Network Working Group
Request for Comments: 4035
Obsoletes: 2535, 3008, 3090, 3445, 3655, 3658, 3755, 3757, 3845
Updates[1]: 1034, 1035, 2136, 2181, 2308, 3225, 3597, 3226
Category: Standards Track

R. Arends
Telematica Instituut
R. Austein
ISC
M. Larson
VeriSign
D. Massey
Colorado State University
S. Rose
NIST

March 2005

Изменение протокола для защитных расширений DNS

Protocol Modifications for the DNS Security Extensions

Статус документа

Этот документ содержит спецификацию стандартного протокола, предложенного сообществу Internet, и служит приглашением к дискуссии в целях развития. Текущее состояние стандартизации и статус описанного здесь протокола можно узнать из документа Internet Official Protocol Standards (STD 1). Документ может распространяться без ограничений.

Авторские права

Copyright (C) The Internet Society (2005).

Аннотация

Этот документ является частью набора документов, описывающих расширения DNS Security (DNSSEC). Эти расширения представляют собой набор новых записей RR и изменений протокола, обеспечивающих совместно аутентификацию источника данных и целостность данных DNS. В данном документе описаны изменения протокола, вносимые DNSSEC. Документ определяет концепцию подписанной зоны и требования к использованию DNSSEC. Описываемые здесь методы позволяют защищённому распознавателю[2] аутентифицировать как записи DNS RR, так и уполномоченную индикацию ошибок DNS.

Данный документ отменяет действие RFC 2535 и включает в себя все обновления к RFC 2535.

Оглавление

[1]В оригинале ошибочно указано также обновление RFC 3007. См. https://www.rfc-editor.org/errata_search.php?rfc=4035. Прим. перев.
[2]Security-aware resolver.

1. Введение

Защитные расширения DNS (DNSSEC) представляют собой набор новых записей и изменений в протоколе, которые добавляют аутентификацию источника данных и поддержку целостности данных DNS. В этом документе рассматриваются изменения протокола, вносимые DNSSEC. Глава 2 определяет концепцию подписанной зоны и список требований для этого. В главе 3 описываются изменения в поведении уполномоченных серверов имён, требуемые для работы с подписанными зонами. В главе 4 описано поведение объектов, включающих функции защищённого распознавателя. В главе 5 определено использование записей DNSSEC RR для аутентификации откликов.

1.1. Связанные документы

Этот документ является частью семейства документов, определяющих расширения DNSSEC. Документы следует читать как единое целое.

[RFC4033] содержит введение в DNSSEC и определения основных терминов; предполагается, что читатель внимательно ознакомился с этим документом. [RFC4033] также включает список документов, обновлённых или отменённых данным набором документов.

[RFC4034] определяет записи DNSSEC RR.

Предполагается, что читатель знаком с основными концепциями DNS, описанными в [RFC1034], [RFC1035] и последующих документах, обновляющих два указанных документа (в частности, [RFC2181] и [RFC2308]).

Данный документ определяет работу протокола DNSSEC.

1.2. Зарезервированные слова

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с [RFC2119].

2. Подписывание зоны

DNSSEC вводит концепцию подписанной зоны. Подписанная зона включает открытый ключ DNS (DNSKEY), сигнатуру RR (RRSIG[1]), записи NSEC[2] и (не обязательно) DS[3], как указывают правила в параграфах 2.1, 2.2, 2.3 и 2.4, соответственно. Зона, не включающая записи в соответствии с указанными правилами, является не подписанной.

DNSSEC требует замены определения записи CNAME ([RFC1035]). Параграф 2.5 меняет CNAME RR, чтобы записи RRSIG и NSEC появлялись с тем же именем владельца, которое указано в CNAME RR.

DNSSEC задаёт положение двух новых типов RR - NSEC и DS, - которые могут размещаться на родительской стороне среза зоны (т. е., в точке передачи полномочий). Это является исключением из общего правила, запрещающего включение данных в родительскую зону на срезе зоны. Это изменение описано в параграфе 2.6.

2.1. Включение в зону записей DNSKEY RR

Для подписывания зоны её администратор генерирует по крайней мере одну пару ключей (закрытый и открытый) и использует закрытый ключ или ключи для подписывания полномочных наборов RRset в зоне. Для каждого закрытого ключа, использованного при создании записи RRSIG RR в зоне, в эту зону **следует** включать запись DNSKEY RR, содержащую соответствующий открытый ключ. Запись DNSKEY RR для ключа зоны **должна** иметь установленный бит Zone Key в поле флагов RDATA (см. параграф 2.1.1 документа [RFC4034]). Открытые ключи, связанные с другими операциями DNS, **могут** сохраняться в записях DNSKEY RR, которые не помечаются как ключи зоны, но такие записи **недопустимо** использовать для проверки RRSIG.

Если администратор зоны разрешает использование подписанной зоны, отличающееся от островка безопасности, вершина зоны **должна** содержать по крайней мере одну запись DNSKEY RR, используемую в качестве защищённой точки входа в зону. Эта защищённая точка входа используется как назначение защищённой передачи полномочий через соответствующую запись DS RR в родительской зоне (см. [RFC4034]).

2.2. Включение в зону записей RRSIG RR

Для каждого полномочного набора RRset в подписанной зоне **должна** присутствовать по крайней мере одна запись RRSIG, удовлетворяющая приведённым ниже требованиям.

- Имя владельца RRSIG совпадает с именем владельца.
- Класс RRSIG совпадает с классом RRset.
- Поле RRSIG Type Covered совпадает с типом RRset.
- Поле RRSIG Original TTL совпадает со значением TTL для RRset.
- Значение TTL для RRSIG RR совпадает со значением TTL для RRset.
- Поле RRSIG Labels совпадает с числом меток в имени владельца RRset, без учёта пустой (null) корневой метки и без учёта самой левой метки, если та является шаблоном.
- Поле RRSIG Signer's Name совпадает с именем зоны, содержащей RRset.
- Поля RRSIG Algorithm, Signer's Name и Key Tag идентичны записи DNSKEY для ключа зоны на срезе последней.

Процесс создания RRSIG RR для данного набора RRset описан в [RFC4034]. Набор RRset **может** включать множество связанных с ним записей RRSIG. Отметим, что по причине тесной связи записей RRSIG RR с наборами RRset, чьи сигнатуры эти записи содержат, записи RRSIG RR, в отличие от других типов DNS RR, не формируют наборов RRset. В частности, значения TTL для записей RRSIG RR с общим именем владельца, не следуют правилам для RRset, описанным в [RFC2181].

Сами записи RRSIG RR **недопустимо** подписывать, поскольку подписывание RRSIG RR не имеет смысла, но создаёт бесконечный цикл в процессе подписывания.

Набор NS RRset, появляющийся около имени среза зоны, **должен** быть подписан, но наборы NS RRset около точек передачи полномочий (т. е., наборы NS RRset в родительской зоне, которые делегируют имя серверам имён дочерней зоны) подписывать **недопустимо**. Склеивающие RRset, связанные с передачей полномочий, подписывать **недопустимо**.

Должна присутствовать запись RRSIG для каждого набора RRset, использующая по крайней мере один ключ DNSKEY для каждого алгоритма из набора DNSKEY RRset на вершине зоны. Сам набор DNSKEY на вершине зоны **должен** быть подписан с использованием каждого алгоритма, включённого в набор DS RRset, расположенный у передающего полномочия родителя (если таковой имеется).

[1]Resource Record Signature.
[2]Next Secure – следующий защищенный владелец.
[3]Delegation Signer – подписавший передачу полномочий.

2.3. Включение в зону записей NSEC RR

Каждое имя владельца в зоне, содержащей полномочные данные или набор NS RRset точки передачи полномочий, **должно** иметь запись NSEC. Формат NSEC RR и процесс создания таких записей для данного имени описан в [RFC4034].

Значение TTL для любой записи NSEC RR **следует** устанавливать таким же, как указано в поле минимального значения TTL записи SOA RR для зоны.

Записи NSEC (и связанному с ней набору RRSIG RRset) **недопустимо** быть единственным набором RRset для любого конкретного имени владельца. Т. е., в процессе подписывания **недопустимо** создавать NSEC или RRSIG RR для узлов имени владельца, которые не являются именами владельца какого-либо набора RRset до подписания зоны. Основная причина этого заключается в обеспечении согласованности пространства имён между подписанной и не подписанной версиями одной зоны и снижения риска несогласованности откликов в обычных (не защищённых) рекурсивных серверах имён.

Битовая последовательность типа каждой записи NSEC в подписанной зоне **должна** показывать наличие самой записи NSEC и соответствующей записи RRSIG.

Различие между наборами имён владельцев, требующими записи RRSIG, и наборами имён владельцев, требующими записи NSEC, трудно уловимо и слабо разъяснено. Записи RRSIG используются с именами владельцев всех полномочных наборов RRset. Записи NSEC используются с именами владельцев всех имён, для которых полномочна подписанная зона, а также с именами владельцев при передаче полномочий из подписанной зоны в дочерние зоны. Ни одна из записей NSEC и RRSIG не используется (в родительской зоне) для имён владельцев наборов RRset склеивающего адреса. Отметим, однако, что упомянутое различие для большинства случаев видимо только в процессе подписывания зоны, поскольку наборы NSEC RRset являются полномочными данными и должны подписываться. Таким образом, любое имя владельца, имеющее набор NSEC RRset в подписанной зоне, будет иметь записи RRSIG RR.

Битовая последовательность для NSEC RR в точке передачи полномочий требует особого внимания. Биты, соответствующие делегируемому набору NS RRset и любым наборам RRset, для которых родительская зона имеет полномочные данные, **должны** быть установлены; биты, соответствующие любым наборам, не относящимся к числу NS RRset **должны** быть сброшены, если родительская зона не содержит для этих наборов полномочные данные.

2.4. Включение в зону записей DS RR

Запись DS создаёт цепочку аутентификации между зонами DNS. Набор DS RRset **следует** включать в точке передачи полномочий, если дочерняя зона подписывается. Набор DS RRset **может** включать множество записей, каждой из которых соответствует открытый ключ в дочерней зоне, используемый для проверки записей RRSIG в этой зоне. Все наборы DS RRset в зоне **должны** быть подписаны и **недопустимо** включать наборы DS RRset на вершине зоны.

Записи DS RR **следует** указывать на запись DNSKEY RR, присутствующую в наборе DNSKEY RRset на вершине дочерней зоны, а набор DNSKEY RRset на вершине дочерней зоны **следует** подписывать с использованием соответствующего закрытого ключа. Записи DS RR, которые не соответствуют этим условиям, бесполезны при проверке корректности, но, поскольку записи DS RR и соответствующие им записи DNSKEY RR находятся в разных зонах, а согласованность DNS достаточно свободна, может возникать временное рассогласование.

Значению TTL в наборах DS RRset **следует** соответствовать значению TTL в делегирующем наборе NS RR (т. е., в наборе NS RRset из той же зоны, где содержится DS RRset).

При создании записей DS RR требуется знать соответствующие записи DNSKEY RR в дочерней зоне, что достигается путём обмена информацией между родительской и дочерней зонами. Этот обмен информацией не рассматривается в данном документе.

2.5. Изменения в CNAME RR

Если для имени в подписанной зоне имеется набор CNAME RRset, для этого имени **требуются** соответствующие наборы RRSIG и NSEC RRset. В целях обеспечения защиты динамических обновлений для этого имени может также использоваться набор KEY RRset ([RFC3007]). Другие типы записей для этого имени **недопустимы**.

Сказанное выше отличается от исходного определения CNAME в [RFC1034], которое не позволяет другим типам существовать вместе с CNAME RR, но подписанная зона требует наличия записей NSEC и RRSIG RR для каждого полномочного имени. Для разрешения этого конфликта данная спецификация меняет определение записи CNAME, разрешая сосуществование записей CNAME с записями NSEC и RRSIG.

2.6. Типы DNSSEC RR, появляющиеся на срезах зон

DNSSEC вводит два новых типа RR, которые являются не совсем обычными в том, что они могут присутствовать на родительской стороне среза зоны. На родительской стороне среза зоны (т. е., в точке передачи полномочий) для имени владельца **требуется** включать записи RR. На срезе может также присутствовать запись DS RR, если делегируемая зона подписывается и имеет цепочку аутентификации в родительскую зону. Это отличается от исходной спецификации DNS ([RFC1034]), в которой сказано, что на родительской стороне среза зоны могут присутствовать лишь наборы NS RRset.

Данная спецификация обновляет исходную спецификацию DNS, разрешая включать записи типов NSEC и DS на родительской стороне среза зон. Эти наборы являются полномочными для родительской зоны при их включении на родительской стороне среза.

2.7. Пример защищённой зоны

Полный пример небольшой подписанной зоны содержится в Приложении A.

3. Работа сервера

В этой главе описано поведение объектов, поддерживающих функции защищённого сервера имён. Во многих случаях такие функции являются частью работы защищённого рекурсивного сервера имён, но и к полномочным серверам имён предъявляются некоторые из этих требований. Функции, относящиеся к рекурсивным серверам имён, описаны в параграфе 3.2, функции полномочных серверов – в параграфе 3.1.

Термины SNAME, SCLASS и STYPE при обсуждении используются так же, как в [RFC1034].

Защищённый сервер имён **должен** поддерживать расширение размера сообщений EDNS0 ([RFC2671]), **должен** поддерживать сообщения размером по крайней мере 1220 октетов **(следует** также поддерживать сообщения размером 4000 октетов). Поскольку пакеты IPv6 могут фрагментироваться только исходным отправителем, защищённому серверу имён **следует** принять меры, чтобы дейтаграммы UDP, передаваемые по протоколу IPv6, были при необходимости фрагментированы до минимального значения IPv6 MTU, если не известно значение MTU для пути. Вопросы фрагментирования и размера пакетов подробно обсуждаются в документах [RFC1122], [RFC2460] и [RFC3226].

Защищённый сервер имён при получении запроса DNS, не включающего псевдозапись EDNS OPT, или имеющего сброшенный бит DO, **должен** трактовать записи RRSIG, DNSKEY и NSEC, как обычные наборы RRset и **недопустимо** выполнять для этих записей какие-либо дополнительные операции, описанные ниже. Поскольку тип DS RR имеет особое свойство быть единственным в точке делегирования в родительской зоне, записи DS RR всегда требуют специальной обработки, описанной в параграфе 3.1.4.1.

Защищённому серверу имён при получении явных запросов на связанные с защитой типы RR, которые соответствуют содержимому нескольких обслуживаемых сервером зон (например, записи NSEC и RRSIG ниже и выше точки делегирования, когда сервер является полномочным для обеих зон), следует обеспечивать самосогласованность. Поскольку отклик для каждого запроса всегда согласован, сервер имён **может** возвратить один из перечисленных вариантов отклика:

- наборы RRset выше точки делегирования;
- наборы RRset ниже точки делегирования;
- наборы RRset выше и ниже точки делегирования;
- пустой раздел answer (нет записей);
- какой-либо иной отклик;
- сообщение об ошибке.

DNSSEC выделяет два новых бита в заголовке сообщений DNS - CD[1] (проверка отключена) и AD[2] (аутентичные данные). Бит CD контролируется распознавателями – защищённый сервер имён **должен** копировать этот бит из запроса в соответствующий отклик. Бит AD контролируется серверами имён – защищённый сервер **должен** игнорировать значение этого бита в полученных запросах. Более детальное описание использования этих битов приведено в параграфах 3.1.6, 3.2.2, 3.2.3, 4 и 4.9.

Защищённым серверам имён, которые создают записи CNAME RR из DNAME RR, как описано в [RFC2672], **не следует** генерировать подписи для синтезированных CNAME RR.

3.1. Полномочные серверы имён

При получении имеющего отношение к делу запроса с установленным битом DO псевдозаписи EDNS ([RFC2671]), защищённый полномочный сервер имён для подписанной зоны **должен** включить дополнительные записи RRSIG, NSEC и DS, в соответствии с приведёнными ниже правилами:

- запись RRSIG RR, которая может использоваться для аутентификации отклика, **должна** включаться в отклик в соответствии с правилами параграфа 3.1.1;
- запись NSEC RR, которая может использоваться для аутентифицированного отказа, **должна** включаться в отклик в соответствии с правилами параграфа 3.1.3;
- набор DS RRset или запись NSEC RR, говорящая об отсутствии DS RR, **должна** включаться автоматически в соответствии с правилами параграфа 3.1.4.

Эти правила применимы только к откликам, семантика которых передаёт информацию о наличии или отсутствии записей RR. Т. е., эти правила не используются при генерации откликов типа RCODE 4 (Not Implemented[3]) или RCODE 5 (Refused[4]).

DNSSEC не меняет протокол переноса зон DNS. В параграфе 3.1.5 обсуждаются требования к переносу зон.

3.1.1. Включение записей RRSIG RR в отклик

При ответе на запрос, содержащий бит DO, защищённому полномочному серверу имён **следует** пытаться передать записи RRSIG RR, которые защищённый распознаватель может использовать для аутентификации наборов RRset в отклике. Серверу имён **следует** каждый раз пытаться сохранить в отклике RRset и связанные с ним записи RRSIG вместе. Включение в отклик записей RRSIG RR выполняется по приведённым ниже правилам:

- При включении подписанного набора RRset в раздел Answer сервер имён **должен** также включить соответствующие записи RRSIG RR в раздел Answer. Записи RRSIG RR имеют при включении более высокий

[1]Checking Disabled.
[2]Authentic Data.
[3]Не реализовано.
[4]Отказ.

приоритет, нежели прочие наборы RRset, которые также могут включаться в отклик. Если свободное пространство не позволяет включить записи RRSIG RR, сервер имён **должен** установить бит TC.

- При включении подписанного набора RRset в раздел Authority сервер имён **должен** также включить соответствующие записи RRSIG RR в раздел Authority. Записи RRSIG RR имеют при включении более высокий приоритет, нежели прочие наборы RRset, которые также могут включаться в отклик. Если свободное пространство не позволяет включить записи RRSIG RR, сервер имён **должен** установить бит TC.

- При включении подписанного набора RRset в раздел Additional сервер имён **должен** также включить соответствующие записи RRSIG RR в раздел Additional. Если свободное пространство не позволяет включить RRset и связанные с ним записи RRSIG RR, сервер имён может сохранить RRset, отбрасывая записи RRSIG RR. В таких случаях для сервера **недопустимо** устанавливать бит TC на основании лишь того, что записи RRSIG RR не поместились в отклик.

3.1.2. Включение записей DNSKEY RR в отклик

При отклики на запросы с установленным битом DO, запрашивающие записи SOA или NS на вершине подписанной зоны, защищённый полномочный сервер имён для этой зоны **может** возвращать расположенный на вершине зоны набор DNSKEY RRset в разделе Additional. В такой ситуации DNSKEY RRset и связанные с ним записи RRSIG RR имеют более низкий приоритет, нежели прочие данные, которые могут быть включены в раздел Additional. Серверу имён **не следует** включать набор DNSKEY RRset, если в сообщении недостаточно места для одновременного включения связанных с набором записей RRSIG RR. Если свободного пространства недостаточно для включения DNSKEY и записей RRSIG RR, сервер **должен** опустить их; **недопустимо** устанавливать бит TC лишь на том основании, что записи не помещаются в сообщение (см. параграф 3.1.1).

3.1.3. Включение записей NSEC RR в отклик

При ответе на запрос с установленным битом DO защищённый полномочный сервер имён для подписанной зоны **должен** включать записи NSEC RR в каждом из перечисленных случаев:

No Data - зона содержит наборы RRset, которые точно соответствуют <SNAME, SCLASS>, но не содержит ни одного набора, в точности соответствующего <SNAME, SCLASS, STYPE>.

Name Error - зона не содержит наборов, соответствующих <SNAME, SCLASS> в точности или с использованием шаблона.

Wildcard Answer - зона не содержит ни одного RRset, в точности соответствующего <SNAME, SCLASS>, но содержит RRset, соответствующий <SNAME, SCLASS, STYPE> с использованием шаблона.

Wildcard No Data - зона не содержит ни одного набора RRset, который точно соответствует <SNAME, SCLASS> и содержит один или несколько наборов RRset, соответствующих <SNAME, SCLASS> с использованием шаблона, но не содержит ни одного набора RRset, соответствующего <SNAME, SCLASS, STYPE> с использованием шаблона.

В каждом из этих случаев сервер имён включает в отклик записи NSEC RR, чтобы показать отсутствие в зоне точного соответствия для <SNAME, SCLASS, STYPE> и указать, что сервер имён возвращает корректные данные для зоны.

3.1.3.1. Включение записей NSEC RR - отклик No Data

Если зона содержит наборы RRset, соответствующие <SNAME, SCLASS>, но не содержит RRset, соответствующих <SNAME, SCLASS, STYPE>, сервер имён **должен** включить NSEC RR для <SNAME, SCLASS> вместе со связанными записями RRSIG RR в раздел отклика Authority (см. параграф 3.1.1). Если свободное пространство не позволяет включить запись NSEC RR или связанные с ней записи RRSIG RR, сервер имён **должен** установить бит TC (см. параграф 3.1.1).

Когда искомое имя существует, поиск по шаблону для такого запроса не применяется и достаточно одной подписанной записи NSEC RR, чтобы показать отсутствие запрошенного типа RR.

3.1.3.2. Включение записей NSEC RR - отклик Name Error

Если зона не содержит наборов RRset, соответствующих <SNAME, SCLASS> в точности или с использованием шаблона, сервер имён **должен** включить в раздел Authority перечисленные ниже записи NSEC RR вместе с соответствующими RRSIG RR:

- запись NSEC RR, показывающая отсутствие точного соответствия <SNAME, SCLASS>;

- запись NSEC RR, показывающая отсутствие в зоне наборов RRset, соответствующих <SNAME, SCLASS> с использованием шаблона.

В некоторых случаях достаточно бывает одной записи NSEC RR для обоих вариантов отсутствия. Тогда серверу имён **следует** включать в раздел Authority одну запись RR и связанные с ней RRSIG RR.

Если свободное пространство не позволяет включить записи NSEC и RRSIG, сервер имён **должен** установить бит TC (см. параграф 3.1.1).

Для имён владельцев в записях NSEC и RRSIG не используются шаблоны при включении таких RR в раздел Authority.

Отметим, что эта форма отклика включает ситуации, когда SNAME соответствует пустому нетерминальному имени в зоне (имя, которое не является именем владельца какого-либо набора RRset, но является родительским именем для одного или нескольких наборов RRset).

3.1.3.3. Включение записей NSEC RR - отклик Wildcard Answer

Если зона не содержит наборов RRset, в точности соответствующих <SNAME, SCLASS>, но содержит RRset, соответствующий <SNAME, SCLASS, STYPE> с использованием шаблона, сервер имён **должен** включить полученный с использованием шаблона ответ и соответствующие записи RRSIG RR в раздел Answer, а также **должен** включить в раздел Authority запись NSEC RR и связанные с ней записи RRSIG RR, показывающие, что зона не содержит более

точного соответствия <SNAME, SCLASS>. Если свободное пространство не позволяет включить записи NSEC и RRSIG RR, сервер имён **должен** установить бит TC (см. параграф 3.1.1).

3.1.3.4. Включение записей NSEC RR - отклик Wildcard No Data

Этот случай представляет собой комбинацию предыдущих. Зона не содержит имени, в точности соответствующего <SNAME, SCLASS>, и, хотя зона содержит наборы RRset, которые соответствуют <SNAME, SCLASS> с использованием шаблона, ни один из этих наборов не соответствует STYPE. Сервер имён **должен** включить в раздел Authority перечисленные ниже записи NSEC RR с соответствующими записями RRSIG RR:

- Запись NSEC RR, показывающую отсутствие наборов RRset, соответствующих STYPE с шаблоном имени владельца, которое соответствует <SNAME, SCLASS> при использовании шаблона.
- Запись NSEC RR, показывающую отсутствие в зоне наборов RRset, более точно соответствующих <SNAME, SCLASS>.

В некоторых случаях достаточно бывает одной записи NSEC RR для обоих вариантов отсутствия. Тогда серверу имён **следует** включать в раздел Authority одну запись RR и связанные с ней RRSIG RR.

К именам владельцев в записях NSEC и RRSIG не применяются шаблоны, когда эти записи включены в раздел отклика Authority.

Если свободное пространство не позволяет включить записи NSEC и RRSIG, сервер имён **должен** установить бит TC (см. параграф 3.1.1).

3.1.3.5. Поиск правильных записей NSEC RR

Как было отмечено выше, возникают ситуации, при которых защищённый полномочный сервер имён включает запись NSEC RR, сообщающую об отсутствии наборов RRset, которые соответствуют отдельному значению SNAME. Включение такой записи NSEC в полномочную зону относительно просто, по крайней мере на концептуальном уровне. В приведённом ниже обсуждении предполагается, что сервер имён является полномочным для зоны, в которой отсутствуют наборы RRset, соответствующие SNAME. Приведённый ниже алгоритм оптимизирован по размеру, но не по эффективности.

Чтобы найти запись NSEC, говорящую об отсутствии наборов RRset, соответствующих имени N в зоне Z, создаётся последовательность S, содержащая имена владельцев каждого набора RRset в Z, отсортированные в каноническом порядке ([RFC4034]) без дубликатов. Далее в списке находится имя M, которое непосредственно предшествовало бы N, если бы это имя содержалось в S. M будет именем владельца записи NSEC RR, которая подтверждает отсутствие наборов RRset с именем владельца N.

Алгоритм поиска записи NSEC RR, показывающей отсутствие соответствия заданному имени при использовании шаблона, похож на описанный, но требует выполнения дополнительного шага. Более точно, алгоритм поиска NSEC, обеспечивающий информацию об отсутствии RRset с совпадающим в точности именем шаблона, совпадает с алгоритмом поиска NSEC RR, который говорит об отсутствии RRset с любым другим именем владельца. Сообщающая об отсутствии часть представляет собой метод определения имени применимого несуществующего шаблона. На практике это просто, поскольку полномочный сервер имён уже проверен на предмет наличия в точности такого имени шаблона на этапе (1)(c) обычного алгоритма поиска, описанного в параграфе 4.3.2 документа [RFC1034].

3.1.4. Включение в отклик записей DS RR

При откликах на запросы с установленным битом DO защищённый полномочный сервер имён, возвращающий информацию, включает данные DNSSEC вместе с набором NS RRset.

Если в точке передачи полномочий присутствует DS RRset, сервер имён **должен** должен возвратить как DS RRset, так и связанные с ним записи (запись) RRSIG RR в разделе Authority вместе с NS RRset.

Если в точке делегирования нет DS RRset, сервер имён **должен** возвратить запись NSEC RR, говорящую об отсутствии DS RRset, и связанные с NSEC RR записи (запись) RR вместе с NS RRset. Сервер имён **должен** разместить NS RRset до NSEC RRset и связанных с этим набором записей (записи) RRSIG RR.

Включение записей DS, NSEC и RRSIG увеличивает размер referral-сообщений и может приводит к опусканию некоторых или всех склеивающих записей. Если свободное пространство не позволяет включить DS или NSEC RRset и связанные записи RRSIG RR, сервер имён **должен установить бит** TC (см. параграф 3.1.1).

3.1.4.1. Отклики на запросы записей DS RR

Записи DS необычны в том смысле, что они появляются только на родительской стороне среза зон. Например, DS RRset для делегирования foo.example сохраняется в зоне example, а не в зоне foo.example. Это требует использования специальных правил обработки как на серверах, так и на распознавателях, поскольку сервер имён для дочерней зоны является полномочным для имени на срезе зоны в соответствии с обычными правилами DNS, но дочерняя зона не содержит DS RRset.

Знающий о защите распознаватель передаёт запросы в родительскую зону для поиска нужной DS RR в точке делегирования (см. параграф 4.2). Однако требуются специальные правила для предотвращения путаницы с не знающими о защите распознавателями, которые могут начать обработку такого отклика (например, в сети, где конфигурация вынуждает знающий о защите распознаватель направлять запросы через незащищённый сервер имён). Остальная часть этого параграфа описывает обработку запросов DS защищённым сервером имён с целью предотвращения описанной проблемы.

Необходимость специальной обработки защищённым сервером имён возникает только при выполнении всех перечисленных ниже условий:

- сервер имён получил запрос для DS RRset на срезе зоны;
- сервер имён является полномочным для дочерней зоны;

- сервер имён не является полномочным для родительской зоны;
- сервер имён не предлагает рекурсии.

Во всех остальных случаях у сервера есть какой-то способ получения DS RRset или не следует ожидать получения DS RRset даже с использованием обычных (до DNSSEC) правил обработки, поэтому сервер может вернуть либо значение DS RRset, либо отклик об ошибке, возникшей при обычной обработке.

Если все указанные выше условия выполняются, однако сервер имён уполномочен для SNAME, но не поддерживает запрошенный RRset, этот сервер имён **должен** возвращать полномочный отклик no data, показывающий, что DS RRset не существует на вершине дочерней зоны. Пример такого отклика представлен в Приложении B.8.

3.1.5. Отклики на запросы типа AXFR или IXFR

DNSSEC не меняет процесс переноса зон DNS. Подписанная зона будет включать записи RRSIG, DNSKEY, NSEC и DS, но эти записи не имеют какого-либо специального значения в контексте операций переноса зон.

Для проверки корректности подписи зоны перед её отправкой или восприятием в процессе переноса не требуется полномочный сервер имён. Однако полномочный сервер **может** принять решение об отказе от переноса зоны целиком, если эта зона не соответствует требованиям к подписи, указанным в разделе 2. Основной целью переноса зон является обеспечение идентичности копий зоны на всех полномочных серверах имён. Полномочному серверу, который принял решение о самостоятельной проверке зоны, **недопустимо** отвергать некоторые RR и воспринимать другие.

Наборы DS RRset появляются только на родительской стороне среза зоны и являются полномочными данными в родительской зоне. Как и любые другие полномочные RRset, наборы DS RRset **должны** включаться в перенос зон, для которых эти RRset являются полномочными данными. В случае DS RRset это будет родительская зона.

Записи NSEC RR присутствуют на срезе зон как в родительской, так и в дочерней зоне и являются полномочными для обеих. Записи NSEC RR в родительской и дочерней зонах никогда не являются идентичными, поскольку NSEC RR на вершине дочерней зоны всегда указывает присутствие SOA RR дочерней зоны, тогда как родительская NSEC RR на срезе зоны никогда не указывает наличия SOA RR. Как и все прочие полномочные RR, записи NSEC RR **должны** включаться в перенос зон, для которых они служат полномочными данными. Родительская NSEC RR на срезе зоны **должна** включаться в перенос родительской зоны, а NSEC на вершине дочерней зоны **должна** включаться в перенос дочерней зоны.

Записи RRSIG RR присутствуют на срезе зон как в родительской, так и в дочерней зоне и являются полномочными для зон, содержащих полномочные RRset, для которых RRSIG RR обеспечивают подписи. Т. е., RRSIG RR для DS RRset или родительской NSEC RR на срезе зоны будет полномочной в родительской зоне, а RRSIG для любых RRset на вершине дочерней зоны будут полномочными для дочерней зоны. Родительские и дочерние записи RRSIG RR на срезе зоны никогда не являются идентичными, поскольку поле Signer's Name в RRSIG RR на вершине дочерней зоны будет указывать DNSKEY RR на вершине дочерней зоны, тогда как аналогичное поле родительской RRSIG RR на срезе зоны будет указывать DNSKEY RR на вершине родительской зоны. Как и все прочие полномочные RR, записи RRSIG RR **должны** включаться в перенос зоны, для которой они являются полномочными.

3.1.6. Биты AD и CD в полномочных откликах

Флаги CD и AD предназначены для использования при взаимодействии защищённых распознавателей и рекурсивных серверов имён. Эти биты по большей части не имеют отношения к обработке запросов защищёнными полномочными серверами имён.

Осведомленные о защите серверы имён не проверяют подпись для полномочных данных в процессе обработки запросов даже при сброшенном бите CD. Таким серверам **следует** сбрасывать флаг CD при генерации полномочных откликов.

Защищённому серверу имён **недопустимо** устанавливать бит AD в откликах, пока сервер не рассмотрит все RRset в разделах Answer и Authority на предмет аутентичности. Локальная политика таких серверов **может** считать данные из полномочной зоны аутентичными без дополнительной проверки. Однако серверам имён **недопустимо** такое поведение, если полномочная зона не была получена с использованием мер защиты (таких, как защищённый механизм переноса зон), а также **недопустимо** делать это без явного указания такого поведения в конфигурации.

Защищённые серверы имён, которые поддерживают рекурсию, **должны** следовать правилам для битов CD и AD, указанным в параграфе 3.2, при генерации откликов, содержащих данные, которые были получены с использованием рекурсии.

3.2. Рекурсивные серверы имён

Как разъяснено в [RFC4033], защищённый сервер имён имеет элемент, действующий в ролях защищённого сервера имён и защищённого распознавателя. В этом параграфе термины «на стороне сервера имён» (name server side) и «на стороне распознавателя» (resolver side) относятся к коду защищённого сервера имён, который исполняет роль защищённого сервера и распознавателя, соответственно.

На стороне распознавателя применяются обычные правила кэширования и негативного кэширования, используемые на всех защищённых распознавателях.

3.2.1. Бит DO

Защищённый рекурсивный сервер имён в роли распознавателя **должен** устанавливать флаг DO при отправке запросов независимо от состояния этого бита в исходном запросе, полученном на стороне сервера имён. Если бит DO в исходном запросе не установлен, на стороне сервера имён из отклика **должны** вырезаться все аутентификационные записи DNSSEC RR, но **недопустимо** вырезание каких-либо типов DNSSEC RR, явно указанных в исходном запросе.

3.2.2. Бит CD

Бит CD служит для того, чтобы позволить защищённому распознавателю отключить проверку подписи на защищённом сервере имён, обрабатывающем конкретный запрос.

Сервер имён **должен** копировать бит CD из запроса в соответствующий отклик.

В серверной роли защищённый рекурсивный сервер имён **должен** передавать состояние бита CD компоненте (роли) распознавателя вместе с остальной частью исходного запроса, чтобы на стороне распознавателя было известно, требуется ли проверять данные, которые будут возвращаться на сторону сервера имён. Установленный бит CD показывает, что передавший запрос распознаватель хочет выполнять аутентификационную проверку, требуемую его локальной политикой. Таким образом, на стороне распознавателя в рекурсивном сервере имён не требуется выполнять аутентификацию наборов RRset в отклике. При установленном флаге CD рекурсивному серверу имён **следует**, по возможности, возвращать запрошенные данные исходному распознавателю даже в тех случаях, когда локальная политика аутентификации рекурсивного сервера имён требует отвергать рассматриваемые записи. Т. е., путём установки бита CD исходный распознаватель указывает, что он принимает на себя ответственность за аутентификацию и рекурсивному серверу не следует об этом думать.

Если на стороне распознавателя реализован кэш BAD (см. параграф 4.7) и на стороне сервера имён получен запрос, соответствующий записи в кэше BAD на стороне распознавателя, отклик серверной стороны зависит от состояния бита CD в исходном запросе. При установленном бите CD серверу имён **следует** возвращать данные из кэша BAD, а при сброшенном сервер имён **должен** возвращать RCODE 2 (сбой сервера).

Указанное выше правило предназначено для того, чтобы предоставлять не разобранные (raw) данные клиентам, которые способны самостоятельно проверить подпись, сохраняя возможность выполнения проверки на стороне распознавателя в защищённом рекурсивном сервере имён для тех клиентов, которым такая проверка нужна. Отказ при проверке подписи на сервере может возникать, например, в ситуациях, когда рекурсивный сервер имён и клиент находятся в разных условиях. Например, рекурсивный сервер имён может быть некорректно настроен или у клиента может быть определённая информация об имеющем отношение к делу «островке» безопасности, которой нет у рекурсивного сервера имён. В таких случаях «защита» клиента, которую он может обеспечить, самостоятельно проверяя подпись, от данных, которые представляются серверу «плохими» ничем не поможет этому клиенту.

3.2.3. Бит AD

На стороне сервера имён защищённого рекурсивного сервера имён **недопустимо** устанавливать бит AD в отклике, пока сервер имён не считает все наборы RRset в разделах Answer и Authority подлинными. На серверной стороне **следует** устанавливать бит AD тогда и только тогда, когда на стороне распознавателя все RRset в разделе Answer и все имеющиеся RR с негативными откликами в разделе Authority признаны подлинными. На стороне распознавателя **должна** выполняться описанная в разделе 5 процедура для решения вопроса о подлинности рассматриваемых записей RR. Однако для совместимости со старыми версиями рекурсивный сервер имён **может** устанавливать бит AD, когда отклик включает неподписанные записи CNAME RR, если эти CNAME RR явно могут быть синтезированы из подлинной записи DNAME RR, которая также включена в отклик в соответствии с правилами синтеза, описанными в [RFC2672].

3.3. Примеры откликов DNSSEC

Примеры пакетов откликов приведены в Приложении B.

4. Преобразование имён

В этом разделе рассматривается поведение элементов с функциями защищённого распознавателя. Во многих случаях такие функции могут быть частью функциональности защищённого рекурсивного сервера имён, но и автономные распознаватели поддерживают большую часть таких функций. Специфические для защищённых рекурсивных серверов имён функции описаны в параграфе 3.2.

4.1. Поддержка EDNS

Защищённый распознаватель **должен** включать EDNS ([RFC2671]) OPT псевдо-RR с установленным при отправке отправке запросов битом DO ([RFC3225]).

Защищённый распознаватель **должен** поддерживать сообщения размером не менее 1220 октетов, **следует** также поддерживать сообщения размером до 4000 октетов, а также распознаватель **должен** использовать поле sender's UDP payload size в EDNS OPT псевдо-RR для анонсирования размера сообщений, которые он будет принимать. Уровень IP защищённых распознавателей **должен** корректно обрабатывать фрагментированные пакеты UDP независимо от того, были такие фрагментированные пакеты получены по IPv4 или IPv6. Эти требования были рассмотрены в [RFC1122], [RFC2460] и [RFC3226].

4.2. Поддержка верификации подписей

Защищённый распознаватель **должен** поддерживать механизмы проверки подписей, описанные в разделе 5 и эти механизмы **следует** применять к каждому полученному отклику, за исключением перечисленных случаев:

- распознаватель является частью защищённого рекурсивного сервера имён и отклик является результатом рекурсии для выполнения запроса, полученного с установленным битом CD;
- отклик является результатом запроса, созданного напрямую через прикладной интерфейс и указавшего защищённому распознавателю необходимость отказа от проверки для этого запроса;
- проверка для данного запроса отключена в соответствии с локальной политикой.

Поддержка проверки подписей в защищённом распознавателе **должна** включать возможность проверки и для шаблонных имён владельцев.

Защищённые распознаватели **могут** запрашивать отсутствующие защитные записи RR при попытках проверки - реализации, выбравшие такой подход, должны быть готовы к тому, что полученного ответа будет не достаточно для проверки исходного отклика. Например, обновление зоны может привести к изменению (или удалению) желаемой информации в интервале между исходным и последующим запросами.

При попытке отыскать отсутствующие записи NSEC RR, которые находятся на родительской стороне среза зоны, защищённый итеративный распознаватель **должен** запрашивать серверы имён для родительской, а не дочерней зоны.

При попытке отыскать отсутствующие записи DS защищённый итеративный распознаватель **должен** запрашивать серверы имён для родительской, а не дочерней зоны. Как разъяснено в параграфе 3.1.4.1, защищённые серверы имён должны применять специальные правила обработки DS RR и в некоторых ситуациях распознаватель также должен применять специальные правила, чтобы найти серверы имён для родительской зоны, если у распознавателя ещё нет родительского NS RRset. Для нахождения родительского NS RRset распознаватель может начать с имени делегирования (delegation name), исключить из него левую метку и отправить запрос для NS RRset по этому имени. Если для этого имени не будет возвращён набор NS RRset, распознаватель вырезает ещё одну метку слева и повторяет запрос. Описанная процедура повторяется до тех пор, пока не будет найден NS RRset или закончатся метки.

4.3. Определение статуса защиты данных

Защищённый распознаватель **должен** быть способен определить, следует ли ожидать наличия подписи для конкретного набора RRset. Точнее говоря, такой распознаватель должен различать перечисленные ниже случаи:

Защищенный (Secure) - RRset, для которого распознаватель способен построить цепочку подписанных записей DNSKEY и DS RR от доверенной защитной привязки (security anchor) до RRset. В этом случае набору RRset следует быть подписанным и для него выполняется проверка подписи, описанная выше.

Незащищённый (Insecure) - RRset, для которого распознаватель знает об отсутствии цепочки подписанных записей DNSKEY и DS RR от любой доверенной стартовой точки до RRset. Это может наблюдаться в тех случаях, когда целевой набор RRset находится в неподписанной зоне или потомке такой зоны. В этом случае набор RRset может быть как подписанным, так и неподписанным и распознаватель не сможет проверить подпись.

Подделка (Bogus) - RRset, для которого распознаватель предполагает возможность установить цепочку доверия, но не может сделать этого по причине того или иного отказа при проверке подписи или отсутствия данных, наличие которых указывают имеющие отношение к делу записи DNSSEC RR. Это может говорить об атаке, ошибке в конфигурации или повреждении данных.

Неопределённость (Indeterminate) - RRset, для которого распознаватель не может определить необходимость наличия подписи, по причине невозможности получить требуемые записи DNSSEC RR. Это может происходить в тех случаях, когда распознаватель не может контактировать с осведомленными о защите серверами имён для соответствующих зон.

4.4. Заданные в конфигурации доверенные привязки

Защищённый распознаватель **должен** обеспечивать возможность задания в конфигурации хотя бы одного доверенного открытого ключа или DS RR, а также **следует** поддерживать возможность задания в конфигурации множества доверенных открытых ключей или DS RR. Поскольку защищённый распознаватель не сможет проверить подписи без такой доверенной привязки, ему **следует** поддерживать некий отказоустойчивый механизм получения таких ключей в процессе загрузки - примерами таких механизмов могут служить сохранение привязок в энергонезависимой памяти (например, на диске) или использование доверенного механизма локальной настройки конфигурации.

Отметим, что доверенные привязки покрывают также ключевой материал, обновляемый защищённым путём, который может включать ту или иную физическую среду, протокол обмена ключами или иные способы обмена по отдельному каналу (out-of-band).

4.5. Кэширование откликов

Защищённому распознавателю **следует** кэшировать каждый отклик, как неделимый элемент, содержащий целиком ответ, включая именованный набор RRset и все связанные с ним записи DNSSEC RR. По истечении срока действия записи распознавателю **следует** отбрасывать её целиком. В большинстве случаев подходящим индексом для таких записей в кэше будет служить тройка <QNAME, QTYPE, QCLASS>, но для откликов, описанных в параграфе 3.1.3.2, подходящим индексом будет служить пара <QNAME,QCLASS>.

Причина таких рекомендаций заключается в том, что между исходным запросом и завершением срока действия данных в кэше полномочные данные могут измениться (например, при динамическом обновлении).

Описанное выше имеет отношение к двум ситуациям.

1. Используя запись RRSIG, можно сделать вывод, что ответ был синтезирован из шаблона. Защищённый рекурсивный сервер имён может сохранять эти шаблонные данные и применять их для генерации позитивных откликов на запросы для имён, отличающегося от того имени, для которого был получен исходный ответ.
2. Записи NSEC RR с подтверждением отсутствия имени, могут повторно использоваться защищённым распознавателем для подтверждения отсутствия любого имени в охватываемом записью диапазоне.

Теоретически распознаватель может использовать шаблоны или записи NSEC RR для генерации позитивных и негативных откликов (соответственно), пока не истечёт TTL или срок действия подписей соответствующих записей. Тем не менее для распознавателей представляется целесообразным избегать блокирования новых полномочных данных или синтеза новых данных на основе имеющихся у него. Следующие этим рекомендациям распознаватели будет иметь более согласованную картину пространства имён.

4.6. Обработка битов CD и AD

Защищённый распознаватель **может** установить в запросе бит CD для того, чтобы показать свою ответственность за выполнение аутентификации, которой его локальная политика требует для наборов RRset в отклике. Влияние установки этого флага на поведение защищённого рекурсивного сервера имён описано в параграфе 3.2.

Защищённый распознаватель **должен** MUST сбрасывать бит AD при создании запросных сообщений с целью защиты от серверов имён с ошибками, которые вслепую копируют непонятные биты заголовка из сообщений с запросами в сообщения откликов.

Защищённый распознаватель **должен** игнорировать биты CD и AD в откликах, если эти отклики не были получены с использованием защищённого канала или в конфигурации распознавателя не задана трактовка битов заголовка даже при использовании незащищённого канала.

4.7. Кэширование неприемлемых данных

Многие ошибки валидации являются временными, но некоторые могут быть постоянными - например, административные ошибки (отказ повторно подписать зону, рассогласование часов и т. п.). Поскольку в случаях постоянной ошибки повторный запрос не решает проблему, проверяющие распознаватели могут создавать значительный ненужный трафик DNS, повторяя запросы для наборов RRsets с постоянным отказом при валидации.

Для предотвращения такого избыточного трафика DNS защищённые распознаватели **могут** кэшировать (с некоторыми ограничениями) данные с неприемлемыми подписями.

Концептуально кэширование таких данных похоже на кэширование негативных откликов ([RFC2308]), отличаясь лишь тем, что вместо кэширования приемлемого негативного отклика распознаватель кэширует факт отказа при проверке конкретного ответа. В этом документе кэш данных с неприемлемой подписью называется BAD-кэшем.

Распознаватели, реализующие BAD-кэширование, **должны** принимать меры предотвращения использования такого кэша для организации атак на отказ служб (DoS), включая перечисленные ниже действия.

- поскольку наборы RRset, для которых проверка привела к отказу, не имеют надёжных значений TTL, реализация **должна** присвоить записи значение TTL и это значение **следует** делать достаточно малым, чтобы снизить эффект кэширования результатов атак;
- для предотвращения кэширования временных отказов при проверке (они могут быть результатом атаки) распознавателям **следует** отслеживать запросы, приводящие к отказам при проверке и отклики из BAD-кэша **следует** возвращать только при достижении определённого числа откликов на запросы, не прошедших проверку, для конкретного набора <QNAME, QTYPE, QCLASS>.

Распознавателям **недопустимо** возвращать наборы RRset из BAD-кэша, если от распознавателя не требуется проверка подписей для интересующих наборов RRset в соответствии с правилами параграфа 4.2. Обсуждение взаимодействия откликов, возвращаемых защищёнными рекурсивными серверами имён, с BAD-кэшем приведено в параграфе 3.2.2.

4.8. Синтезированные записи CNAME

Выполняющий проверку (валидацию) защищённый распознаватель **должен** трактовать, подписи с корректно подписанной DNAME RR, как покрывающие и неподписанные записи CNAME RR, которые могут быть синтезированы из DNAME RR, как описано в [RFC2672], по крайней мере не отвергая сообщения откликов исключительно на основе наличия в них записей CNAME RR. Распознаватель **может** (не не обязан) сохранять такие записи CNAME RR в своём кэше или в ответах, которые он возвращает.

4.9. Оконечные распознаватели

Защищённый оконечный распознаватель **должен** поддерживать типы DNSSEC RR, по крайней мере не отвергая отклики лишь на основании наличия записей DNSSEC RR.

4.9.1. Обработка бита DO

Не выполняющий проверку защищённый оконечный распознаватель **может** (но не обязан) включать записи DNSSEC RR, возвращённые защищённым рекурсивным сервером имён, как часть данных, которые этот распознаватель возвращает обратившемуся к нему приложению. Если распознаватель решает делать это, ему требуется установить флаг DO для получения записей DNSSEC RR от рекурсивного сервера имён.

Выполняющий проверку защищённый оконечный распознаватель **должен** установить флаг DO, поскольку в противном случае он не получит записей DNSSEC RR, требуемых для проверки подписи.

4.9.2. Обработка бита CD

Не выполняющему проверку защищённому оконечному распознавателю **не следует** устанавливать флаг CD при отправке запросов, если такая установка не была запрошена приложением, поскольку по определению такой распознаватель зависит от выполнения проверки защищённым сервером имён от его имени.

Выполняющему проверку защищённому оконечному распознавателю **следует** устанавливать бит CD, так как в противном случае защищённый рекурсивный сервер имён будет отвечать на запрос, используя свою локальную политику, которая может блокировать получение оконечным распознавателем части данных, разрешённых его политикой.

4.9.3. Обработка бита AD

Не выполняющий проверку защищённый оконечный распознаватель **может** выбрать проверку состояния бита AD в получаемых откликах с целью определения факта криптографической верификации передавшим сообщение защищённым рекурсивным сервером имён данных в разделах отклика Answer и Authority. Отметим, однако, что

получаемые защищённым оконечным распознавателем отклики существенно зависят от локальной политики защищённого рекурсивного сервера имён. Поэтому проверка состояния флага AD может не иметь большого практического значения, за исключением помощи в отладке. В любом случае защищённому оконечному распознавателю **недопустимо** полагаться на проверку подписи, выполненную, якобы, от его имени, за исключением ситуаций, когда защищённый оконечный распознаватель получает соответствующие данные от доверенного защищённого рекурсивного сервера имён по защищённому каналу.

Выполняющему проверку защищённому оконечному распознавателю **не следует** проверять состояние бита AD в откликах, поскольку он, по определению, выполняет свою проверку подписи, независимо от состояния флага AD.

5. Аутентификация откликов DNS

Для использования записей DNSSEC RR при аутентификации защищённый распознаватель должен быть настроен так, чтобы он знал хотя бы одну аутентифицированную запись DNSKEY или DS. Процесс организации и аутентификации этой начальной цепочки доверия обеспечивается тем или иным внешним механизмом. Например, распознаватель может использовать тот или иной специальный канал для аутентификационного обмена, позволяющий получить запись DNSKEY RR для зоны или запись DS RR, которая идентифицирует и аутентифицирует DNSKEY RR для зоны. Далее в этом разделе предполагается, что распознаватель уже получил тем или иным способом набор начальных доверенных привязок.

Исходную запись DNSKEY RR можно использовать для аутентификации набора DNSKEY RRset на вершине зоны. Для идентификации DNSKEY RRset на вершине зоны с использованием начального ключа распознаватель **должен**:

1. убедиться в том, что исходная запись DNSKEY RR присутствует на вершине DNSKEY RRset и для этой записи DNSKEY RR установлен флаг Zone Key (бит 7 в DNSKEY RDATA bit 7);

2. убедится в наличии той или иной записи RRSIG RR, покрывающей вершину DNSKEY RRset, а также в том, что комбинация RRSIG RR и исходной DNSKEY RR аутентифицирует набор DNSKEY RRset (процесс использования RRSIG RR для аутентификации RRset описан в параграфе 5.3).

После того, как распознаватель подтвердил подлинность набора DNSKEY RRset на вершине зоны, используя начальную запись DNSKEY RR, делегирования из этой зоны можно аутентифицировать с использованием записей DS RR. Это позволяет распознавателю начать со стартового ключа и использовать наборы DS RRset для рекурсивного перемещения по дереву DNS вниз, получая наборы DNSKEY RRset на других вершинах. Если в конфигурации распознавателя задана корневая запись DNSKEY RR и каждое делегирование имеет связанную с ним запись DS RR, этот распознаватель может получить и проверить набор DNSKEY RRset на любой вершине. Процесс использования записей DS RR для аутентификации отсылок описан в параграфе 5.2.

В параграфе 5.3 показано, как распознаватель может использовать записи DNSKEY RR на вершине DNSKEY RRset и записи RRSIG RR из зоны для аутентификации любых других наборов RRset в зоне, если он имеет аутентифицированный набор DNSKEY RRset для вершины зоны. В параграфе 5.4 показан, как распознаватель может использовать аутентифицированные наборы NSEC RRset из зоны для подтверждения отсутствия RRset в зоне.

Когда распознаватель указывает поддержку DNSSEC (устанавливая флаг DO), защищённому серверу имён следует предпринять попытку обеспечить в отклике требуемые DNSKEY, RRSIG, NSEC и DS RRset (см. раздел 3). Однако защищённый распознаватель может продолжать получать отклики без соответствующих DNSSEC RR в результате конфигурационных проблем типа восходящего рекурсивного сервера имён, игнорирующего защиту, который непреднамеренно конфликтует с записями DNSSEC RR, или в результате преднамеренной атаки, когда злоумышленники будут вырезать записи DNSSEC RR из откликов или менять запрос таким образом, чтобы записи DNSSEC RR представлялись не запрошенными. Отсутствие данных DNSSEC в отклике, само по себе, **недопустимо** трактовать, как индикацию отсутствия аутентификационной информации.

Распознавателю **следует** ожидать аутентификационной информации от подписанных зон. Распознавателю **следует** считать, что зона является подписанной, если в конфигурации распознавателя имеются сведения об открытом ключе для этой зоны или её родительская зона является подписанной и делегирование из той родительской зоны содержит набор DS RRset.

5.1. Островки безопасности

Островки безопасности (см. [RFC4033]) являются подписанными зонами, для которых невозможно создание цепочки аутентификации от зоны до её родителя. Проверка подписей внутри островка безопасности требует наличия у проверяющего тех или иных путей получения начального ключа аутентифицированной зоны для островка. Если проверяющий не может получить такой ключ, ему **следует** переключиться в режим, при котором зоны внутри островка безопасности считаются не подписанными.

К островкам безопасности применимы все обычные процессы проверки откликов. Единственным отличием является способ получения проверяющим доверенной привязки для цепочки аутентификации.

5.2. Аутентификация отсылок

После аутентификации вершины DNSKEY RRset для подписанной родительской зоны наборы DS RRset могут применяться для аутентификации делегирования подписанной дочерней зоны. Запись DS RR идентифицирует DNSKEY RR в наборе DNSKEY RRset на вершине дочерней зоны и содержит криптографический отпечаток записи DNSKEY RR дочерней зоны. Использование строгого криптографического алгоритма для создания отпечатков гарантирует невозможность расчётным путём создать злоумышленнику запись DNSKEY RR, соответствующую отпечатку. Таким образом, аутентификационный отпечаток позволяет распознавателю проверит подлинность DNSKEY RR. После этого распознаватель может применять эту дочернюю DNSKEY RR для аутентификации всего набора DNSKEY RRset вершины дочерней зоны.

На основе DS RR для делегирования набор DNSKEY RRset на вершине дочерней зоны может считаться подлинным, если выполняются все перечисленные ниже условия.

- Запись DS RR была аутентифицирована с использованием той или иной записи DNSKEY RR из набора DNSKEY RRset на вершине родительской зоны (см. параграф 5.3).
- Поля Algorithm и Key Tag в записи DS RR соответствуют аналогичным полям в записи DNSKEY RR из набора DNSKEY RRset на вершине дочерней зоны и при хэшировании имени владельца DNSKEY RR и RDATA с использованием алгоритма, указанного в поле Digest Type записи DS RR, результирующий отпечаток соответствует полю Digest в записи DS RR.
- Соответствующая запись DNSKEY RR в дочерней зоне имеет установленный бит Zone, соответствующий секретный ключ имеет подписанный набор DNSKEY RRset вершины дочерней зоны и результирующая запись RRSIG RR аутентифицирует набор DNSKEY RRset на вершине дочерней зоны.

Если отсылка из родительской зоны не содержит DS RRset, отклику следует включать подписанный набор NSEC RRset, подтверждающий наличие DS RRset для делегированного имени (см. параграф 3.1.4).Защищённый распознаватель **должен** запрашивать серверы имён для родительской зоны набора DS RRset, если отсылка не включает ни DS RRset, ни NSEC RRset, подтверждающий существование DS RRset (см. раздел 4).

Если проверяющий аутентифицирует набор NSEC RRset, который подтверждает отсутствие DS RRset для этой зоны, это говорит об отсутствии пути аутентификации от родителя к потомку. Если у распознавателя имеется начальная запись DNSKEY или DS RR, относящаяся к дочерней зоне или любой точке делегирования ниже дочерней зоны, такая запись DNSKEY или DS RR **может** быть использована для организации аутентификационного пути. Если такой записи нет, проверяющий не сможет аутентифицировать наборы RRset в дочерней зоне и ниже её.

Если проверяющий не поддерживает ни одного из алгоритмов, указанных в аутентифицированном наборе DS RRset, это означает, что проверяющий не имеет поддерживаемого аутентификационного пути от родителя к потомку. Распознавателю следует трактовать такую ситуацию так же, как наличие аутентифицированного набора NSEC RRset, подтверждающего отсутствие DS RRset (см. выше).

Отметим, что для подписанного делегирования имеются две записи NSEC RR, связанные с именем делегирования. Одна NSEC RR размещается в родительской зоне и может использоваться для проверки наличия набора DS RRset для делегированного имени. Вторая запись NSEC RR размещается в дочерней зоне и указывает какие наборы RRset присутствуют на вершине этой зоны. Родительская и дочерняя записи NSEC RR в любом случае могут различаться, поскольку бит SOA будет установлен для дочерней NSEC RR и сброшен для родительской. Защищённый распознаватель **должен** использовать родительскую запись NSEC RR при попытке проверки отсутствия набора DS RRset.

Если распознаватель не поддерживает ни одного алгоритма, указанного в DS RRset, он не сможет проверить путь аутентификации к дочерней зоне. В таких случаях распознавателю **следует** трактовать дочернюю зону, как не подписанную.

5.3. Аутентификация RRset с помощью RRSIG RR

Проверяющий может использовать запись RRSIG RR и соответствующую ей DNSKEY RR для попытки аутентифицировать наборы RRset. Валидатор сначала проверяет RRSIG RR,чтобы убедиться в том, что эта запись покрывает RRset, срок её действия не истек и она идентифицирует приемлемую запись DNSKEY RR. Затем проверяющий создаёт каноническую форму подписанных данных, добавляя в конце RRSIG RDATA (за исключением поля Signature) с канонической формой охватываемого набора RRset. В заключение проверяющий использует открытый ключ и подпись для аутентификации подписанных данных. Подробные описания всех этапов приведены в параграфах 5.3.1, 5.3.2 и 5.3.3.

5.3.1. Проверка корректности RRSIG RR

Защищённый распознаватель **может** использовать запись RRSIG RR для аутентификации RRset, если выполняются все перечисленные ниже условия:

- RRSIG RR и RRset **должны** иметь одинаковые имя владельца и класс;
- поле Signer's Name записи RRSIG RR **должно** совпадать с именем зоны, содержащей RRset;
- поле Type Covered записи RRSIG RR **должно** совпадать с полем типа в наборе RRset;
- число меток в имени владельца RRset **должно** быть не меньше значения поля Labels в RRSIG RR;
- Текущее время у проверяющего **должно** быть не больше значения поля Expiration в RRSIG RR;
- Текущее время у проверяющего **должно** быть не меньше значения поля Inception в RRSIG RR;
- поля Signer's Name, Algorithm и Key Tag в записи RRSIG RR **должны** совпадать с именем владельца, алгоритмом и тегом ключа для той или иной DNSKEY RR в наборе DNSKEY RRset на вершине зоны;
- соответствующая запись DNSKEY RR **должна** присутствовать в наборе DNSKEY RRset на вершине зоны и **должна** иметь установленный флаг Zone (бит 7 в DNSKEY RDATA Flag).

Вполне возможно наличие нескольких DNSKEY RR, соответствующих приведённым выше условиям. В этом случае проверяющий не может заранее определить, какую из записей DNSKEY RR использовать для аутентификации подписи и он **должен** проверять все соответствующие условиям DNSKEY RR, пока подлинность подписи не будет подтверждена или соответствующие открытые ключи не закончатся.

Отметим, что этот процесс аутентификации имеет смысл только в том случае, когда проверяющий аутентифицирует запись DNSKEY RR для её использования для проверки подписей. Соответствующая запись DNSKEY RR считается подлинной при выполнении любого из приведённых ниже условий:

- набор DNSKEY RRset на вершине зоны, содержащий DNSKEY RR считается подлинным;

– набор RRset, охватываемый записью RRSIG RR, является вершиной самого DNSKEY RRset, а DNSKEY RR соответствует аутентифицированной DS RR из родительской зоны или доверенной привязке.

5.3.2. Реконструкция подписанных данных

После того, как проверено соответствие RRSIG RR требованиям, описанным в параграфе 5.3.1, проверяющий восстанавливает исходные подписанные данные. Эти данные включают RRSIG RDATA (без поля Signature) и каноническую форму RRset. Помимо изменения порядка, каноническая форма RRset может отличаться от полученного набора RRset за счёт сжатия имён DNS, декрементирования значений TTL, преобразования шаблонов. Проверяющему следует использовать при восстановлении исходных подписанных данных форму

```
signed_data = RRSIG_RDATA | RR(1) | RR(2)...
```

где | обозначает конкатенацию, RRSIG_RDATA - поля RRSIG RDATA в формате передачи, за исключением поля Signature и канонического представления Signer's Name.

```
RR(i) = name | type | class | OrigTTL | RDATA length | RDATA
```

Значение name в соответствии с приведённой ниже функцией

class - класс RRset;

type - тип RRset и всех RR в классе;

OrigTTL - значение поля Original TTL из RRSIG;

все имена в поле RDATA имеют каноническую форму;

Все RR(i) сортируются в каноническом порядке.

Для расчёта значение name выполняются перечисленные ниже действия:

```
rrsig_labels = значение поля Labels из записи RRSIG;
fqdn = каноническая форма полного доменного имени RRset;
fqdn_labels = счётчик меток в значении fqdn;
если rrsig_labels = fqdn_labels,
name = fqdn
если rrsig_labels < fqdn_labels,
name = "*." | самые правые метки rrsig_label в fqdn
если rrsig_labels > fqdn_labels
RRSIG RR не проходит требуемых проверок приемлемости и
эту запись НЕДОПУСТИМО использовать для аутентификации
данного набора RRset.
```

Канонические формы для имён и наборов RRset определены в [RFC4034].

Наборы NSEC RRset на границах делегирования требуют специальной обработки. Есть два разных набора NSEC RRset, связанных с подписанным делегированным именем. Один из NSEC RRset размещается в родительской зоне и указывает, какие наборы RRset присутствуют в родительской зоне. Второй набор NSEC RRset размещается в дочерней зоне и указывает, какие RRset присутствуют на вершине дочерней зоны. Родительский и дочерний наборы NSEC RRset всегда различаются, поскольку только дочерняя NSEC RR будет указывать наличие для имени набора SOA RRset. При восстановлении исходного NSEC RRset для точки делегирования из родительской зоны записи NSEC RR **недопустимо** комбинировать с записями NSEC RR из дочерней зоны. При восстановлении исходного набора NSEC RRset для вершины дочерней зоны записи NSEC RR **недопустимо** объединять с записями NSEC RR из родительской зоны.

Отметим, что каждый из двух наборов NSEC RRset у точки делегирования имеет соответствующую запись RRSIG RR с именем владельца, соответствующим делегированному имени, и каждая из этих RRSIG RR является полномочными данными, связанными с той же самой зоной, которая содержит соответствующий набор NSEC RRset. При необходимости распознаватель может различать эти записи RRSIG RR по значению поля Signer's Name.

5.3.3. Проверка подписи

После проверки распознавателем пригодности RRSIG RR, как описано в параграфе 5.3.1, и восстановления исходных подписанных данных, как описано в параграфе 5.3.2, проверяющий может предпринять попытку использования криптографической подписи для аутентификации подписанных данных и, таким образом, (окончательной!) аутентификации RRset.

Поле Algorithm в записи RRSIG RR указывает криптографический алгоритм, использованный для создания подписи. Сама подпись содержится в поле Signature записи RRSIG RDATA, а открытый ключ, используемый для проверки подписи, - в поле Public Key соответствующей записи (записей) DNSKEY RR (найденных, как описано в 5.3.1). В [RFC4034] приведён список типов алгоритмов и ссылки на документы, в которых определено применение каждого из этих алгоритмов.

Отметим, что возможно наличие множества DNSKEY RR, соответствующих условиям параграфа 5.3.1. В этом случае проверяющий может найти подходящую запись DNSKEY RR только пробуя каждый соответствующий открытый ключ, пока проверка подписи не завершится успехом или не будут исчерпаны все ключи.

Если значение поля Labels в RRSIG RR не совпадает с числом меток в полном доменном имени владельца RRset, это говорит о некорректности набора RRset или преобразовании шаблонного имени. Распознаватель **должен** проверить корректность преобразования шаблона до того, как признать набор RRset подлинным. Процедура проверки корректности преобразования шаблонного имени описана в параграфе 5.3.4.

Если другие записи RRSIG RR также охватывают этот набор RRset, локальная политика безопасности распознавателя определяет, следует ли распознавателю проверять эти RRSIG RR, а также задаёт способ разрешения конфликтов в тех случаях, когда разные записи RRSIG RR дают различные результаты.

Если распознаватель признает набор RRset подлинным, он **должен** установить для TTL в RRSIG RR и каждой записи RR в аутентифицированном наборе RRset значение, не превышающее минимального из перечисленных ниже:

- значение TTL из RRset, указанное в полученном отклике;
- значение TTL из RRSIG RR, указанное в полученном отклике;
- значение Original TTL в RRSIG RR;
- разница между значением Signature Expiration в RRSIG RR и текущим временем.

5.3.4. Проверка RRset из позитивного отклика с преобразованием шаблона

Если число меток в имени владельца RRset превышает значение поля Labels в охватывающей этот набор записи RRSIG RR, это говорит о том, что RRset и покрывающая его запись RRSIG RR были созданы в результате преобразования шаблонного имени. После того, как валидатор проверит подпись, как описано в параграфе 5.3, он должен выполнить дополнительные действия для проверки отсутствия точного или более близкого соответствия шаблону для запроса, описанные в параграфе 5.4.

Отмети, что полученный распознавателем отклик должен включать все записи NSEC RR, требуемые для проверки его подлинности (см. параграф 3.1.3).

5.4. Аутентифицированный ответ об отсутствии

Распознаватель может использовать аутентифицированные записи NSEC RR для подтверждения отсутствия RRset в подписанной зоне. Защищённым серверам имён следует автоматически включать все требуемые записи NSEC RR для подписанных зон в свои отклики защищённым распознавателям.

Отсутствие определяется по приведённым ниже правилам.

- Если имя запрошенной RR совпадает с именем владельца аутентифицированной NSEC RR, поле битового отображения типов в NSEC RR будет указывать все типы RR, присутствующие для данного имени владельца и распознаватель может убедиться в отсутствии RR запрошенного типа, проверяя битовую маску типов. Если число меток в имени владельца аутентифицированной записи NSEC RR совпадает со значением поля Labels в покрывающей записи RRSIG RR, существование NSEC RR подтверждает, что преобразование шаблона не использовалось для этого запроса.
- Если имя запрашиваемой RR появляется после имени владельца аутентифицированной записи NSEC RR и до имени, указанного в поле Next Domain Name этой NSEC RR в соответствии с каноническим порядком имён DNS, определённом в [RFC4034], это говорит об отсутствии в зоне наборов RRset с запрошенным именем. Однако возможны ситуации с использованием шаблона для определения соответствия запрашиваемой RR имени владельца и типу, поэтому для подтверждения отсутствия запрошенного RRset требуется также подтвердить отсутствие каких-либо RRset, соответствующих шаблону.

В дополнение к этому защищённые распознаватели **должны** аутентифицировать наборы NSEC RRset, включающие подтверждение отсутствия, как описано в параграфе 5.3.

Для подтверждения отсутствия RRset распознаватель должен быть способен проверить как отсутствие запрошенного набора RRset, так и отсутствие подходящих шаблонных наборов RRset. Для этого может потребоваться более одного набора NSEC RRset из зоны. Если в отклике нет полного комплекта требуемых наборов NSEC RRset (возможно, в результате отсечки сообщения), защищённый распознаватель **должен** заново передать запрос для того, чтобы попытаться получить полный комплект наборов NSEC RR, требуемых для проверки отсутствия запрошенного RRset. Однако, как и во всех операциях DNS, распознаватель **должен** ограничивать действия, которые он выполняет при ответе на любой конкретный запрос.

Поскольку проверенная запись NSEC RR подтверждает существование самой себя и соответствующей записи RRSIG RR, проверяющая сторона **должна** игнорировать установки битов NSEC и RRSIG в записи NSEC RR.

5.5. Поведение распознавателя в тех случаях, когда подпись не проверяется

Если по той или иной причине ни одна из записей RRSIG не может быть подтверждена, отклик **следует** рассматривать, как неприемлемый (BAD). Если проверка выполняется для обработки рекурсивного запроса, сервер имён **должен** возвращать клиенту, инициировавшему запрос, результат RCODE 2. Однако он **должен** возвращать полный отклик тогда и только тогда, когда в исходном запросе установлен бит CD. См. также параграф 4.7 по части кэшированных откликов, которые не проверяются.

5.6. Пример аутентификации

В Приложении C приведён пример процесса аутентификации.

6. Согласование с IANA

[RFC4034] включает обзор связанных с агентством IANA вопросов, относящихся к DNSSEC. Этот документ добавляет два связанных с IANA вопроса.

[RFC2535] резервирует биты CD и AD в заголовках сообщений. Значение бита AD было переопределено в [RFC3655], а данный документ заново определяет смысл битов CD и AD. Новых битов в заголовках сообщений DNS данный документ не определяет.

[RFC2671] вводит EDNS, а [RFC3225] резервирует бит DNSSEC OK и определяет его использование. Данный документ повторно определяет использование бита без внесения каких-либо изменений.

7. Вопросы безопасности

Этот документ описывает, как защитные расширения DNS используют криптографию с открытыми ключами для подписания и аутентификации наборов записей о ресурсах DNS. Терминология и общее описание вопросов

безопасности, связанных с DNSSEC, приведены в [RFC4033], а в [RFC4034] описаны специфические для DNSSEC типы записей о ресурсах.

Активный атакующий, способный установить бит CD в запросном сообщении DNS или бит AD в отклике DNS, может воспользоваться этими битами для преодоления защиты, которую DNSSEC пытается организовать для незащищённых рекурсивных распознавателей. По этой причине для использования этих битов защищёнными рекурсивными распознавателями требуется организация защищённого канала. Этот вопрос подробно рассмотрен в параграфах 3.2.2 и 4.9.

Описанный в этом документе протокол пытается расширить преимущества DNSSEC на незащищённые оконечные распознаватели. Однако, в силу специфики восстановления после отказов при проверке для конкретных приложений, средства, предлагаемые DNSSEC для оконечных распознавателей, могут оказаться не подходящими. Операторам защищённых рекурсивных серверов имён следует обращать пристальное внимание на поведение пользующихся услугами сервера приложений при выборе локальной политики проверки - пренебрежение этим может приводить к тому, что рекурсивный сервер имён может стать источником атаки на службы для клиентов, которых он должен поддерживать.

8. Благодарности

Этот документ был создан на основе предложений и идей членов рабочей группы DNS Extensions, а также подписчиков списка рассылок этой группы. Авторы рады выразить свою признательность за комментарии и предложения, полученные в процессе пересмотра этой спецификации защитных расширений. Хотя явно перечислить всех, кто внёс свои предложения в течение десятилетия разработки DNSSEC просто не возможно, в [RFC4033] приведён список некоторых активных участников обсуждения этих документов.

9. Литература

9.1. Нормативные документы

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, August 1999.

[RFC2672] Crawford, M., "Non-Terminal DNS Name Redirection", RFC 2672, August 1999.

[RFC3225] Conrad, D., "Indicating Resolver Support of DNSSEC", RFC 3225, December 2001.

[RFC3226] Gudmundsson, O., "DNSSEC and IPv6 A6 aware server/resolver message size requirements", RFC 3226, December 2001.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for DNS Security Extensions", RFC 4034, March 2005.

9.2. Дополнительная литература

[RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, March 1998.

[RFC2535] Eastlake 3rd, D., "Domain Name System Security Extensions", RFC 2535, March 1999.

[RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.

[RFC3655] Wellington, B. and O. Gudmundsson, "Redefinition of DNS Authenticated Data (AD) bit", RFC 3655, November 2003.

Приложение A. Пример подписанной зоны

Ниже приведён пример (небольшой) полной подписанной зоны.

```
example.        3600 IN SOA ns1.example. bugs.x.w.example. (
                            1081539377
                            3600
                            300
                            3600000
                            3600
                            )
                3600 RRSIG  SOA 5 1 3600 20040509183619 (
                            20040409183619 38519 example.
                            ONx0k36rcjaxYtcNgq6iQnpNV5+drqYAsC9h
                            7TSJaHCqbhE67Sr6aH2xDUGcqQWu/n0UVzrF
                            vkgO9ebarZ0GWDKcuwlM6eNB5SiX2K74l5LW
                            DA7S/Un/IbtDq4Ay8NMNLQI7Dw7n4p8/rjkB
                            jV7j86HyQgM5e7+miRAz8V01b0I= )
                3600 NS     ns1.example.
                3600 NS     ns2.example.
```

```
                3600 RRSIG  NS 5 1 3600 20040509183619 (
                            20040409183619 38519 example.
                            gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
                            EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
                            4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
                            RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
                            0HjMeRaZB/FRPGfJPajngcq6Kwg= )
                3600 MX     1 xx.example.
                3600 RRSIG  MX 5 1 3600 20040509183619 (
                            20040409183619 38519 example.
                            HyDHYVT5KHSZ7HtO/vypumPmSZQrcOP3tzWB
                            2qaKkHVPfau/DgLgS/IKENkYOGL95G4N+NzE
                            VyNU8dcTOckT+ChPcGeVjguQ7a3Ao9Z/ZkUO
                            6gmmUW4b89rz1PUxW4jzUxj66PTwoVtUU/iM
                            W6OISukd1EQt7a0kygkg+PEDxdI= )
                3600 NSEC   a.example. NS SOA MX RRSIG NSEC DNSKEY
                3600 RRSIG  NSEC 5 1 3600 20040509183619 (
                            20040409183619 38519 example.
                            O0k558jHhyrC97ISHnislm4kLMW48C7U7cBm
                            FTfhke5iVqNRVTB1STLMpgpbDIC9hcryoO0V
                            Z9ME5xPzUEhbvGnHd5sfzgFVeGxr5Nyyq4tW
                            SDBgIBiLQUv1ivy29vhXy7WgR62dPrZ0PWvm
                            jfFJ5arXf4nPxp/kEowGgBRzY/U= )
                3600 DNSKEY 256 3 5 (
                            AQOy1bZVvpPqhg4j7EJoM9rI3ZmyEx2OzDBV
                            rZy/lvI5CQePxXHZS4i8dANH4DX3tbHol61e
                            k8EFMcsGXxKciJFHyhl94C+NwILQdzsUlSFo
                            vBZsyl/NX6yEbtw/xN9ZNcrbYvgjjZ/UVPZI
                            ySFNsgEYvh0z2542lzMKR4Dh8uZffQ==
                            )
                3600 DNSKEY 257 3 5 (
                            AQOeX7+baTmvpVHb2CcLnL1dMRWbuscRvHXl
                            LnXwDzvqp4tZVKp1sZMepFb8MvxhhW3y/0QZ
                            syCjczGJ1qk8vJe52iOhInKROVLRwxGpMfzP
                            RLMlGybr51bOV/1se0ODacj3DomyB4QB5gKT
                            Yot/K9alk5/j8vfd4jWCWD+E1Sze0Q==
                            )
                3600 RRSIG  DNSKEY 5 1 3600 20040509183619 (
                            20040409183619 9465 example.
                            ZxgauAuIj+k1YoVEOSlZfx41fcmKzTFHoweZ
                            xYnz99JVQZJ33wFS0Q0jcP7VXKkaElXk9nYJ
                            XevO/7nAbo88iWsMkSpSR6jWzYYKwfrBI/L9
                            hjYmyVO9m6FjQ7uwM4dCP/bIuV/DKqOAK9NY
                            NC3AHfvCV1Tp4VKDqxqG7R5tTVM= )
                3600 RRSIG  DNSKEY 5 1 3600 20040509183619 (
                            20040409183619 38519 example.
                            eGL0s90glUqcOmloo/2y+bSzyEfKVOQViD9Z
                            DNhLz/Yn9CQZlDVRJffACQDAUhXpU/oP34ri
                            bKBpysRXosczFrKqS5Oa0bzMOfXCXup9qHAp
                            eFIku28Vqfr8Nt7cigZLxjK+u0Ws/4lIRjKk
                            7z5OXogYVaFzHKillDt3HRxHIZM= )
a.example.      3600 IN NS  ns1.a.example.
                3600 IN NS  ns2.a.example.
                3600 DS     57855 5 1 (
                            B6DCD485719ADCA18E5F3D48A2331627FDD3
                            636B )
                3600 RRSIG  DS 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            oXIKit/QtdG64J/CB+Gi8dOvnwRvqrto1AdQ
                            oRkAN15FP3iZ7suB7gvTBmXzCjL7XUgQVcoH
                            kdhyCuzp8W9qJHgRUSwKKkczSyuL64nhgjuD
                            EML8l9wlWVsl7PR2VnZduM9bLyBhaaPmRKX/
                            Fm+v6ccF2EGNLRiY08kdkz+XHHo= )
                3600 NSEC   ai.example. NS DS RRSIG NSEC
                3600 RRSIG  NSEC 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            cOlYgqJLqlRqmBQ3iap2SyIsK4O5aqpKSoba
                            U9fQ5SMApZmHfq3AgLflkrkXRXvgxTQSKkG2
                            039/cRUs6Jk/25+fi7Xr5nOVJsb0lq4zsB3I
                            BBdjyGDAHE0F5ROJj87996vJupdm1fbH481g
                            sdkOW6Zyqtz3Zos8N0BBkEx+2G4= )
ns1.a.example.  3600 IN A   192.0.2.5
ns2.a.example.  3600 IN A   192.0.2.6
ai.example.     3600 IN A   192.0.2.9
                3600 RRSIG  A 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            pAOtzLP2MU0tDJUwHOKE5FPIIHmdYsCgTb5B
                            ERGgpnJluA9ixOyf6xxVCgrEJW0WNZSsJicd
                            hBHXfDmAGKUajUUlYSAH8tS4ZnrhyymIvk3u
                            ArDu2wfT130e9UHnumaHHMpUTosKe22PblOy
                            6zrTpg9FkS0XGVmYRvOTNYx2HvQ= )
                3600 HINFO  "KLH-10" "ITS"
                3600 RRSIG  HINFO 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            Iq/RGCbBdKzcYzlGE4ovbr5YcB+ezxbZ9W0l
                            e/7WqyvhOO9J16HxhhL7VY/IKmTUY0GGdcfh
```

```
                                ZEOCkf4lEykZF9NPok1/R/fWrtzNp8jobuY7
                                AZEcZadp1WdDF3jc2/ndCa5XZhLKD3JzOsBw
                                FvL8sqlS5QS6FY/ijFEDnI4RkZA= )
                    3600 AAAA   2001:db8::f00:baa9
                    3600 RRSIG  AAAA 5 2 3600 20040509183619 (
                                20040409183619 38519 example.
                                nLcpFuXdT35AcE+EoafOUkl69KB+/e56XmFK
                                kewXG2IadYLKAOBIoR5+VoQV3XgTcofTJNsh
                                1rnF6Eav2zpZB3byI6yo2bwY8MNkr4A7cL9T
                                cMmDwV/hWFKsbGBsj8xSCN/caEL2CWY/5XP2
                                sZM6QjBBLmukH30+w1z3h8PUP2o= )
                    3600 NSEC   b.example. A HINFO AAAA RRSIG NSEC
                    3600 RRSIG  NSEC 5 2 3600 20040509183619 (
                                20040409183619 38519 example.
                                QoshyPevLcJ/xcRpEtMft1uoIrcrieVcc9pG
                                CScIn5Glnib40T6ayVOimXwdSTZ/8ISXGj4p
                                P8Sh0PlA6olZQ84L453/BUqB8BpdOGky4hsN
                                3AGcLEv1Gr0QMvirQaFcjzOECfnGyBm+wpFL
                                AhS+JOVfDI/79QtyTI0SaDWcg8U= )
b.example.          3600 IN NS  ns1.b.example.
                    3600 IN NS  ns2.b.example.
                    3600 NSEC   ns1.example. NS RRSIG NSEC
                    3600 RRSIG  NSEC 5 2 3600 20040509183619 (
                                20040409183619 38519 example.
                                GNuxHn844wfmUhPzGWKJCPY5ttEX/RfjDoOx
                                9ueK1PtYkOWKOOdiJ/PJKCYB3hYX+858dDWS
                                xb2qnV/LSTCNVBnkm6owOpysY97MVj5VQEWs
                                0lm9tFoqjcptQkmQKYPrwUnCSNwvvclSF1xZ
                                vhRXgWT7OuFXldoCG6TfVFMs9xE= )
ns1.b.example.      3600 IN A   192.0.2.7
ns2.b.example.      3600 IN A   192.0.2.8
ns1.example.        3600 IN A   192.0.2.1
                    3600 RRSIG  A 5 2 3600 20040509183619 (
                                20040409183619 38519 example.
                                F1C9HVhIcs10cZU09G5yIVfKJy5yRQQ3qVet
                                5pGhp82pzhAOMZ3K22JnmK4c+IjUeFp/to06
                                im5FVpHtbFisdjyPq84bhTv8vrXt5AB1wNB+
                                +iAqvIfdgW4sFNC6oADb1hK8QNauw9VePJhK
                                v/iVXSYC0b7mPSU+EOlknFpVECs= )
                    3600 NSEC   ns2.example. A RRSIG NSEC
                    3600 RRSIG  NSEC 5 2 3600 20040509183619 (
                                20040409183619 38519 example.
                                I4hj+Kt6+8rCcHcUdolks2S+Wzri9h3fHas8
                                1rGN/eILdJHN7JpV6lLGPIh/8fIBkfvdyWnB
                                jjf1q3O7JgYO1UdI7FvBNWqaaEPJK3UkddBq
                                ZIaLi8Qr2XHkjq38BeQsbp8X0+6h4ETWSGT8
                                IZaIGBLryQWGLw6Y6X8dqhlnxJM= )
ns2.example.        3600 IN A   192.0.2.2
                    3600 RRSIG  A 5 2 3600 20040509183619 (
                                20040409183619 38519 example.
                                V7cQRw1TR+knlaL1z/psxlS1PcD37JJDaCMq
                                Qo6/u1qFQu6x+wuDHRH22Ap9ulJPQjFwMKOu
                                yfPGQPC8KzGdE3vt5snFEAoE1Vn3mQqtu7SO
                                6amIjk13Kj/jyJ4nGmdRIc/3cM3ipXFhNTKq
                                rdhx8SZ0yy4ObIRzIzvBFLiSS8o= )
                    3600 NSEC   *.w.example. A RRSIG NSEC
                    3600 RRSIG  NSEC 5 2 3600 20040509183619 (
                                20040409183619 38519 example.
                                N0QzHvaJf5NRw1rE9uxS1Ltb2LZ73Qb9bKGE
                                VyaISkqzGpP3jYJXZJPVTq4UVEsgT3CgeHvb
                                3QbeJ5Dfb2V9NGCHj/OvF/LBxFFWwhLwzngH
                                l+bQAgAcMsLu/nL3nDi1y/JSQjAcdZNDl4bw
                                Ymx28EtgIpo9A0qmP08rMBqs1Jw= )
*.w.example.        3600 IN MX  1 ai.example.
                    3600 RRSIG  MX 5 2 3600 20040509183619 (
                                20040409183619 38519 example.
                                OMK8rAZlepfzLWW75Dxd63jy2wswESzxDKG2
                                f9AMN1CytCd10cYISAxfAdvXSZ7xujKAtPbc
                                tvOQ2ofO7AZJ+d01EeeQTVBPq4/6KCWhqe2X
                                TjnkVLNvvhnc0u28aoSsG0+4InvkkOHknKxw
                                4kX18MMR34i8lC36SR5xBni8vHI= )
                    3600 NSEC   x.w.example. MX RRSIG NSEC
                    3600 RRSIG  NSEC 5 2 3600 20040509183619 (
                                20040409183619 38519 example.
                                r/mZnRC3I/VIcrelgIcteSxDhtsdlTDt8ng9
                                HSBlABOlzLxQtfgTnn8f+aOwJIAFe1Ee5RvU
                                5cVhQJNP5XpXMJHfyps8tVvfxSAXfahpYqtx
                                91gsmcV/1V9/bZAG55CefP9cM4Z9Y9NT9XQ8
                                s1InQ2UoIv6tJEaaKkP701j8OLA= )
x.w.example.        3600 IN MX  1 xx.example.
                    3600 RRSIG  MX 5 3 3600 20040509183619 (
                                20040409183619 38519 example.
                                Il2WTZ+Bkv+OytBx4LItNW5mjB4RCwhOO8y1
                                XzPHZmZUTVYL7LaA63f6T9ysVBzJRI3KRjAP
                                H3U1qaYnDoN1DrWqmi9RJe4FoObkbcdm7P3I
                                kx70ePCoFgRz1Yq+bVVXCvGuAU4xALv3W/Y1
```

```
                                jNSlwZ2mSWKHfxFQxPtLj8s32+k= )
                    3600 NSEC   x.y.w.example. MX RRSIG NSEC
                    3600 RRSIG  NSEC 5 3 3600 20040509183619 (
                                20040409183619 38519 example.
                                aRbpHftxggzgMXdDlym9SsADqMZovZZl2QWK
                                vw8J0tZEUNQByH5Qfnf5N1FqH/pS46UA7A4E
                                mcWBN9PUA1pdPY6RVeaRlZlCr1IkVctvbtaI
                                NJuBba/VHm+pebTbKcAPIvL9tBOoh+to1h6e
                                IjgiM8PXkBQtxPq37wDKALkyn7Q= )
x.y.w.example.  3600 IN MX  1 xx.example.
                    3600 RRSIG  MX 5 4 3600 20040509183619 (
                                20040409183619 38519 example.
                                k2bJHbwP5LH5qN4is39UiPzjAWYmJA38Hhia
                                t7i9t7nbX/e0FPnvDSQXzcK7UL+zrVA+3MDj
                                q1ub4q3SZgcbLMgexxIW3Va//LVrxkP6Xupq
                                GtOB9prkK54QTl/qZTXfMQpW480YOvVknhvb
                                +gLcMZBnHJ326nb/TOOmrqNmQQE= )
                    3600 NSEC   xx.example. MX RRSIG NSEC
                    3600 RRSIG  NSEC 5 4 3600 20040509183619 (
                                20040409183619 38519 example.
                                OvE6WUzN2ziieJcvKPWbCAyXyP6ef8cr6Csp
                                ArVSTzKSquNwbezZmkU7E34o5lmb6CWSSSpg
                                xw098kNUFnHcQf/LzY2zqRomubrNQhJTiDTX
                                a0ArunJQCzPjOYq5t0SLjm6qp6McJI1AP5Vr
                                QoKqJDCLnoAlcPOPKAm/jJkn3jk= )
xx.example.     3600 IN A   192.0.2.10
                    3600 RRSIG  A 5 2 3600 20040509183619 (
                                20040409183619 38519 example.
                                kBF4YxMGWF0D8r0cztL+2fWWOvN1U/GYSpYP
                                7SoKoNQ4fZKyk+weWGlKLIUM+uE1zjVTPXoa
                                0Z6WG0oZp46rkl1EzMcdMgoaeUzzAJ2BMq+Y
                                VdxG9IK1yZkYGY9AgbTOGPoAgbJyO9EPULsx
                                kbIDV6GPPSZVusnZU6OMgdgzHV4= )
                    3600 HINFO  "KLH-10" "TOPS-20"
                    3600 RRSIG  HINFO 5 2 3600 20040509183619 (
                                20040409183619 38519 example.
                                GY2PLSXmMHkWHfLdggiox8+chWpeMNJLkML0
                                t+U/SXSUsoUdR91KNdNUkTDWamwcF8oFRjhq
                                BcPZ6EqrF+vl5v5oGuvSF7U52epfVTC+wWF8
                                3yCUeUw8YklhLWlvk8gQ15YKth0ITQy8/wI+
                                RgNvuwbioFSEuv2pNlkq0goYxNY= )
                    3600 AAAA   2001:db8::f00:baaa
                    3600 RRSIG  AAAA 5 2 3600 20040509183619 (
                                20040409183619 38519 example.
                                Zzj0yodDxcBLnnOIwDsuKo5WqiaK24DlKg9C
                                aGaxDFiKgKobUj2jilYQHpGFn2poFRetZd4z
                                ulyQkssz2QHrVrPuTMS22knudCiwP4LWpVTr
                                U4zfeA+rDz9stmSBP/4PekH/x2IoAYnwctd/
                                xS9cL2QgW7FChw16mzlkH6/vsfs= )
                    3600 NSEC   example. A HINFO AAAA RRSIG NSEC
                    3600 RRSIG  NSEC 5 2 3600 20040509183619 (
                                20040409183619 38519 example.
                                ZFWUln6Avc8bmGl5GFjD3BwT530DUZKHNuoY
                                9A8lgXYyrxu+pqgFiRVbyZRQvVB5pccEOT3k
                                mvHgEa/HzbDB4PIYY79W+VHrgOxzdQGGCZzi
                                asXrpSGOWwSOElghPnMIi8xdF7qtCntr382W
                                GghLahumFIpg4MO3LS/prgzVVWo= )
```

Вершина набора DNSKEY включает две записи DNSKEY RR и поле DNSKEY RDATA Flags показывает, что каждая из этих DNSKEY RR является ключом зоны. Одна из этих записей DNSKEY RR имеет также установленный флаг SEP и служит подписью вершины DNSKEY RRset; этот ключ следует хэшировать для генерации записи DS, которая будет помещаться в родительскую зону. Другая запись DNSKEY используется в качестве подписи для всех остальных RRset в зоне.

Зона включает шаблонную запись *.w.example. Отметим, что имя *.w.example используется при создании цепочек NSEC и подпись RRSIG, покрывающая набор *.w.example MX RRset имеет значение счётчика меток 2.

Зона также включает два делегирования. Делегирование для b.example включает NS RRset, склеивающие записи и NSEC RR, причём подписывается только NSEC RRset. Делегирование для a.example обеспечивает DS RR и подписываются только NSEC и наборы DS RRset.

Приложение B. Примеры откликов

Приведённые ниже примеры показывают сообщения откликов для зоны из Приложения A.

B.1. Ответ

Успешный запрос к полномочному серверу.

```
;; Header: QR AA DO RCODE=0
;;
;; Question
x.w.example.        IN MX

;; Answer
x.w.example.    3600 IN MX  1 xx.example.
```

```
x.w.example.    3600 RRSIG  MX 5 3 3600 20040509183619 (
                            20040409183619 38519 example.
                            Il2WTZ+Bkv+OytBx4LItNW5mjB4RCwhOO8y1
                            XzPHZmZUTVYL7LaA63f6T9ysVBzJRI3KRjAP
                            H3U1qaYnDoN1DrWqmi9RJe4FoObkbcdm7P3I
                            kx70ePCoFgRz1Yq+bVVXCvGuAU4xALv3W/Y1
                            jNSlwZ2mSWKHfxFQxPtLj8s32+k= )

;; Authority
example.        3600 NS     ns1.example.
example.        3600 NS     ns2.example.
example.        3600 RRSIG  NS 5 1 3600 20040509183619 (
                            20040409183619 38519 example.
                            gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
                            EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
                            4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
                            RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
                            0HjMeRaZB/FRPGfJPajngcq6Kwg= )

;; Additional
xx.example.     3600 IN A   192.0.2.10
xx.example.     3600 RRSIG  A 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            kBF4YxMGWF0D8r0cztL+2fWWOvN1U/GYSpYP
                            7SoKoNQ4fZKyk+weWGlKLIUM+uE1zjVTPXoa
                            0Z6WG0oZp46rkl1EzMcdMgoaeUzzAJ2BMq+Y
                            VdxG9IK1yZkYGY9AgbTOGPoAgbJyO9EPULsx
                            kbIDV6GPPSZVusnZU6OMgdgzHV4= )
xx.example.     3600 AAAA   2001:db8::f00:baaa
xx.example.     3600 RRSIG  AAAA 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            Zzj0yodDxcBLnnOIwDsuKo5WqiaK24DlKg9C
                            aGaxDFiKgKobUj2jilYQHpGFn2poFRetZd4z
                            ulyQkssz2QHrVrPuTMS22knudCiwP4LWpVTr
                            U4zfeA+rDz9stmSBP/4PekH/x2IoAYnwctd/
                            xS9cL2QgW7FChw16mzlkH6/vsfs= )
ns1.example.    3600 IN A   192.0.2.1
ns1.example.    3600 RRSIG  A 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            F1C9HVhIcs10cZU09G5yIVfKJy5yRQQ3qVet
                            5pGhp82pzhAOMZ3K22JnmK4c+IjUeFp/to06
                            im5FVpHtbFisdjyPq84bhTv8vrXt5AB1wNB+
                            +iAqvIfdgW4sFNC6oADb1hK8QNauw9VePJhK
                            v/iVXSYC0b7mPSU+EOlknFpVECs= )
ns2.example.    3600 IN A   192.0.2.2
ns2.example.    3600 RRSIG  A 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            V7cQRw1TR+knlaL1z/psxlS1PcD37JJDaCMq
                            Qo6/u1qFQu6x+wuDHRH22Ap9ulJPQjFwMKOu
                            yfPGQPC8KzGdE3vt5snFEAoE1Vn3mQqtu7SO
                            6amIjk13Kj/jyJ4nGmdRIc/3cM3ipXFhNTKq
                            rdhx8SZ0yy4ObIRzIzvBFLiSS8o= )
```

B.2. Ошибка имени

Ответ полномочного сервера об ошибке в имени. Записи NSEC RR говорят об отсутствии имени и покрывающего его шаблонного имени.

```
;; Header: QR AA DO RCODE=3
;;
;; Question
ml.example.         IN A

;; Answer
;; (пусто)

;; Authority
example.        3600 IN SOA ns1.example. bugs.x.w.example. (
                            1081539377
                            3600
                            300
                            3600000
                            3600
                            )
example.        3600 RRSIG  SOA 5 1 3600 20040509183619 (
                            20040409183619 38519 example.
                            ONx0k36rcjaxYtcNgq6iQnpNV5+drqYAsC9h
                            7TSJaHCqbhE67Sr6aH2xDUGcqQWu/n0UVzrF
                            vkgO9ebarZ0GWDKcuwlM6eNB5SiX2K74l5LW
                            DA7S/Un/IbtDq4Ay8NMNLQI7Dw7n4p8/rjkB
                            jV7j86HyQgM5e7+miRAz8V01b0I= )
b.example.      3600 NSEC   ns1.example. NS RRSIG NSEC
b.example.      3600 RRSIG  NSEC 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            GNuxHn844wfmUhPzGWKJCPY5ttEX/RfjDoOx
                            9ueK1PtYkOWKOOdiJ/PJKCYB3hYX+858dDWS
                            xb2qnV/LSTCNVBnkm6owOpysY97MVj5VQEWs
```

```
                            0lm9tFoqjcptQkmQKYPrwUnCSNwvvclSF1xZ
                            vhRXgWT7OuFXldoCG6TfVFMs9xE= )
example.        3600 NSEC   a.example. NS SOA MX RRSIG NSEC DNSKEY
example.        3600 RRSIG  NSEC 5 1 3600 20040509183619 (
                            20040409183619 38519 example.
                            O0k558jHhyrC97ISHnislm4kLMW48C7U7cBm
                            FTfhke5iVqNRVTB1STLMpgpbDIC9hcryoO0V
                            Z9ME5xPzUEhbvGnHd5sfzgFVeGxr5Nyyq4tW
                            SDBgIBiLQUv1ivy29vhXy7WgR62dPrZ0PWvm
                            jfFJ5arXf4nPxp/kEowGgBRzY/U= )

;; Additional
;; (пусто)
```

B.3. Нет данных

Отклик об отсутствии данных. Запись Запись NSEC RR говорит, что имя существует, а запись RR запрошенного типа не существует.

```
;; Header: QR AA DO RCODE=0
;;
;; Question
ns1.example.        IN MX

;; Answer
;; (пусто)

;; Authority
example.        3600 IN SOA ns1.example. bugs.x.w.example. (
                            1081539377
                            3600
                            300
                            3600000
                            3600
                            )
example.        3600 RRSIG  SOA 5 1 3600 20040509183619 (
                            20040409183619 38519 example.
                            ONx0k36rcjaxYtcNgq6iQnpNV5+drqYAsC9h
                            7TSJaHCqbhE67Sr6aH2xDUGcqQWu/n0UVzrF
                            vkgO9ebarZ0GWDKcuwlM6eNB5SiX2K7415LW
                            DA7S/Un/IbtDq4Ay8NMNLQI7Dw7n4p8/rjkB
                            jV7j86HyQgM5e7+miRAz8V01b0I= )
ns1.example.    3600 NSEC   ns2.example. A RRSIG NSEC
ns1.example.    3600 RRSIG  NSEC 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            I4hj+Kt6+8rCcHcUdolks2S+Wzri9h3fHas8
                            1rGN/eILdJHN7JpV6lLGPIh/8fIBkfvdyWnB
                            jjf1q3O7JgYO1UdI7FvBNWqaaEPJK3UkddBq
                            ZIaLi8Qr2XHkjq38BeQsbp8X0+6h4ETWSGT8
                            IZaIGBLryQWGLw6Y6X8dqhlnxJM= )

;; Additional
;; (пусто)
```

B.4. Отсылка к подписанной зоне

Запись DS RR содержит данные, которые будут нужны распознавателю для проверки соответствующей DNSKEY RR на вершине дочерней зоны.

```
;; Header: QR DO RCODE=0
;;
;; Question
mc.a.example.       IN MX

;; Answer
;; (пусто)

;; Authority
a.example.      3600 IN NS  ns1.a.example.
a.example.      3600 IN NS  ns2.a.example.
a.example.      3600 DS     57855 5 1 (
                            B6DCD485719ADCA18E5F3D48A2331627FDD3
                            636B )
a.example.      3600 RRSIG  DS 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            oXIKit/QtdG64J/CB+Gi8dOvnwRvqrto1AdQ
                            oRkAN15FP3iZ7suB7gvTBmXzCjL7XUgQVcoH
                            kdhyCuzp8W9qJHgRUSwKKkczSyuL64nhgjuD
                            EML8l9wlWVsl7PR2VnZduM9bLyBhaaPmRKX/
                            Fm+v6ccF2EGNLRiY08kdkz+XHHo= )

;; Additional
ns1.a.example. 3600 IN A    192.0.2.5
ns2.a.example. 3600 IN A    192.0.2.6
```

B.5. Отсылка к неподписанной зоне

Запись NSEC RR говорит об отсутствии в родительской зоне записи DS RR для этого делегирования.

```
;; Header: QR DO RCODE=0
;;
;; Question
mc.b.example.       IN MX

;; Answer
;; (пусто)

;; Authority
b.example.      3600 IN NS  ns1.b.example.
b.example.      3600 IN NS  ns2.b.example.
b.example.      3600 NSEC   ns1.example. NS RRSIG NSEC
b.example.      3600 RRSIG  NSEC 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            GNuxHn844wfmUhPzGWKJCPY5ttEX/RfjDoOx
                            9ueK1PtYkOWKOOdiJ/PJKCYB3hYX+858dDWS
                            xb2qnV/LSTCNVBnkm6owOpysY97MVj5VQEWs
                            0lm9tFoqjcptQkmQKYPrwUnCSNwvvclSF1xZ
                            vhRXgWT7OuFXldoCG6TfVFMs9xE= )
;; Additional
ns1.b.example. 3600 IN A    192.0.2.7
ns2.b.example. 3600 IN A    192.0.2.8
```

B.6. Преобразование шаблона

Успешный запрос, для ответа на который выполнялось преобразование шаблонного имени. Счётчик меток в записи RRSIG RR отклика показывает, что шаблонный набор RRset был преобразован для создания этого отклика с заменой шаблона реальной меткой, NSEC RR подтверждает, что более точного соответствия шаблону в зоне не существует.

```
;; Header: QR AA DO RCODE=0
;;
;; Question
a.z.w.example.      IN MX

;; Answer
a.z.w.example. 3600 IN MX  1 ai.example.
a.z.w.example. 3600 RRSIG  MX 5 2 3600 20040509183619 (
                           20040409183619 38519 example.
                           OMK8rAZlepfzLWW75Dxd63jy2wswESzxDKG2
                           f9AMN1CytCd10cYISAxfAdvXSZ7xujKAtPbc
                           tvOQ2ofO7AZJ+d01EeeQTVBPq4/6KCWhqe2X
                           TjnkVLNvvhnc0u28aoSsG0+4InvkkOHknKxw
                           4kX18MMR34i8lC36SR5xBni8vHI= )

;; Authority
example.       3600 NS     ns1.example.
example.       3600 NS     ns2.example.
example.       3600 RRSIG  NS 5 1 3600 20040509183619 (
                           20040409183619 38519 example.
                           gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
                           EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
                           4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
                           RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
                           0HjMeRaZB/FRPGfJPajngcq6Kwg= )
x.y.w.example. 3600 NSEC   xx.example. MX RRSIG NSEC
x.y.w.example. 3600 RRSIG  NSEC 5 4 3600 20040509183619 (
                           20040409183619 38519 example.
                           OvE6WUzN2ziieJcvKPWbCAyXyP6ef8cr6Csp
                           ArVSTzKSquNwbezZmkU7E34o5lmb6CWSSSpg
                           xw098kNUFnHcQf/LzY2zqRomubrNQhJTiDTX
                           a0ArunJQCzPjOYq5t0SLjm6qp6McJI1AP5Vr
                           QoKqJDCLnoAlcPOPKAm/jJkn3jk= )

;; Additional
ai.example.    3600 IN A    192.0.2.9
ai.example.    3600 RRSIG   A 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            pAOtzLP2MU0tDJUwHOKE5FPIIHmdYsCgTb5B
                            ERGgpnJluA9ixOyf6xxVCgrEJW0WNZSsJicd
                            hBHXfDmAGKUajUUlYSAH8tS4ZnrhyymIvk3u
                            ArDu2wfT130e9UHnumaHHMpUTosKe22PblOy
                            6zrTpg9FkS0XGVmYRvOTNYx2HvQ= )
ai.example.    3600 AAAA    2001:db8::f00:baa9
ai.example.    3600 RRSIG   AAAA 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            nLcpFuXdT35AcE+EoafOUkl69KB+/e56XmFK
                            kewXG2IadYLKAOBIoR5+VoQV3XgTcofTJNsh
                            1rnF6Eav2zpZB3byI6yo2bwY8MNkr4A7cL9T
                            cMmDwV/hWFKsbGBsj8xSCN/caEL2CWY/5XP2
                            sZM6QjBBLmukH30+w1z3h8PUP2o= )
```

B.7. Отсутствие заданных шаблоном данных

Отклик no data (нет данных) для имени, покрываемого шаблоном. Записи NSEC RR подтверждают, что для соответствующего шаблону имени нет каких-либо записей RR запрошенного типа и в зоне нет более точного соответствия шаблону имени.

```
;; Header: QR AA DO RCODE=0
;;
;; Question
a.z.w.example.      IN AAAA

;; Answer
;; (пусто)

;; Authority
example.        3600 IN SOA ns1.example. bugs.x.w.example. (
                            1081539377
                            3600
                            300
                            3600000
                            3600
                            )
example.        3600 RRSIG  SOA 5 1 3600 20040509183619 (
                            20040409183619 38519 example.
                            ONx0k36rcjaxYtcNgq6iQnpNV5+drqYAsC9h
                            7TSJaHCqbhE67Sr6aH2xDUGcqQWu/n0UVzrF
                            vkgO9ebarZ0GWDKcuwlM6eNB5SiX2K74l5LW
                            DA7S/Un/IbtDq4Ay8NMNLQI7Dw7n4p8/rjkB
                            jV7j86HyQgM5e7+miRAz8V01b0I= )
x.y.w.example.  3600 NSEC   xx.example. MX RRSIG NSEC
x.y.w.example.  3600 RRSIG  NSEC 5 4 3600 20040509183619 (
                            20040409183619 38519 example.
                            OvE6WUzN2ziieJcvKPWbCAyXyP6ef8cr6Csp
                            ArVSTzKSquNwbezZmkU7E34o5lmb6CWSSSpg
                            xw098kNUFnHcQf/LzY2zqRomubrNQhJTiDTX
                            a0ArunJQCzPjOYq5t0SLjm6qp6McJI1AP5Vr
                            QoKqJDCLnoAlcPOPKAm/jJkn3jk= )
*.w.example.    3600 NSEC   x.w.example. MX RRSIG NSEC
*.w.example.    3600 RRSIG  NSEC 5 2 3600 20040509183619 (
                            20040409183619 38519 example.
                            r/mZnRC3I/VIcrelgIcteSxDhtsdlTDt8ng9
                            HSBlABOlzLxQtfgTnn8f+aOwJIAFe1Ee5RvU
                            5cVhQJNP5XpXMJHfyps8tVvfxSAXfahpYqtx
                            91gsmcV/1V9/bZAG55CefP9cM4Z9Y9NT9XQ8
                            s1InQ2UoIv6tJEaaKkP701j8OLA= )

;; Additional
;; (пусто)
```

B.8. Отсутствие данных DS для дочерней зоны

Отклик об отсутствии данных (no data) для запроса QTYPE=DS, ошибочно направленного серверу имён для дочерней зоны.

```
;; Header: QR AA DO RCODE=0
;;
;; Question
example.            IN DS

;; Answer
;; (пусто)

;; Authority
example.        3600 IN SOA ns1.example. bugs.x.w.example. (
                            1081539377
                            3600
                            300
                            3600000
                            3600
                            )
example.        3600 RRSIG  SOA 5 1 3600 20040509183619 (
                            20040409183619 38519 example.
                            ONx0k36rcjaxYtcNgq6iQnpNV5+drqYAsC9h
                            7TSJaHCqbhE67Sr6aH2xDUGcqQWu/n0UVzrF
                            vkgO9ebarZ0GWDKcuwlM6eNB5SiX2K74l5LW
                            DA7S/Un/IbtDq4Ay8NMNLQI7Dw7n4p8/rjkB
                            jV7j86HyQgM5e7+miRAz8V01b0I= )
example.        3600 NSEC   a.example. NS SOA MX RRSIG NSEC DNSKEY
example.        3600 RRSIG  NSEC 5 1 3600 20040509183619 (
                            20040409183619 38519 example.
                            O0k558jHhyrC97ISHnislm4kLMW48C7U7cBm
                            FTfhke5iVqNRVTB1STLMpgpbDIC9hcryoO0V
                            Z9ME5xPzUEhbvGnHd5sfzgFVeGxr5Nyyq4tW
                            SDBgIBiLQUv1ivy29vhXy7WgR62dPrZ0PWvm
                            jfFJ5arXf4nPxp/kEowGgBRzY/U= )

;; Additional
;; (пусто)
```

Приложение C. Примеры аутентификации

Примеры в этом приложении показывают, как выполняется аутентификация откликов, приведённых в Приложении B.

C.1. Аутентификация ответа

Запрос из Приложения B.1 возвращает набор MX RRset для x.w.example.com. Соответствующая запись RRSIG показывает, что набор MX RRset был подписан ключом DNSKEY для example с использованием алгоритма 5 и тега ключа 38519. Распознавателю нужна соответствующая запись DNSKEY RR для проверки подлинности этого ответа. Ниже описано, как распознаватель может получить эту запись DNSKEY RR.

Запись RRSIG показывает, что исходное значение TTL для MX RRset было 3600, и для целей аутентификации текущее значение TTL меняется на 3600. Значение 3 в поле Labels записи RRSIG говорит о том, что ответ не является результатом преобразования шаблона. Набор x.w.example.com MX RRset помещается в канонической форме и, в предположении того, что текущее время попадает в интервал между вводом и завершением срока действия подписи, эта подпись считается подлинной.

C.1.1. Пример аутентификации DNSKEY RR

Этот пример показывает логический процесс аутентификации, который начинается с заданного в конфигурации корня DNSKEY (или DS RR) и перемещается вниз по дереву для аутентификации нужной записи example DNSKEY RR. Отметим, что логический порядок представлен для лучшего понимания. Реализации могут выбрать проведение аутентификации по мере получения отсылок (referral) или путём создания аутентификационной цепочки только после получения всех наборов RRset, а также с использованием любой другой комбинации. Приведённый пример демонстрирует лишь логику, не задавая правил реализации.

Предполагается, что распознаватель начинает с заданной в конфигурации записи DNSKEY RR для корневой зоны (или с записи DS RR для корневой зоны). Распознаватель проверяет наличие заданной в конфигурации записи DNSKEY RR в корневом наборе DNSKEY RRset (или проверяет соответствие DS RR тому или иному ключу DNSKEY в корневом наборе DNSKEY RRset), факт подписания DNSKEY RR корневым набором DNSKEY RRset, а также корректность срока действия подписи. При выполнении всех перечисленных условий все ключи в DNSKEY RRset считаются подлинными. После этого распознаватель использует одну или несколько корневых записей DNSKEY RR для проверки подлинности example DS RRset. Отметим, что распознаватель может запросить у корневой зоны набор DNSKEY RRset или example DS RRset.

После подтверждения подлинности DS RRset с использованием корневой записи DNSKEY распознаватель проверяет example DNSKEY RRset на предмет наличия записи example DNSKEY RR, соответствующей аутентифицированным записям example DS RR. Если такая запись example DNSKEY найдена, распознаватель проверяет, что данная запись DNSKEY RR подписана example DNSKEY RRset и срок действия подписи не истек. При выполнении всех этих условий все ключи в наборе example DNSKEY RRset считаются аутентифицированными.

В заключение распознаватель проверяет, что та или иная запись DNSKEY RR в наборе example DNSKEY RRset использует алгоритм 5 и имеет тег ключа 38519. Эта запись DNSKEY служит для проверки подлинности записи RRSIG, включённой в отклик. Если номеру алгоритма и тегу ключа соответствует множество записей example DNSKEY RR, проверяется каждая запись DNSKEY RR и ответ считается подлинным, если любая из соответствующих записей DNSKEY RR подтверждает подпись, как было описано выше.

C.2. Ошибка имени

Запрос, приведённый в Приложении B.2, возвращает записи NSEC RR, подтверждающие отсутствие запрошенных данных и подходящих шаблонов. Негативные отклики аутентифицируются путём проверки обеих записей NSEC RR. Эти записи аутентифицируются, подобно описанному выше набору MX RRset.

C.3. Нет данных

Запрос, приведённый в Приложении B.3, возвращает запись NSEC RR, подтверждающую наличие запрошенного имени и отсутствие запрошенного типа RR. Негативный отклик аутентифицируется путём проверки записи NSEC RR. Эта запись аутентифицируется, подобно описанному выше набору MX RRset.

C.4. Отсылка к подписанной зоне

Запрос, приведённый в Приложении B.4, возвращает отсылку к подписанной зоне a.example. Запись DS RR аутентифицируется, подобно описанному выше набору MX RRset. Эта DS RR служит для проверки подлинности набора a.example DNSKEY RRset.

После того, как набор a.example DS RRset будет аутентифицирован с использованием записи example DNSKEY, распознаватель проверяет набор a.example DNSKEY RRset на предмет наличия записи a.example DNSKEY RR, соответствующей DS RR. При обнаружении такой записи a.example DNSKEY распознаватель проверяет, подписана ли эта DNSKEY RR с помощью a.example DNSKEY RRset и приемлема ли по сроку действия. Если все эти условия выполнены, набор a.example DNSKEY RRset считается подлинным.

C.5. Отсылка к неподписанной зоне

Запрос, приведённый в Приложении B.5, возвращает отсылку к неподписанной зоне b.example.. Запись NSEC подтверждает отсутствие аутентификации со стороны example для зоны b.example и запись NSEC RR аутентифицируется аналогично описанному выше набору MX RRset.

C.6. Преобразование шаблона

Запрос, приведённый в Приложении B.6, возвращает отклик, созданный в результате преобразования шаблона. Раздел answer содержит шаблонный набор RRset, полученный как в традиционном отклике DNS, и соответствующая запись RRSIG показывает, что шаблонный набор MX RRset был подписан example DNSKEY с использованием алгоритма 5 и тега ключа 38519. запись RRSIG показывает, что исходное значение TTL для набора MX RRset было 3600 и для целей аутентификации текущее значение TTL заменено на 3600. Поле Labels записи RRSIG со значением 2 показывает, что отклик является результатом преобразования шаблона, поскольку имя a.z.w.example включает 4

метки. Имя a.z.w.w.example заменяется на *.w.example, набор MX RRset помещается в канонической форме и, в предположении того, что текущее время попадает в срок действия подписи, последняя считается подлинной.

Запись NSEC подтверждает отсутствие более точного соответствия (включая полное совпадение) для отклика на этот запрос, а запись NSEC RR должна быть аутентифицирована до того, как ответ будет признан подлинным.

C.7. Отсутствие данных для шаблона

Запрос, приведённый в Приложении B.7, возвращает записи NSEC RR, которые подтверждают отсутствие запрошенных данных и подходящего шаблона. Негативный отклик аутентифицируется путём проверки обеих записей NSEC RR.

C.8. Отсутствие данных DS для дочерней зоны

Запрос, приведённый в Приложении B.8, возвращает записи NSEC RR, которые показывают, что ответ на запрос был получен от дочернего сервера (сервер example). Запись NSEC RR показывает наличие записи SOA RR, говорящей о получении ответа от потомка. Запросы для example DS RRset следует направлять родительским (корневым) серверам.

Адреса авторов

Roy Arends
Telematica Instituut
Brouwerijstraat 1
7523 XC Enschede
NL
EMail: roy.arends@telin.nl

Rob Austein
Internet Systems Consortium
950 Charter Street
Redwood City, CA 94063
USA
EMail: sra@isc.org

Matt Larson
VeriSign, Inc.
21345 Ridgetop Circle
Dulles, VA 20166-6503
USA
EMail: mlarson@verisign.com

Dan Massey
Colorado State University
Department of Computer Science
Fort Collins, CO 80523-1873
EMail: massey@cs.colostate.edu

Scott Rose
National Institute for Standards and Technology
100 Bureau Drive
Gaithersburg, MD 20899-8920
USA
EMail: scott.rose@nist.gov

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

Полное заявление авторских прав

Copyright (C) The Internet Society (2005).

К этому документу применимы права, лицензии и ограничения, указанные в BCP 78, и, за исключением указанного там, авторы сохраняют свои права.

Этот документ и содержащаяся в нем информация представлены "**как есть**" и **автор, организация, которую он/она представляет или которая выступает спонсором (если таковой имеется), Internet Society и IETF отказываются от каких-либо гарантий (явных или подразумеваемых), включая (но не ограничиваясь) любые гарантии того, что использование представленной здесь информации не будет нарушать чьих-либо прав, и любые предполагаемые гарантии коммерческого использования или применимости для тех или иных задач**.

Интеллектуальная собственность

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

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

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

Подтверждение

Финансирование функций RFC Editor в настоящее время обеспечивается Internet Society.