Структура сообщений ISAKMP/IKE

П 3.1. Заголовок ISAKMP/IKE

Поскольку в протоколе ISAKMP/IKE синтаксис задаётся спецификацией ISAKMP, в данном приложении в качестве именования протокола будет употребляться "ISAKMP", а не "ISAKMP/IKE" или "IKE".

Сообщения ISAKMP начинаются с заголовка, после которого следует переменное число блоков данных. Наличие и порядок различных типов блоков данных в сообщении ISAKMP определяются содержимым поля Exchange Type (тип обмена) в заголовке пакета. На рис. П 3.1 представлен формат заголовка ISAKMP:

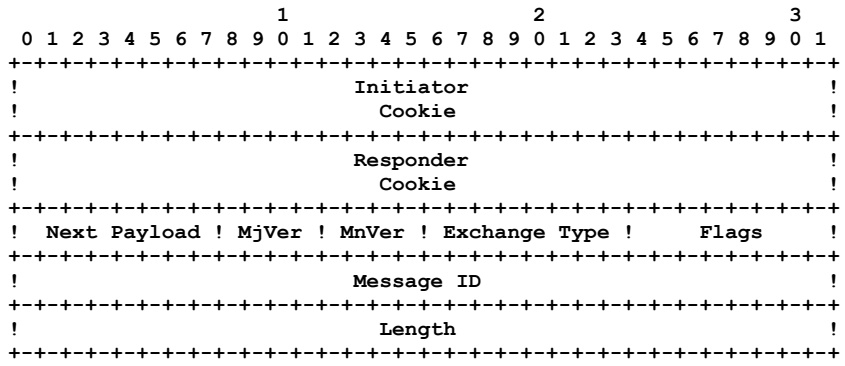

Рис. П 3.1.

Заголовок имеет фиксированный формат. Это ускоряет разбор пакета, упрощает программы синтаксического анализа пакетов ISAKMP.

Восьмибайтные идентифицирующие цепочки инициатора (Initiator Cookie) и ответчика (Responder Cookie) создаются соответствующими участниками и используются вместе с идентификатором сообщения (Message ID) для распознавания состояния, которое определяет ход обмена ISAKMP.

Помимо перечисленных полей, для этого также используется поле (SPI) в блоке данных Предложение (Proposal payload), о котором говорится ниже. Индекс параметров безопасности (SPI - Security Parameter Index) – это идентификатор для Контекста Безопасности (SA). У каждого из протоколов безопасности имеется своё "пространство SPI". В общем случае пара {протокол безопасности, SPI} может однозначно определять SA (RFC 2408),

однако в конкретных доменах интерпретации может использоваться дополнительная информация. Так, IPsec SA однозначно определяется тройкой {идентификатор протокола (AH/ESP), SPI, IP-адрес партнера}.

Четырехбайтное поле Message ID (идентификатор сообщения) применяется для определения состояния протокола в ходе второй фазы.

В таблице П 3.1 демонстрируется использование указанных четырёх полей сообщения ISAKMP в зависимости от состояния протокола (в этой таблице "+" означает наличие какого-то поля (оно не пустое), "0" – отсутствие, I-cookie - идентифицирующая цепочка инициатора, R-cookie - идентифицирующая цепочка ответчика).

	I-cookie	R-cookie	Message ID	SPI
Фаза 1				
(1) Начать согласование ISAKMP SA	+	0	0	0
(2) Ответить на согласование ISAKMP SA	+	+	0	0
Фаза 2				
(3) Начать согласование другого SA	+	+	+	+
(4) Ответить на согласование другого SA	+	+	+	+

Таблица П 3.1

Детали создания идентифицирующих цепочек для сообщений (1) и (2) в спецификации 2408 относятся на усмотрение реализаций ISAKMP (с оговоркой определённых требований).

Однобайтное поле Next Payload (следующий блок данных) указывает, какой из типов блоков данных следует за заголовком (в RFC 2408 определяются 13 типов блоков данных). Четырехбитные поля Major Version (старшая часть номера версии) и Minor Version (младшая часть номера версии) указывают версию ISAKMP. Однобайтное поле Exchange Type (тип обмена) определяет тип конкретного обмена, к которому относится данное сообщение. Четырехбайтное поле Length (длина) содержит длину в байтах сообщения в целом (заголовок + блоки данных).

Дополнительная информация передаётся с использованием битовой маски флагов в однобайтном поле Flags (флаги). Каждый бит определяет присутствие или отсутствие конкретной настройки. Определены три флага:

- E (encryption, шифрование) указывает, что блоки данных, следующие за заголовком, зашифрованы. Для всех обменов ISAKMP следует начать шифрование, как только оба участника обменялись блоками данных Key Exchange (обмен ключами).
- C (commit, подтверждение) говорит о потребности подтверждения сообщения. Используется для синхронизации (см. п. 3.5.6).
- A (authentication, только аутентификация) предназначен для использования в информационном обмене (Informational Exchange) вместе с блоком данных Notify (Уведомление). Эта настройка

позволяет указать на передачу информации без шифрования, обеспечивая только аутентификацию.

П 3.2. Сообщения и блоки данных

Каждый блок данных начинается с заголовка, имеющего общую структуру, за которым следует тело блока данных (рис. П 3.2):

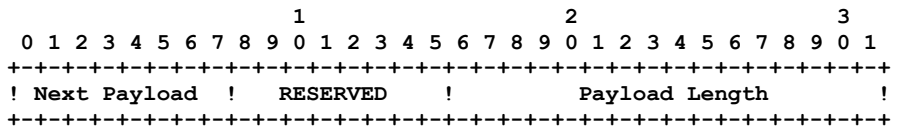

Рис. П 3.2.

В двухбайтном поле Payload Length (длина блока данных) указывается общая длина данного блока в байтах, включая заголовок (для блоков данных Контекст Безопасности и Предложение ситуация несколько сложнее, см. ниже).

Содержимое однобайтного поля Reserved (зарезервировано) равно 0.

Поле Next Payload указывает на тип следующего блока данных. Значение этого поля соответствует номеру, выделенному IANA для соответствующего типа блока данных (табл. П 3.2):

Следующий Блок Данных	**Значение**
Отсутствует	0
Контекст Безопасности (SA)	1
Предложение (P)	2
Преобразование (T)	3
Обмен Ключами (KE)	4
Идентификация (ID)	5
Сертификат (CERT)	6
Запрос Сертификата (CR)	7
Хэш-значение (HASH)	8
Цифровая подпись (SIG)	9
Одноразовый номер (NONCE)	10
Уведомление (N)	11

Следующий Блок Данных	**Значение**
Удаление (D)	12
Код разработчика (VID)	13
Зарезервировано	14-127
Частное использование	128-255

Таблица П 3.2.

Для синтаксического объединения заголовка и блоков данных в единое сообщение в ISAKMP применяется техника образования цепочки блоков данных (payload chaining). Поле Next Payload заголовка сообщения или заголовка каждого блока данных указывает на тип следующего блока данных. Таким образом, можно проводить разбор сообщения, последовательно проходя от заголовка к первому блоку данных и от каждого блока данных – к следующему блоку данных.

Следует отметить сходство с методом формирования пакета, используемым в протоколе IPv6. В заголовке каждого IPv6-пакета есть поле Next Header (следующий заголовок), указывающее на тип первого заголовка расширения (Extension Header). Поле Next Header каждого из заголовков расширения указывает на тип следующего заголовка расширения. Образуется цепочка заголовков расширения. Однако поле Next Header последнего заголовка расширения пакета IPv6 указывает на протокол транспортного уровня, сегмент которого передаётся в качестве полезной нагрузки, а поле Next Payload последнего блока данных пакета ISAKMP содержит 0.

Ниже на рис. П 3.3 представлен пример, иллюстрирующий организацию цепочки блоков данных.

Сцепление блоков данных – это не единственная особенность составных элементов пакета ISAKMP. Некоторые блоки данных зависят от других. Например, блок данных Transform (Преобразование) всегда находится внутри блока данных Proposal (Предложение), который, в свою очередь, всегда упаковывается внутрь блока данных Security Association (Контекст Безопасности). В некоторые блоки данных вводят атрибуты (attributes) индивидуальные для своего типа. Например, блок данных Certificate (Сертификат) определяет, какого типа сертификат он содержит.

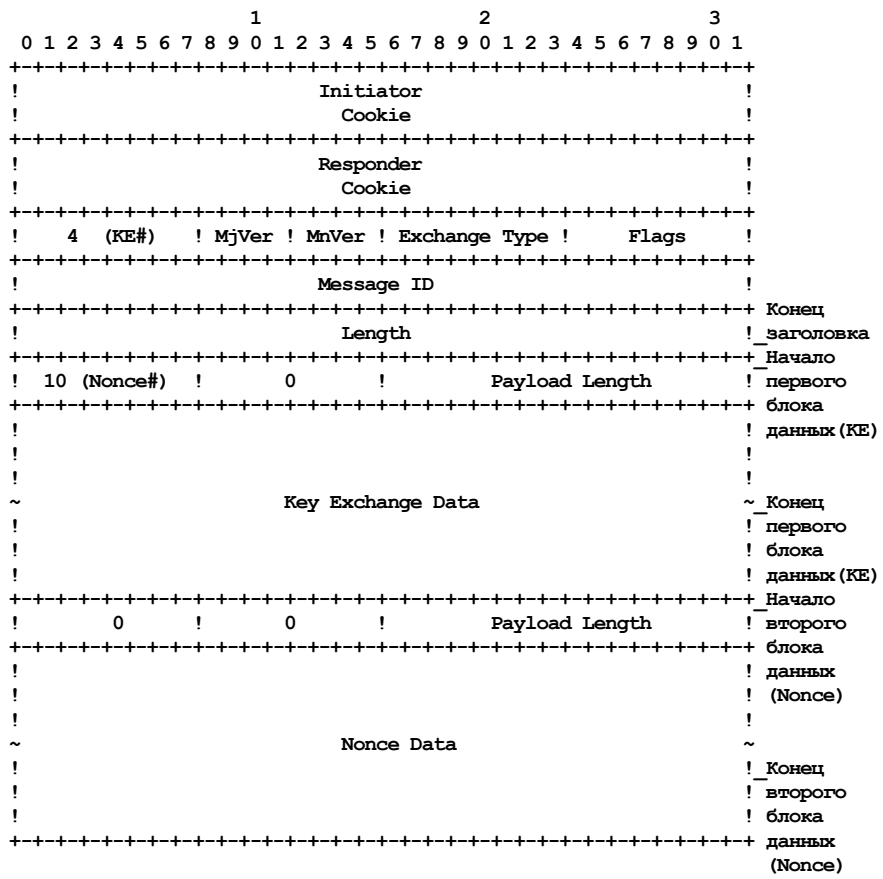

Рис. П 3.3.

П 3.3. Блоки данных

Блоки данных ISAKMP могут быть универсальными или специальными. Здесь и далее рассматриваются лишь существенные особенности организации сообщений протокола ISAKMP. Ряд деталей опущен, например, выравнивание полей.

Универсальные блоки данных синтаксически устроены одинаково (за исключением заголовка, их тело состоит из единственного поля, в котором и передаются данные). Они различаются семантикой этого поля. К этим типам блоков данных относятся следующие:

- блок данных Key Exchange (Обмен Ключами) содержит информацию, необходимую для проведения обмена ключами;
- блок данных Hash (Хэш) содержит результат действия хэш-функции;
- блок данных Signature (Подпись) содержит цифровую подпись;
- блок данных Nonce (Одноразовый Номер) содержит псевдослучайную величину, применяемую в ходе обмена;
- блок данных Vendor ID (Код Разработчика) зарезервирован для разработчиков; каждый производитель реализации ISAKMP может использовать этот блок данных по своему усмотрению.

Например, блок данных Key Exchange имеет следующий формат (рис. П 3.4):

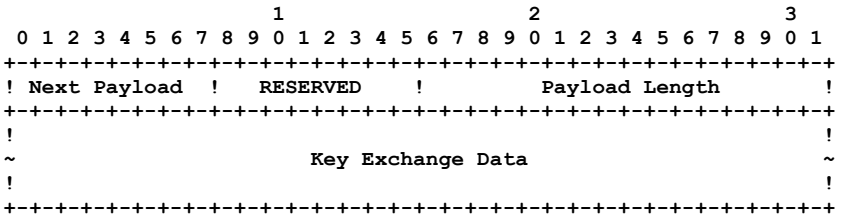

Рис. П 3.4.

Здесь Key Exchange Data – это данные обмена ключами, например, открытые значения участников взаимодействия для обмена Диффи-Хеллмана.

Следующие блоки данных по своему формату отличаются незначительно:

- блок данных Certificate (Сертификат);
- блок данных Certificate Request (Запрос Сертификата);
- блок данных Identification (Идентификация).

Тело начинается с одного (или двух для блока данных Идентификация) вспомогательных полей, конкретизирующих или предоставляющих дополнительную информацию о данных, передаваемых в последнем поле переменной длины. Например, блок данных Certificate Request имеет следующий формат (рис. П 3.5):

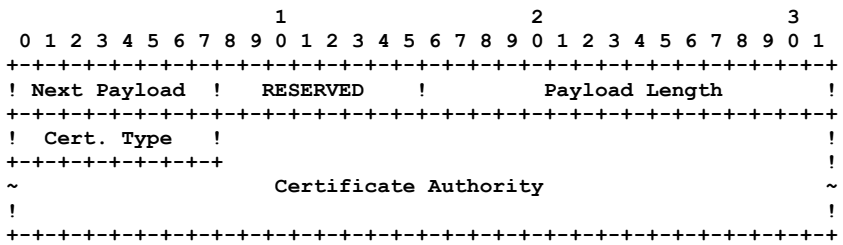

Рис. П 3.5.

Здесь поле Certificate Type указывает тип запрашиваемого сертификата, а Certificate Authority – название центра сертификации, выпустившего соответствующий сертификат.

Блоки данных Notification (Уведомление) и Delete (Удаление) имеют более сложную структуру. Её детали, как и подробное описание всех полей каждого из существующих в ISAKMP блоков данных, применяемые константы, ограничения на поля можно найти в спецификации RFC 2408 [2]. Для понимания принципов работы протокола это не имеет значения и здесь будет опущено.

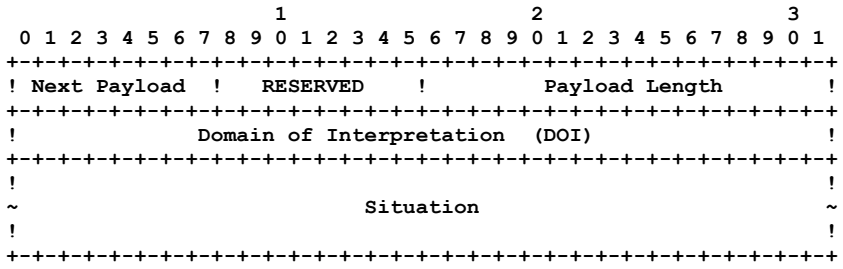

Рис. П 3.6.

Блок данных Security Association (Контекст Безопасности) применяется для согласования контекстов безопасности, а также для информирования о домене интерпретации (DOI) и ситуации, при которой происходит согласование. На рис. П 3.6 представлен формат этого блока данных:

В сообщении, служащем для согласования защитных наборов для устанавливаемого контекста безопасности, за блоком Security Association (SA) следуют блоки данных Proposal (Предложение) и Transform (Преобразование). Поэтому, в отличие от предыдущих случаев, поле Next payload не может иметь произвольное значение. Блоки данных Предложение и Преобразование, идущие за блоком Контекст Безопасности, считаются с ним связанными, и

поэтому их номера (2 и 4 соответственно, см. таблицу кодов блоков данных) не могут быть значением рассматриваемого поля в блоке данных SA.

Поле Payload Length, в отличие от обычной ситуации, содержит длину в байтах как самого блока данных Контекст Безопасности, так и всех связанных с ним (и следующих за ним) блоков данных Предложение и Преобразование.

Поле Domain of Interpretation определяет домен интерпретации, при котором осуществляется согласование. Это 32-битное целое число без знака. Значение 0 в ходе первой фазы соответствует согласованию контекста безопасности ISAKMP. Значение 1 присвоено домену интерпретации IPsec. Остальные значения зарезервированы IANA.

Поле переменной длины Situation (ситуация) зависит от домена интерпретации и описывает ситуацию, при которой происходит согласование. Интерпретация данного поля описывается в спецификации RFC 2407 "IPsec DOI" [1], в которой этот вопрос рассматривается в деталях.

Оставшиеся блоки данных связаны с блоком Контекст Безопасности и не встречаются сами по себе. Это блоки данных Proposal (Предложение) и Transform (Преобразование). Они могут использоваться только в ходе согласования устанавливаемого контекста безопасности. Блоки первого типа описывают предложение отдельного контекста безопасности, второго – отдельное преобразование в таком предложении. К одному Контексту Безопасности может относиться несколько предложений, и к каждому предложению может относиться несколько преобразований. Формат блока данных Proposal имеет следующий вид (рис. П 3.7):

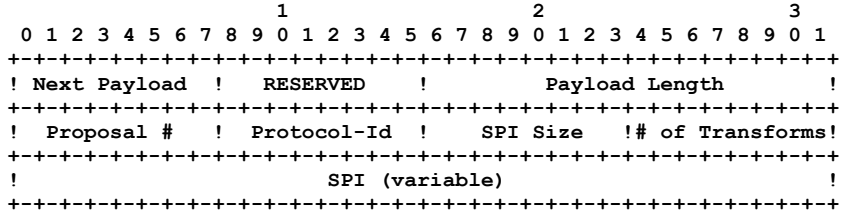

Рис. П 3.7.

Поле Next Payload может принимать лишь значения 0 или 2. Оно равно 0, если это предложение является последним из предложений, соответствующих контексту безопасности, к которому оно относится. Таким образом, значение 0 в одном из блоков данных необязательно свидетельствует о конце всего сообщения. Оно равно 2, если имеются другие предложения. Блок данных Преобразование, который может следовать лишь за блоками Предложение, считается относящимся к своему предложению, поэтому поле Next Payload не может принимать значение, равное его номеру 3.

Поле Payload Length (длина блока данных) содержит суммарную длину в байтах этого блока данных Предложение и длину всех связанных с ним блоков Преобразование.

Поле Protocol-ID (идентификатор протокола) определяет тип протокола для данного согласования (например, ESP, AH, OSPF, TLS).

Поле "Proposal #" (номер предложения) будет рассмотрено в подразделе П 3.4.

Поле "# of Transforms" (количество преобразований) определяет количество блоков Преобразование, относящихся к данному блоку данных Предложение. Об использовании этого поля также пойдёт речь в подразделе П 3.4.

Поле SPI Size (размер индекса параметров безопасности) указывает длину поля SPI.

Поле SPI содержит индекс параметров безопасности и гарантирует, что SPI связан с полем Protocol-ID в соответствии со спецификацией RFC 2401 " Security Architecture for the Internet Protocol" [17].

Ниже на рис. П 3.8 представлен формат блока данных Transform:

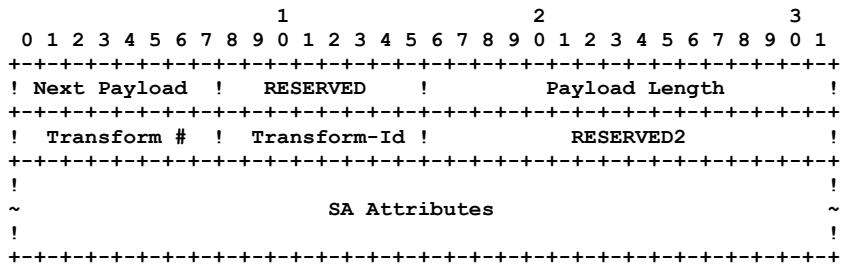

Рис. П 3.8

Поле Next Payload принимает значения либо 0 (это преобразование является последним для предложения, которому оно соответствует), либо 3 (за этим преобразованием следует ещё одно преобразование, соответствующее тому же предложению).

Поле Payload Length содержит длину только данного блока Преобразование.

Поле " Transform #" (номер преобразования) содержит номер текущего преобразования среди всех преобразований, соответствующих предложению, к которому оно относится. Подробнее об этом см. в подразделе П 3.4.

Поле Transform-ID определяет идентификатор преобразования для идентификатора протокола блока данных Предложение, которому соответствует это преобразование. Идентификаторы преобразований определены доменом интерпретации и зависят от согласовываемого протокола.

Поле SA Attributes (атрибуты контекста безопасности) имеет переменную длину. Оно содержит атрибуты контекста безопасности, в соответствии с определением преобразования, заданного в поле Transform-Id. Внутри одного блока может быть несколько атрибутов. Атрибут, в сущности, – это пара тип/значение. У каждого атрибута имеется определённый тип, и этому типу соответствует значение. Например, атрибут типа "алгоритм шифрования" может иметь значение "CAST" (т.е. не слово "CAST", а соответствующий номер алгоритма, присвоенный IANA).

Первый бит поля называется битом формата атрибута (Attribute Format bit). Если он равен 1, то атрибут называется простым и имеет длину 4 байта (рис. П 3.9):

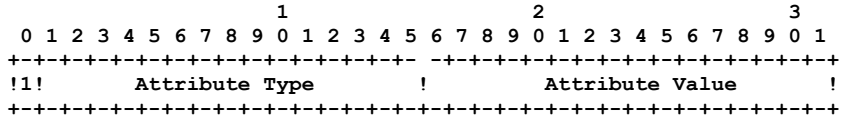

Рис. П 3.9.

Если бит формата атрибута равен 0, то атрибут считается сложным и имеет переменную длину (рис. П 3.10):

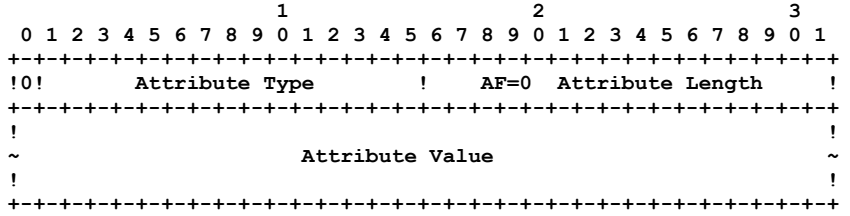

Рис. П 3.10.

П 3.4. Примеры пакетов ISAKMP

Модульное строение сообщения ISAKMP облегчает разбор сообщения, позволяет разделять информацию, имеющую различное применение. Внутри сообщения блоки данных связаны в цепочку. Как было показано выше, такая методика не является уникальной.

Особенностью ISAKMP следует считать сложную зависимость между некоторыми блоками данных (Контекст Безопасности, Предложение, Преобразование). В сообщении, обеспечивающем согласование параметров устанавливаемого контекста безопасности, можно условно выделить одну группу блоков данных, объединённых вокруг блока данных Контекст Безопасности, и не связанные с ней другие блоки данных (их номенклатура и

число определяются типом обмена). Эта условная группа блоков данных состоит из одного блока Контекст Безопасности, за которым следует один или несколько блоков Предложение, связанных с этим контекстом безопасности, за каждым из которых следует один или несколько блоков Преобразование, связанных с соответствующим предложением. Блоки Контекст Безопасности, Предложение и Преобразование, входящие в эту условную группу, рассматриваются вместе как одно целое, и поэтому между ними не может быть блоков никаких других типов.

Поле Next Payload блока Контекст Безопасности указывает на первый блок данных, следующий за указанной условной группой блоков (а не на блок Предложение, который следует непосредственно за блоком Контекст Безопасности).

Как уже было отмечено в подразделе П 3.3, поле Next Payload блока Предложение может содержать лишь значения 2 или 0 (см. таблицу кодов блоков данных). Это можно объяснить следующим образом. Поскольку блок Предложение входит в состав условной группы и зависит от Контекста Безопасности, он не может указывать ни на один из "внешних" блоков, возможно следующих за рассматриваемой группой. Кроме того, поскольку блок Преобразование, следующий за данным блоком Предложение, или несколько таких блоков рассматриваются как относящиеся к конкретному предложению, поле Next Payload блока Предложение не может указывать на блок Преобразование. Таким образом, поле Next Payload блока Предложение указывает либо на следующий блок Предложение (относящийся к тому же Контексту Безопасности), либо на конец списка (внутри группы).

Поле Next Payload блока Преобразование может содержать значения 3 или 0. Значение 3 означает, что имеются дополнительные блоки Преобразование, соответствующие тому же предложению. 0 означает, что данное преобразование является последним для данного предложения.

Блок данных Предложение определяет базовые параметры согласовываемого контекста безопасности – используемый протокол (Protocol-ID) и индекс параметров безопасности (SPI).

Для согласования сложных политик необходимо согласование нескольких контекстов безопасности, поэтому одному блоку Контекст Безопасности может соответствовать несколько блоков Предложение. Каждое из предложений может породить один согласованный контекст безопасности. Поле "Proposal #" блока Предложение содержит номер предложения.

Поскольку для каждого предложения может быть согласован только один из нескольких криптографических алгоритмов, одному блоку Предложение может соответствовать несколько блоков Преобразование. Поле "# of Transforms" содержит количество блоков Преобразование для данного блока Предложение.

В качестве иллюстрации этой сложности рассмотрим примеры строения сообщения ISAKMP, участвующего в согласовании контекста безопасности, взятые из [18].

В качестве первого примера рассмотрим вполне практически реалистичное предложение двух защитных наборов (ЗН): "Применять к трафику протокол AH с хэш-функцией MD5 и протокол ESP с алгоритмом шифрования 3DES или применять на выбор протокол ESP с алгоритмом шифрования 3DES или протокол ESP с алгоритмом шифрования DES". Первый защитный набор состоит из двух предложений, каждое из которых содержит единственное преобразование. Второй защитный набор включает единственное предложение, но с двумя возможными преобразованиями.

На рис. П 3.11 показано строение сообщения, предлагающего указанные защитные наборы (для упрощения все прочие блоки данных опущены).

Как отмечалось в подразделе 3.1 настоящей статьи, поле "Proposal #" блока Предложение используется для выражения логических отношений между блоками Предложение. Совпадающие значения полей "Proposal #" указывают на отношение "И" между этими предложениями. Различные значения полей "Proposal #" указывают на отношение "ИЛИ". В приведённом выше примере для первого защитного набора требуется применять AH MD5 (первое предложение и соответствующее преобразование) "И" ESP 3DES (второе предложение и преобразование), поэтому для обоих предложений первого защитного набора применяется одинаковый номер 1. Предложения, имеющие одинаковые номера, должны идти друг за другом и не могут быть разделены блоком Предложение с другим номером.

Поскольку каждое преобразование – это уточнённый вариант предложения, для которого определены криптографические алгоритмы, то между преобразованиями могут быть только отношения "ИЛИ". Поэтому значения поля "Transform #" для преобразований, соответствующих одному предложению, не могут совпадать. В частности, в примере для второго защитного набора указывается единственное предложение с двумя преобразованиями. Преобразования предлагают для согласования алгоритмы шифрования 3DES или DES, поэтому используются разные номера. Получатель для каждого из протоколов (предложений) должен выбрать единственное преобразование или отказаться от предложения в целом.

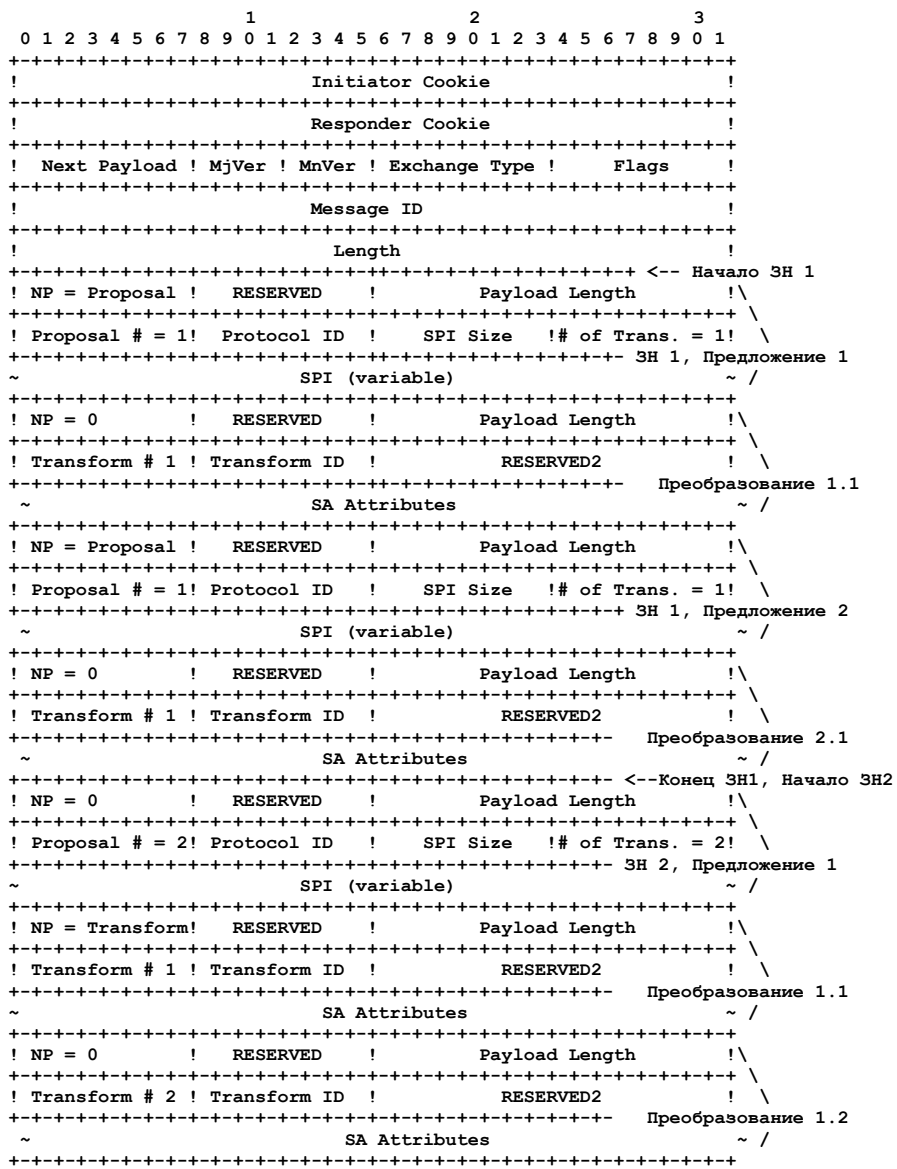

Рис. П 3.11.

Теперь рассмотрим более сложный пример. ISAKMP согласовывает следующую политику: "Аутентифицировать весь трафик, если возможно, его также шифровать и, если возможно, его также сжимать". Использование

протокола сжатия полезной нагрузки пакетов IP PCP "IP Payload Compression Protocol" [19] определено в RFC 2407 "IPsec DOI" [1]. Фактически, в данном примере инициатор предлагает на выбор три защитных набора (порядок существенен): AH-1 или ESP-2 или (ESP-3 и PCP-3). Здесь индекс после названия протокола означает номер предложения соответствующего протокола (протоколов). Заметим, что в качестве протокола для второго и третьего защитных наборов выбран ESP, а не (AH и ESP), поскольку протокол ESP сам обеспечивает аутентификацию.

Конкретизируем эти предложения. Пусть для аутентификации предлагаются хэш-функции SHA и MD5, для шифрования – алгоритмы 3DES и AES, для сжатия – алгоритмы LZS и Deflate. Заметим, что порядок перечисления алгоритмов отражает предпочтения инициатора. Тогда в данном сообщении блоки данных, связанные с блоком Контекст Безопасности, будут выглядеть следующим образом (для упрощения заголовок и все прочие блоки данных опущены):

Предложение 1: AH
- Преобразование 1: HMAC-SHA
- Преобразование 2: HMAC-MD5

Предложение 2: ESP
- Преобразование 1: 3DES с HMAC-SHA
- Преобразование 2: 3DES с HMAC-MD5
- Преобразование 3: AES с HMAC-SHA
- Преобразование 4: AES с HMAC-MD5

Предложение 3: ESP
- Преобразование 1: 3DES с HMAC-SHA
- Преобразование 2: 3DES с HMAC-MD5
- Преобразование 3: AES с HMAC-SHA
- Преобразование 4: AES с HMAC-MD5

Предложение 3: PCP
- Преобразование 1: LZS
- Преобразование 2: Deflate

Это достаточно сложное предложение инициатора можно описать следующим логическим выражением:

((AH-HMAC-SHA OR AH-HMAC-MD5) OR (3DES с HMAC-SHA OR 3DES с HMAC-MD5 OR AES с HMAC-SHA OR AES с HMAC-MD5) OR [(3DES с HMAC-SHA OR 3DES с HMAC-MD5 OR AES с HMAC-SHA OR AES с HMAC-MD5) AND (PCP-LZS OR PCP-DEFLATE)]).

Наличие нескольких протоколов безопасности, каждый из которых может быть реализован с помощью различных криптографических алгоритмов, является причиной сложности политик ISAKMP (а значит и IKE). Для согласования сложных политик требуется применение нетривиальной

структуры сообщений, отражающей эту сложность. В этом и состоит причина определённой замысловатости связей между блоками данных.

Следует отметить, что ограничения на минимальную и максимальную длину сообщений ISAKMP/IKE (размер некоторых блоков данных, таких как Сертификат, может быть очень большим), а также возможная фрагментация IP-дейтаграмм, переносящих сообщения ISAKMP/IKE в спецификации ISAKMP/IKE первой версии (RFC 2407, 2408, 2409) не рассматриваются.