Network Working Group
Request for Comments: 4034
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

Записи RR для защитных расширений DNS

Resource Records 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.

Аннотация

Этот документ является одним из группы документов, описывающих расширения DNSSEC[2]. Защитные расширения DNS представляют собой множество записей о ресурсах и изменений в протоколах для обеспечения аутентификации источника данных в системе DNS. В данном документе определены записи для открытых ключей (DNSKEY), подписавшего передачу полномочий (DS), цифровой подписи (RRSIG) и аутентифицированного указания отсутствия (NSEC). Назначение и формат каждой записи подробно описаны в документе с примерами использования записей.

Этот документ отменяет действие RFC 2535 и включает в себя все обновления, предложенные для RFC 2535.

Оглавление

[1]В оригинале ошибочно указано также обновление RFC 3007. См. https://www.rfc-editor.org/errata_search.php?eid=3045. *Прим. перев.*
[2]DNS Security Extensions - защитные расширения DNS.

1. Введение

В защитных расширениях DNSSEC добавлены четыре новых типа записей о ресурсах DNS - открытый ключ DNS (DNSKEY[1]), подпись записи о ресурсе (RRSIG[2]), Next Secure (NSEC) и подписавший передачу полномочий (DS[3]). Этот документ определяет назначение каждой RR[4], формат RDATA и формат представления записи (ASCII-представление).

1.1. Связанные документы

Этот документ является частью семейства документов, определяющих DNSSEC, которые образуют единый комплекс.

В [RFC4033] дано введение в DNSSEC и определения общих терминов. Предполагается, что читатель уже ознакомился с этим документом. [RFC4033] также содержит список документов, отменяемых или изменяемых данным комплектом документов.

В [RFC4035] определены протокольные операции DNSSEC.

Предполагается знакомство читателя с базовыми концепциями DNS, описанными в [RFC1034], [RFC1035], а также последующих документах с обновлениями (в частности, [RFC2181] и [RFC2308]).

В этот документе определены записи DNSSEC о ресурсах. Все числовые коды типов DNS в этом документе указаны в десятичном представлении.

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. Запись DNSKEY RR

DNSSEC использует криптографию с открытыми ключами для подписания и аутентификации наборов записей о ресурсах DNS (RRset). Открытые ключи хранятся в записях DNSKEY и применяются в процессе аутентификации DNSSEC, описанном в [RFC4035] - зона подписывает свои полномочные RRset, используя секретный ключ и сохраняя соответствующий открытый ключ в DNSKEY RR. Преобразователь может использовать открытый ключ для проверки подписей, покрывающий наборы RRset в зоне, и, таким образом, аутентифицировать их.

DNSKEY RR не предназначены для хранения произвольных открытых ключей и их **недопустимо** использовать для хранения сертификатов или открытых ключей, не относящихся напрямую к инфраструктуре DNS.

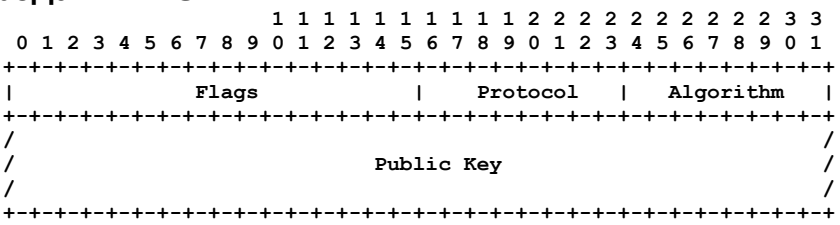
Поле Type для записей DNSKEY RR имеет значение 48.

Записи DNSKEY RR не зависят от класса.

К DNSKEY RR не предъявляется специальных требований в части времени жизни (TTL).

[1]DNS Public Key.
[2]Resource Record Signature.
[3]Delegation Signer.
[4]Resource record.

2.1. Формат передачи DNSKEY RDATA



RDATA для записей DNSKEY RR включает 2-октетное поле Flags, 1-октетные поля Protocol и Algorithm, а также поле Public Key.

2.1.1. Поле Flags

Бит 7 поля Flags является флагом ключа зоны (Zone Key). Если этот бит установлен (1), запись DNSKEY включает ключ зоны DNS и имя владельца DNSKEY RR **должно** быть именем зоны. Если бит 7 имеет значение 0, запись DNSKEY содержит какой-либо другой открытый ключ DNS и её **недопустимо** использовать для проверки подписей RRSIG, которые покрывают наборы RRset.

Бит 15 поля Flags является флагом защищённой точки входа (Secure Entry Point), описанным в [RFC3757]. Если этот бит имеет значение 1, запись DNSKEY содержит ключ, предназначенный для использования в качестве защищённой точки входа. Этот флаг служит лишь подсказкой для программ отладки и подписывания зон о возможности использования данной записи DNSKEY, для валидаторов **недопустимо** менять своё поведение в процессе проверки подписи на основе значения этого флага. Это также означает, что DNSKEY RR с установленным битом SEP будет требовать установленного флага Zone Key для обеспечения возможности легального создания подписей. Записи DNSKEY RR с установленным битом SEP и сброшенным флагом Zone Key **недопустимо** использовать для проверки подписей RRSIG, покрывающих RRset.

Биты 0 - 6 и 8 - 14 являются резервными, они **должны** сбрасываться при создании DNSKEY RR и игнорироваться при получении.

2.1.2. Поле Protocol

Поле Protocol **должно** иметь значение 3, а DNSKEY RR **должна** трактоваться, как непригодная в процессе проверки подписи, если это поле имеет другое значение.

2.1.3. Поле Algorithm

Поле Algorithm указывает криптографический алгоритм с открытым ключом и определяет формат поля Public Key. Список типов алгоритма DNSSEC приведён в Приложении A.1

2.1.4. Поле Public Key

Поле Public Key используется для ключевого материала. Формат поля зависит от алгоритма хранения ключей и описывается в отдельных документах.

2.1.5. Замечания по устройству DNSKEY RDATA

Хотя поле Protocol всегда имеет значение 3, оно сохраняется для обеспечения совместимости с ранними версиями записи KEY.

2.2. Формат представления DNSKEY RR

Часть RDATA представляется следующим образом:

поле Flags **должно** представляться, как десятичное целое число без знака; с учётом определённых в настоящее время флагов возможны значения поля 0, 256 и 257;

поле Protocol **должно** представляться десятичным целым числом без знака, имеющим значение 3;

поле Algorithm **должно** представляться десятичным целым числом без знака или с использованием мнемоники, описанной в Приложении A.1;

поле Public Key **должно** содержать представление ключа Public Key в формате Base64; в тексте Base64 могут использоваться пробелы, определение Base64 дано в работе [RFC3548].

2.3. Пример DNSKEY RR

Приведённая ниже запись DNSKEY RR хранит ключ зоны DNS для домена example.com.

```
example.com. 86400 IN DNSKEY 256 3 5 ( AQPSKmynfzW4kyBv015MUG2DeIQ3
                                       Cbl+BBZH4b/0PY1kxkmvHjcZc8no
                                       kfzj31GajIQKY+5CptLr3buXA10h
                                       WqTkF7H6RfoRqXQeogmMHfpftf6z
                                       Mv1LyBUgia7za6ZEzOJBOztyvhjL
                                       742iU/TpPSEDhm2SNKLijfUppn1U
                                       aNvv4w==  )
```

Первые четыре тестовых поля указывают имя владельца, TTL, класс и тип RR (DNSKEY). Значение 256 показывает, что флаг Zone Key (бит 7) в поле Flags имеет значение 1. Значение 3 в поле Protocol указывается всегда. Значение 5 указывает алгоритм с открытым ключом. В Приложении A.1 указано, что тип 5 используется для алгоритма RSA/SHA1, следовательно это значение говорит об использовании открытых ключей RSA/SHA1 в соответствии с [RFC3110]. Оставшаяся часть записи содержит представление Base64 для открытого ключа.

3. Запись RRSIG RR

DNSSEC использует криптографию с открытым ключом для подписания и аутентификации наборов записей о ресурсах DNS (RRset). Цифровые подписи хранятся в записях RRSIG и используются в процессе аутентификации DNSSEC, описанном в [RFC4035]. Проверяющий может использовать эти записи RRSIG для аутентификации RRset зоны. Записи RRSIG RR **должны** применяться только для передачи верификационного материала (цифровых подписей), используемого при защищённых операциях DNS.

Запись RRSIG содержит подпись для RRset с конкретным именем, классом и типом. RRSIG RR указывает интервал достоверности для подписи и использует поля Algorithm, Signer's Name и Key Tag для идентификации DNSKEY RR, содержащей открытый ключ, который проверяющий может использовать для проверки подписи.

Поскольку все полномочные RRset в зоне должны защищаться цифровой подписью, записи RRSIG RR должны присутствовать для имён, содержащих CNAME RR. Это меняет стандартную спецификацию DNS [RFC1034], в которой указано, что при наличии CNAME для имени, это будет единственный разрешенный для данного имени тип. В подписанной зоне для того же имени, что и CNAME, **должны** существовать записи RRSIG и NSEC (см. раздел 4).

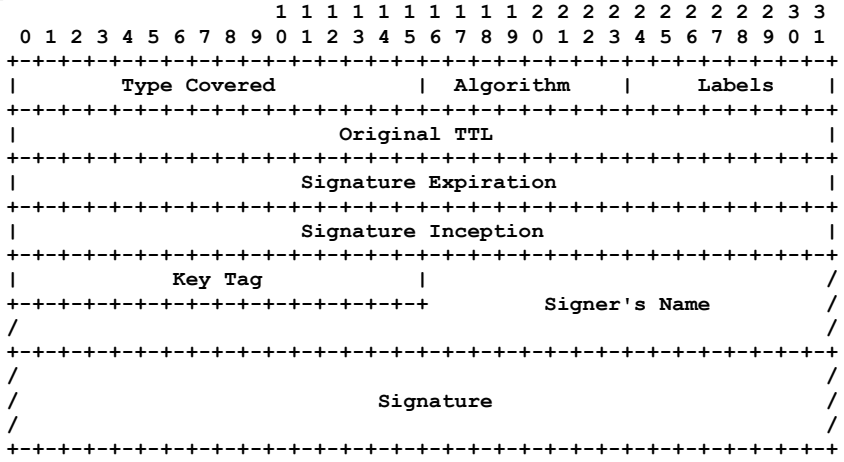
Поле Type для RRSIG RR имеет значение 46.

Записи RRSIG RR не зависят от класса.

Запись RRSIG RR **должна** относиться к тому же классу, что и подписываемый ею набор RRset.

Значение TTL для RRSIG RR **должно** совпадать с TTL подписываемого Rrset. Имеется исключение из правил [RFC2181] для TTL отдельных записей RR внутри RRset – отдельные RRSIG RR с общим именем владельца будут иметь разные значения TTL, если покрываемые ими наборы RRset имеют разные значения TTL.

3.1. Формат передачи RRSIG RDATA



RDATA для записей RRSIG RR включает двухоктетное поле Type Covered, одноктетные поля Algorithm и Labels, четырехоктетные поля Original TTL, Signature Expiration и Signature Inception, двухоктетное поле Key tag, а также поля Signer's Name и Signature.

3.1.1. Type Covered

Поле Type Covered указывает тип RRset, покрываемого этой записью RRSIG.

3.1.2. Algorithm Number

Поле Algorithm Number указывает криптографический алгоритм, использованный для создания подписи. Список алгоритмов DNSSEC приведён в Приложении A.1

3.1.3. Labels

Поле Labels указывает число меток в исходной записи RRSIG RR для имени владельца. С помощью этого поля проверяющий (validator) определяет был ли ответ синтезирован из шаблона. При шаблонном ответе проверяющий может использовать это поле для определения имени владельца, использованного при генерации подписи.

Для проверки подписи проверяющему требуется исходное имя владельца, которое было использовано при создании подписи. Если исходное имя содержит метку шаблона (*), имя владельца может быть определено в процессе отклика, в течение которого проверяющий восстанавливает исходное имя владельца для проверки подписи. Использование поля Labels для восстановления исходного имени владельца описано в [RFC4035].

В значении поля Labels **недопустимо** учитывать пустую (корневую) метку, завершающую имя владельца или метку-шаблон (при её наличии). Значение поля Labels **должно** быть не больше числа меток в имени владельца RRSIG. Например, для www.example.com. поле Labels имеет значение 3, а для *.example.com. - 2. Для корня (.) поле Labels имеет значение 0.

Хотя шаблонные метки не учитываются в поле Labels записей RRSIG RR, такие метки являются частью имени владельца RRset при генерации и проверке подписи.

3.1.4. Original TTL

Поле Original TTL указывает значение TTL для покрываемого подписью RRset, как оно появляется в полномочной зоне.

Поле Original TTL требуется по причине того, что кэширующие преобразователи декрементируют значения TTL для кэшируемых RRset. Для проверки подписи проверяющему требуется знать исходное значение TTL. В [RFC4035] описано использование значения поля Original TTL для восстановления исходного значения TTL.

3.1.5. Signature Expiration и Signature Inception

Поля Signature Expiration и Signature Inception указывают период действия подписи. Запись RRSIG **недопустимо** использовать для аутентификации до момента создания и после завершения срока её действия.

Поля Signature Expiration и Signature Inception задают дату и время в форме 32-битового целого числа без знака с сетевым порядком байтов, указывающего количество секунд с 00:00:00 часов UTC 1 января 1970 г, без учёта високосных секунд. Максимальный интервал, который может быть задан в таком формате, составляет приблизительно 136 лет. Запись RRSIG RR может иметь поле Expiration, значение которого численно меньше значения поля Inception, если значение первого находится вблизи границы 32-битового диапазона или подпись имеет очень большой срок действия. По этой причине все операции сравнения для этих полей **должны** использовать арифметику порядковых номеров, описанную в [RFC1982]. Прямым следствием этого является то, что значения полей не могут указывать даты, отличающиеся от текущей более, чем на 68 лет в ту или иную сторону.

3.1.6. Key Tag

Поле Key Tag содержит значение тега ключа DNSKEY RR, который «заверяет» эту подпись, с сетевым порядком байтов. Расчёт значений Key Tag описан в Приложении B.

3.1.7. Signer's Name

Значение поля Signer's Name указывает имя владельца записи DNSKEY RR, которое проверяющий предложил использовать для проверки этой подписи. Поле Signer's Name **должно** содержать имя зоны, покрываемой набором RRset. Отправителю **недопустимо** использовать сжатие имени DNS для поля Signer's Name при передаче RRSIG RR.

3.1.8. Signature

Поле Signature содержит криптографическую подпись, покрывающую поля RRSIG RDATA (за исключением поля Signature) и набор RRset, указанный именем владельца RRSIG, класс RRSIG и RRSIG Type Covered. Формат поля подписи зависит используемого алгоритма и должен описываться в соответствующих документах.

3.1.8.1. Расчёт подписи

Подпись учитывает RRSIG RDATA (за исключением поля Signature) и набор RRset, указанный именем владельца RRSIG, а также поля класса RRSIG и RRSIG Type Covered. RRset имеет каноническую форму (см. раздел 6) и набор RR(1),...RR(n) подписывается следующим образом:

```
signature = sign(RRSIG_RDATA | RR(1) | RR(2)... )
```

где | означает конкатенацию (слияние);

RRSIG_RDATA представляет формат передачи полей RRSIG RDATA с канонической формой Signer's Name и исключённым полем Signature.

```
RR(i) = owner | type | class | TTL | RDATA length | RDATA
```

owner - полное (fully qualified) имя владельца RRset в канонической форме (для записей RR с шаблонами имён владельца символ шаблона включается в имя);

все RR **должны** иметь то же имя владельца, которое указано в RRSIG RR;

все RR **должны** иметь тот же класс, который указан в RRSIG RR;

все RR в RRset **должны** иметь тип RR, указанный в поле Type Covered записи RRSIG RR;

все RR в RRset **должны** иметь значение TTL, указанное в поле Original TTL записи RRSIG;

все имена DNS в полях RDATA каждой записи RR **должны** иметь каноническую форму;

набор RRset **должен** быть отсортирован в каноническом порядке.

Подробная информация о канонической форме и упорядочении RRset приведена в параграфах 6.2 и 6.3.

3.2. Формат представления RRSIG RR

Ниже показан формат представления части RDATA.

Поле Type Covered представляется мнемоникой типа RR. Если мнемоническое значение не известно, **должно** применяться представление TYPE, описанное в разделе 5 [RFC3597].

Значение поля Algorithm **должно** быть представлено десятичным целым числом без знака или мнемоническим значением, как указано в Приложении A.1.

Значение поля Labels **должно** представляться десятичным целым числом без знака.

Значение поля Original TTL **должно** представляться десятичным целым числом без знака.

Значения полей Signature Expiration Time и Signature Inception Time **должны** представляться десятичным целым числом без знака, указывающим число секунд с начала суток 1 января 1970 года, или указывать время часового пояса UTC в формате YYYYMMDDHHmmSS, где

YYYY - год (0001-9999, см. параграф 3.1.5);

MM - месяц (01-12);

DD - число месяца (01-31);

HH - число часов в 24-часовом формате (00-23);

mm - число минут (00-59);

SS - число секунд (00-59).

Отметим, что форматы представления времени всегда можно различить, поскольку формат YYYYMMDDHHmmSS всегда использует 14 цифр, а 32-битовое целое число без знака не может иметь более 10 цифр.

Значение поля Key Tag **должно** представляться десятичным целым числом без знака.

Значение поля Signer's Name **должно** представляться в виде доменного имени.

Поле Signature указывает подпись в представлении Base64, где разрешено использовать пробельные символы (см. параграф 2.2).

3.3. Пример RRSIG RR

Приведённая ниже запись RRSIG RR содержит набор A RRset для хоста host.example.com.

```
host.example.com. 86400 IN RRSIG A 5 3 86400 20030322173103 (
                           20030220173103 2642 example.com.
                           oJB1W6WNGv+ldvQ3WDG0MQkg5IEhjRip8WTr
                           PYGv07h108dUKGMeDPKijVCHX3DDKdfb+v6o
                           B9wfuh3DTJXUAfI/M0zmO/zz8bW0Rznl8O3t
                           GNazPwQKkRN20XPXV6nwwfoXmJQbsLNrLfkG
                           J5D6fwFm8nN+6pBzeDQfsS3Ap3o= )
```

Первые 4 поля указывают имя владельца, TTL, класс, и тип RR (RRSIG). Значение A представляет поле Type Covered. Значение 5 указывает использованный для создания подписи алгоритм (RSA/SHA1). Значение 3 указывает число меток в исходном имени владельца. Значение 86400 в RRSIG RDATA представляет собой Original TTL для покрываемого подписью набора A RRset. Значения 20030322173103 и 20030220173103 указывают даты окончания и создания подписи, соответственно. 2642 указывает Key Tag, а example.com. - имя подписавшего (Signer's Name). Остальная часть записи - представление Base64 для подписи.

Отметим, что комбинация имени владельца, класса и покрываемого типа в RRSIG RR указывает, что эта запись RRSIG покрывает набор A RRset для host.example.com. Значение Label = 3 показывает, что шаблонное расширение не используется. Поля Algorithm, Signer's Name и Key Tag показывают, что эта подпись может быть аутентифицирована с помощью DNSKEY RR зоны example.com, где используется алгоритм 5 и тег ключа 2642.

4. Запись NSEC RR

Запись NSEC содержит два отдельных элемента - имя следующего владельца (в каноническом порядке для зоны), содержащего полномочные данные, или точка передачи полномочий (делегирования) NS RRset и множество типов RR, присутствующих в имени владельца NSEC RR [RFC3845]. Полный набор записей NSEC RR в зоне показывает, какие из полномочных RRset имеются в зоне, а также формирует цепочку имён полномочных владельцев в зоне. Эта информация служит для предоставления аутентифицированных сведений об отсутствии данных в DNS, как описано в [RFC4035].

Поскольку каждое полномочное имя в зоне должно быть частью цепочки NSEC, для имён, содержащих CNAME RR, должны присутствовать записи NSEC RR. Это является отличием от традиционной спецификации DNS [RFC1034], где сказано, что CNAME представляется для имени и является единственным типом, дозволенным для этого имени. Записи RRSIG (см. раздел 3) и NSEC **должны** существовать для имён, имеющих запись CNAME в подписанной зоне.

В документе [RFC4035] описано, как подписывающий зону точно определяет включаемые в зону записи NSEC RR.

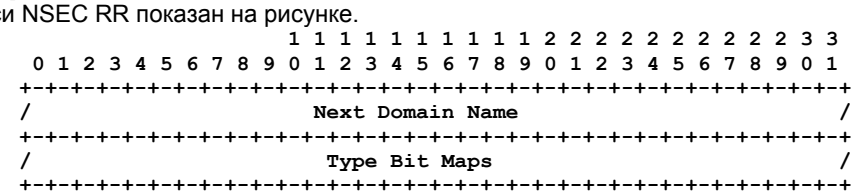
Тип NSEC RR имеет значение 47.

NSEC RR не зависит от класса.

В записи NSEC RR **следует** указывать такое же значение TTL, которое приведено в поле минимального TTL записи SOA. Такое поведение следует духу негативного кэширования ([RFC2308]).

4.1. Формат передачи NSEC RDATA

Формат RDATA записи NSEC RR показан на рисунке.



4.1.1. Поле Next Domain Name

Поле Next Domain содержит имя следующего владельца (в каноническом порядке зоны, см. параграф 6.1), который имеет полномочные данные или содержит точку передачи полномочий (делегирования) NS RRset. Значением поля Next Domain Name в последней записи NSEC данной зоны является имя на вершине зоны (имя владельца записи SOA RR для зоны). Это показывает, что имя владельца NSEC RR является последним при каноническом упорядочении имён в зоне.

Отправителю **недопустимо** использовать сжатие имён DNS для полей Next Domain Name при передаче NSEC RR.

Имена владельцев наборов RRset, для которых данная зона не является полномочной (такие, как склеивающие записи) **недопустимо** указывать в Next Domain Name, если не существует хотя бы одного полномочного RRset с таким же именем владельца.

4.1.2. Поле Type Bit Maps

Поле Type Bit Maps указывает типы RRset, существующие с именем владельца NSEC RR.

Пространство типов RR разделено на 256 окон (блоков), каждое из которых представляет 8 младших битов 16-битового пространства типов RR. Каждый блок, имеющий хотя бы один активный тип RR, кодируется с помощью 1-октетного номера окна (0 - 255), 1-октетного размера битового отображения (1 - 32), указывающего число октетов, используемых для битового отображения (bitmap) блока, и до 32 октетов (256 битов) самого битового отображения.

Блоки представляются в NSEC RR RDATA по порядку возрастания числовых значений.

```
Type Bit Maps = ( Window Block # | Bitmap Length | Bitmap )+
```

где | обозначает конкатенацию.

Каждое битовое отображение представляет 8 младших битов типов RR в блоке с использованием сетевого порядка битов. Первый бит имеет номер 0. Для оконного блока 0 бит 1 соответствует RR типа 1 (A), бит 2 - RR типа 2 (NS) и т. д. Для блока 1, бит 1 соответствует RR типа 257, в бит 2 - RR типа 258. Установленный бит показывает, что набор RRset данного типа присутствует для имени владельца NSEC RR, сброшенный бит говорит об отсутствии RRset данного типа для имени владельца NSEC RR.

Биты, представляющие псевдотипы, **должны** быть сброшены, поскольку псевдотипы не присутствуют в данных зоны. Установленные биты псевдотипов **должны** игнорироваться при чтении.

Блоки, не содержащие типов, включать **недопустимо**. Нулевые октеты в конце битового отображения **должны** опускаться. Размер битового отображения каждого блока определяется кодом типа с наибольшим значением в данном блоке среди множества типов RR, присутствующих для имени владельца NSEC RR. Не указанные в спецификации октеты в конце **должны** интерпретироваться, как нулевые октеты.

Битовое отображение для NSEC RR в точке передачи полномочий требует отдельного внимания. Биты, соответствующие делегирования NS RRset и типы RR, для которых родительская зона имеет полномочные данные, **должны** быть установлены, биты, соответствующие всем отличным от NS RRset, для которых родительская зона не является полномочной, **должны** быть сброшены.

В зону **недопустимо** включать NSEC RR для каких-либо доменных имён, которые имеют лишь склеивающие записи.

4.1.3. Включение шаблонных имён в NSEC RDATA

При наличии в зоне шаблонного имени владельца символ шаблона (*) трактуется буквально и одинаково для всех прочих имён владельцев при генерации записей NSEC RR. Шаблонные имена указываются в поле Next Domain Name без расширения шаблона. Влияние шаблонов на аутентифицированные данные от отсутствии (denial of existence) рассмотрено в [RFC4035].

4.2. Формат представления NSEC RR

Представление части RDATA использует следующий формат:

поле Next Domain Name представляется, как доменное имя;

поле Type Bit Maps представляется в форме последовательности мнемонических обозначений типов RR, а при отсутствии мнемоники **должно** использоваться представление TYPE, описанное в разделе 5 [RFC3597].

4.3. Пример NSEC RR

Приведённая ниже запись NSEC RR идентифицирует наборы RRset, связанные с alfa.example.com. И указывает следующее после alfa.example.com. полномочное имя.

```
alfa.example.com. 86400 IN NSEC host.example.com. ( A MX RRSIG NSEC TYPE1234 )
```

Первые четыре текстовых поля задают имя, TTL, класс и тип RR (NSEC). Поле host.example.com. указывает следующее в каноническом порядке после alfa.example.com. полномочное имя. Мнемонические обозначения A, MX, RRSIG, NSEC и TYPE1234 показывают наличие наборов (RRset) A, MX, RRSIG, NSEC и TYPE1234, связанных с именем alfa.example.com.

Часть RDATA записи NSEC RR будет представлена в виде

```
0x04 'h'  'o'  's'  't'
0x07 'e'  'x'  'a'  'm'  'p'  'l'  'e'
0x03 'c'  'o'  'm'  0x00
0x00 0x06 0x40 0x01 0x00 0x00 0x00 0x03
0x04 0x1b 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x20
```

В предположении, что проверяющий может аутентифицировать эту запись NSEC, она может быть использована для подтверждения отсутствия beta.example.com или подтверждения отсутствия записи AAAA, связанной с именем alfa.example.com. Аутентификация данных об отсутствии рассматривается в [RFC4035].

5. Запись DS RR

Запись DS RR указывает на DNSKEY RR и используется в процессе аутентификации DNS с помощью ключа DNSKEY. Запись DS RR указывает DNSKEY RR путём сохранения тега ключа, номера алгоритма и отпечатка (digest) DNSKEY RR. Отметим, что, несмотря на достаточность отпечатка для идентификации открытого ключа, сохранение тега ключа и алгоритма помогает повысить эффективность процесса идентификации. Путём аутентификации записи DS распознаватель может аутентифицировать запись DNSKEY RR, на которую указывает данная DS. Процесс аутентификации ключей описан в [RFC4035].

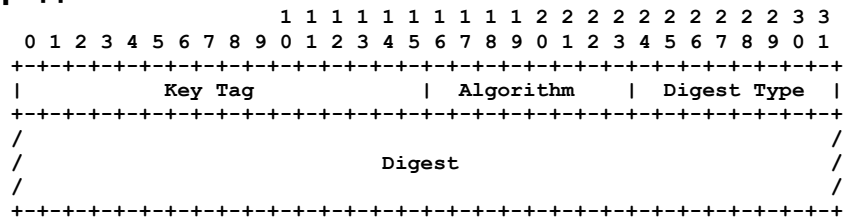
Запись DS RR и соответствующая ей DNSKEY RR имеют общее имя владельца, но хранятся в разных местах. Запись DS RR появляется только на верхней (родительской) стороне делегирования и относится к полномочным данным родительской зоны. Например, DS RR для example.com хранится в зоне com (родительская), а не example.com (дочерняя). Соответствующая запись DNSKEY RR сохраняется в зоне example.com (дочерняя). Это упрощает управление зонами DNS и их подписание, но требует специальной обработки откликов для DS RR (см. [RFC4035]).

Номер типа для записи DS равен 43.

Записи DS не зависят от класса.

Для записей DS RR нет особых требований к TTL.

5.1. Формат передачи DS RDATA



RDATA для DS RR состоит из 2-октетного поля Key Tag, 1-октетных полей Algorithm и Digest Type, а также поля Digest.

5.1.1. Поле Key Tag

Поле Key Tag содержит тег ключа для записи DNSKEY RR, указанной в записи DS, с использованием сетевого порядка байтов.

Поле Key Tag в записях DS RR идентично одноимённому поля в RRSIG RR. Расчёт тегов описан в Приложении B.

5.1.2. Поле Algorithm

Поле Algorithm содержит номер алгоритма для записи DNSKEY RR, указанной в записи DS.

Номер алгоритма в DS RR идентичен номерам алгоритмов, используемым в записях RRSIG и DNSKEY. Номера типов алгоритмов приведены в Приложении A.1.

5.1.3. Поле Digest Type

DS RR указывает на DNSKEY RR путём включения отпечатка (digest) данной DNSKEY RR. Поле Digest Type указывает алгоритм, использованный для создания отпечатка. Алгоритма создания отпечатков перечислены в Приложении A.2.

5.1.4. Поле Digest

Запись DS указывает на DNSKEY RR путём включения отпечатка (digest) данной DNSKEY RR .

Отпечаток рассчитывается путём использования алгоритма создания отпечатка к конкатенации канонической формы полного имени владельца DNSKEY RR и DNSKEY RDATA.

```
digest = digest_algorithm( DNSKEY owner name | DNSKEY RDATA);
```

"|" указывает конкатенацию (слияние)

```
DNSKEY RDATA = Flags | Protocol | Algorithm | Public Key.
```

Размер отпечатка может меняться в зависимости от применяемого алгоритма и размера DNSKEY RR. На момент подготовки этого документа единственным алгоритмом создания отпечатков был SHA-1, обеспечивающий на выходе 20 октетов.

5.2. Обработка DS RR при проверке откликов

Записи DS RR связывают аутентификационные цепочки через границы зон, поэтому для DS RR требуется дополнительная обработка. DNSKEY RR, указанная в DS RR, **должна** быть ключом зоны DNSSEC. В поле флагов DNSKEY RR Flags **должен** быть установлен бит 7. Если флаги DNSKEY не указывают ключ зоны DNSSEC, записи DS RR (и указываемые ими DNSKEY RR) **недопустимо** использовать в процессе проверки.

5.3. Формат представления DS RR

Формат представления части RDATA показан ниже.

Поле Key Tag **должно** быть представлено десятичным целым числом без знака.

Поле Algorithm **должно** быть представлено десятичным целым числом без знака или мнемоническим обозначением алгоритма из числа приведённых в Приложении A.1.

Поле Digest Type **должно** быть представлено десятичным целым числом без знака.

Поле Digest **должно** быть представлено в виде последовательности 16-ричных цифр (независимо от регистра символов). В шестнадцатеричной последовательности допускаются пробельные символы.

5.4. Пример DS RR

Ниже приведён пример записи DNSKEY RR и соответствующей ей DS RR.

```
dskey.example.com. 86400 IN DNSKEY 256 3 5 ( AQOeiiR0GOMYkDshWoSKz9Xz
                                           fwJr1AYtsmx3TGkJaNXVbfi/
                                           2pHm822aJ5iI9BMzNXxeYCmZ
```

```
                                        DRD99WYwYqUSdjMmmAphXdvx
                                        egXd/M5+X7OrzKBaMbCVdFLU
                                        Uh6DhweJBjEVv5f2wwjM9Xzc
                                        nOf+EPbtG9DMBmADjFDc2w/r
                                        ljwvFw==
                                        ) ;  key id = 60485
dskey.example.com. 86400 IN DS 60485 5 1 ( 2BB183AF5F22588179A53B0A
                                           98631FAD1A292118 )
```

Первые 4 поля указывают имя, TTL, класс и тип RR (DS). Значение 60485 является тегом ключа, соответствующего dskey.example.com. DNSKEY RR и значение 5 указывают алгоритм, используемый этой записью dskey.example.com. DNSKEY RR. Значение 1 указывает алгоритм создания отпечатка, а остальная часть RDATA содержит отпечаток в 16-ричном формате.

6. Каноническая форма и порядок RR

В этом разделе определяется каноническая форма записей о ресурсах, канонический порядок имён DNS и канонический порядок записей о ресурсах в наборах RRset. Канонический порядок имён требуется при создании цепочек имён NSEC. Каноническая форма RR и канонический их порядок в RRset требуются для создания и проверки записей RRSIG RR.

6.1. Канонический порядок имён DNS

Для использования в целях защиты DNS имена владельцев упорядочиваются путём трактовки индивидуальных меток, как беззнаковых строк октетов с выравниванием по левому краю. Отсутствующий октет помещается впереди октета с нулевым значением, буквы верхнего регистра трактуются, как соответствующие буквы нижнего регистра кода US-ASCII.

Для определения канонического порядка множества имён DNS сначала выполняется сортировка имён по старшим (правым) меткам. Для имён с одинаковой правой меткой выполняется сортировка по значению следующей по старшинству метки и т. д.

Ниже приведён набор имён DNS, отсортированных в каноническом порядке. Старшей меткой в данном случае является example. Поэтому первым в списке появляется имя, содержащее лишь старшую метку, за ним следует имя a.example, а заканчивается список именами z.example. В каждом уровне осуществляется внутренняя сортировка меток.

```
example
a.example
yljkjljk.a.example
Z.a.example
zABC.a.EXAMPLE
z.example
\001.z.example
*.z.example
\200.z.example
```

6.2. Каноническая форма RR

Для использования в целях защиты DNS в качестве канонической формы RR используется формат передачи RR, где:

1. каждое доменное имя в RR является несжатым (нет компрессии DNS) и полным (fully qualified);
2. все символы верхнего регистра US-ASCII в имени владельца RR заменяются соответствующими символами нижнего регистра US-ASCII;
3. [1]
4. если имя владельца RR является шаблонным, оно сохраняется без преобразования (с сохранением символа *);
5. для поля TTL в записи RR устанавливается исходное значение, которое указано в полномочной зоне-источнике или в поле Original TTL покрывающей записи RRSIG RR.

6.3. Канонический порядок RR в наборе RRset

Для использования в целях защиты DNS записи RR с одинаковыми именами владельца, классом и типом сортируются путём трактовки части RDATA канонической формы каждой RR, как строки беззнаковых октетов с выравниванием по левом краю. При этом отсутствующий октет считается предшествующим октету с нулевым значением.

В [RFC2181] указано, что в RRset не допускается наличие дубликатов записей (множество RR с одинаковым именем владельца, классом, типом и RDATA). Следовательно, если реализация обнаруживает дубликат RR при преобразовании RRset в каноническую форму, это должно трактоваться, как протокольная ошибка. Если реализация считает нужным самостоятельно обрабатывать такие ошибки в целях повышения отказоустойчивости (либеральное отношение к входным данным), она **должна** удалить все дубликаты RR до приведения RRset в каноническую форму.

7. Взаимодействие с IANA

Этот документ не требует решения каких-либо вопросов с агентством IANA, поскольку все используемые в документе параметры протокола уже были выделены в предшествующих спецификациях. Однако в силу продолжительности и некоторой путаности процесса развития DNSSEC требуются некоторые разъяснения для чего в этом разделе описано текущее состояние реестров IANA и других протокольных параметров, связанных с DNSSEC.

Дополнительное рассмотрение связанных с IANA вопросов приведено в [RFC4035].

[1] Текст этого пункта в исходном документе «если типом RR указан NS, MD, MF, CNAME, SOA, MB, MG, MR, PTR, HINFO, MINFO, MX, HINFO, RP, AFSDB, RT, SIG, PX, NXT, NAPTR, KX, SRV, DNAME, A6, RRSIG или NSEC, все заглавные буквы US-ASCII в именах DNS, содержащихся в RDATA, заменяются соответствующими строчными буквами US-ASCII;» является ошибочным. См. https://www.rfc-editor.org/errata_search.php?eid=1062. *Прим. перев.*

Типы записей о ресурсах DNS. В [RFC2535] выделены типы 24, 25 и 30 для записей типа SIG, KEY и NXT, соответственно. В [RFC3658] выделен тип 43 для записей DS. В [RFC3755] выделены типы 46, 47 и 48 для RRSIG, NSEC и DNSKEY, соответственно. В [RFC3755] тип 30 (NXT) указан, как устаревший (Obsolete) и ограничено применение типов 24 (SIG) и 25 (KEY) до SIG(0) в протоколе защиты транзакций, описанном в [RFC2931], и KEY RR [RFC2930].

Номера алгоритмов защиты DNS. В [RFC2535] создан реестр IANA для значений поля DNSSEC Resource Record Algorithm и выделены значения 1 - 4 и 252 - 255. В [RFC3110] выделено значение 5. В [RFC3755] были внесены изменения в реестр с включением флагов для каждого элемента, связанных с использованием защитных расширений DNS. Каждый элемент может указывать алгоритм, который может применяться для подписания зоны и/или защиты транзакций (см. [RFC2931]). Значения 6 - 251 доступны для выделения в процессах стандартизации IETF ([RFC3755]). Полный список номеров алгоритмов защиты и статус их применения в DNSSEC на момент подготовки документа приведены в Приложении A.

В [RFC3658] создан реестр IANA для значений DNSSEC DS Digest Type и выделены значение 0 в качестве резервного и 1 для SHA-1.

Значения протоколов ключей. В [RFC2535] создан реестр IANA для значений KEY Protocol, но в [RFC3445] переопределены все значения этого реестра, за исключением резервного номера 3 и реестр был закрыт. Реестр остаётся закрытым и во всех записях KEY и DNSKEY октет Protocol должен иметь значение 3.

Биты флагов в записях KEY и DNSKEY. В [RFC3755] создан реестр IANA для битов флагов в записях DNSSEC KEY и DNSKEY. Изначально были выделены значения лишь для битов 7 (флаг ZONE) и 15 (флаг SEP[1], см. [RFC3757]). Как указано в [RFC3755], биты 0 - 6 и 8 - 14 доступны для выделения в процессах стандартизации IETF.

8. Вопросы безопасности

Этот документ описывает формат четырёх записей о ресурсах DNS, используемых защитными расширениями DNS, и представляет алгоритмы расчёта тегов для открытых ключей. За исключением перечисленных ниже элементов, сами по себе записи о ресурсах не создают проблем безопасности. Дополнительное рассмотрение вопросов безопасности в связи с использованием этих записей приведено в [RFC4033] и [RFC4035].

Запись DS указывает на DNSKEY RR путём использования криптографического отпечатка, типа алгоритма и тега ключа. Записи DS предназначены для идентификации имеющихся DNSKEY RR, но теоретически возможно их использование атакующими для генерации DNSKEY, соответствующих всем полям записи DS. Возможность создания соответствующих DNSKEY зависит от типа применяемого алгоритма создания отпечатков. В настоящее время определен только алгоритм SHA-1 и рабочая группа предполагает, что создание открытого ключа, который будет соответствовать алгоритму, тегу ключа и отпечатку SHA-1, представленным в записи DS, является достаточно сложной проблемой и такие атаки в настоящее время не представляют угрозы.

Теги ключей служат для помощи в эффективном выборе записей DNSKEY, но не являются уникальными идентификаторами таких записей. Возможны ситуации, когда две разных записи DNSKEY RR будут иметь одинаковые имена владельцев, тип алгоритма и тег ключа. Реализации, использующие для выбора DNSKEY RR только тег ключа, могут в некоторых случаях выбирать не тот открытый ключ. Более подробно этот вопрос рассмотрен в Приложении B.

Таблица алгоритмов в Приложении A и алгоритмы расчёта тегов ключей в Приложении B включают для полноты алгоритм RSA/MD5, но его применение **не рекомендуется** (см. [RFC3110]).

9. Благодарности

Этот документ был создан на основе идей и результатов работы группы DNS Extensions, а также обсуждений в списке рассылки этой группы. Редакторы рады выразить свою признательность за комментарии и предложения, полученные в процессе пересмотра этих защитных расширений. Хотя указать всех, кто принял участие в десятилетней разработке DNSSEC, просто невозможно, в [RFC4033] приведён список некоторых активных участников обсуждения документов.

10. Литература

10.1. 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.

[RFC1982] Elz, R. and R. Bush, "Serial Number Arithmetic", RFC 1982, August 1996.

[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.

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

[RFC2536] Eastlake 3rd, D., "DSA KEYs and SIGs in the Domain Name System (DNS)", RFC 2536, March 1999.

[RFC2931] Eastlake 3rd, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, September 2000.

[RFC3110] Eastlake 3rd, D., "RSA/SHA-1 SIGs and RSA KEYs in the Domain Name System (DNS)", RFC 3110, May 2001.

[RFC3445] Massey, D. and S. Rose, "Limiting the Scope of the KEY Resource Record (RR)", RFC 3445, December 2002.

[RFC3548] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 3548, July 2003.

[RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource Record (RR) Types", RFC 3597, September 2003.

[RFC3658] Gudmundsson, O., "Delegation Signer (DS) Resource Record (RR)", RFC 3658, December 2003.

[1] Secure Entry Point - защищённая точка входа.

[RFC3755] Weiler, S., "Legacy Resolver Compatibility for Delegation Signer (DS)", RFC 3755, May 2004.

[RFC3757] Kolkman, O., Schlyter, J., and E. Lewis, "Domain Name System KEY (DNSKEY) Resource Record (RR) Secure Entry Point (SEP) Flag", RFC 3757, April 2004.

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

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

10.2. Дополнительная литература

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

[RFC2537] Eastlake 3rd, D., "RSA/MD5 KEYs and SIGs in the Domain Name System (DNS)", RFC 2537, March 1999.

[RFC2539] Eastlake 3rd, D., "Storage of Diffie-Hellman Keys in the Domain Name System (DNS)", RFC 2539, March 1999.

[RFC2930] Eastlake 3rd, D., "Secret Key Establishment for DNS (TKEY RR)", RFC 2930, September 2000.

[RFC3845] Schlyter, J., "DNS Security (DNSSEC) NextSECure (NSEC) RDATA Format", RFC 3845, August 2004.

Приложение A. Типы алгоритмов и отпечатков DNSSEC

Защитные приложения DNS разрабатываются так, чтобы они не зависели от используемого криптографического алгоритма. Записи DNSKEY, RRSIG и DS используют DNSSEC Algorithm Number для идентификации используемого записью криптографического алгоритма. Записи DS также указывают Digest Algorithm Number для идентификации алгоритма цифровой подписи, использованного для создания записи DS. Определённые в настоящее время алгоритмы (Algorithm) и типы отпечатков (Digest Type) перечислены ниже. По мере развития криптографии могут добавляться новые алгоритмы и типы отпечатков.

Осведомленные о DNSSEC преобразователи и серверы имён **должны** реализовать все **обязательные** (MANDATORY) алгоритмы.

A.1. Типы алгоритмов DNSSEC

Записи DNSKEY, RRSIG и DS используют 8-битовые целые числа для идентификации используемого алгоритма защиты. Эти значения хранятся в поле Algorithm number элемента RDATA данной записи.

Некоторые алгоритмы подходят только для подписывания зон (DNSSEC), другие - только для механизмов защиты транзакций (SIG(0) и TSIG), а некоторые - для обоих случаев. Пригодные для подписывания зон алгоритмы могут присутствовать в записях DNSKEY, RRSIG и DS RR, пригодные для защиты транзакций - в записях SIG(0) и KEY, как описано в [RFC2931].

Значение	*Алгоритм*	*Обозначение*	*Подпись зоны*	*Документ*	*Статус*
0	резерв				
1	RSA/MD5	RSAMD5	нет	RFC2537	Не рекомендуется
2	Diffie-Hellman	DH	нет	RFC2539	-
3	DSA/SHA-1	DSA	есть	RFC2536	Опционально
4	Elliptic Curve	ECC		В работе	-
5	RSA/SHA-1	RSASHA1	есть	RFC3110	Обязательно
252	Indirect	INDIRECT	нет		-
253	Private	PRIVATEDNS	есть	см. ниже	Опционально
254	Private	PRIVATEOID	есть	см. ниже	Опционально
255	резерв				
6 - 251	Резерв для стандартизации IETF				

A.1.1. Частные типы алгоритмов

Номер 253 зарезервирован для частных применений и никогда не будет выделен для конкретного алгоритма. Область открытого ключа в DNSKEY RR и область подписи в RRSIG RR начинаются с доменного имени в формате передачи, которое **недопустимо** сжимать. Доменное имя указывает приватный алгоритм для использования и оставшаяся часть области открытого ключа определяется этим алгоритмом. Объектам следует использовать только те доменные имена, для которых они контролируют применение частных алгоритмов.

Номер 254 зарезервирован для частных применений и никогда не будет выделен для конкретного алгоритма. Область открытого ключа в DNSKEY RR и область подписи в RRSIG RR начинаются с (беззнакового) байта размера, за которым следует BER-представление идентификатора объекта (ISO OID[1]) указанного размера. OID указывает используемый приватный алгоритм и остальная часть области определяется этим алгоритмом. Объектам следует использовать только те идентификаторы OID, для которых они контролируют применение частных алгоритмов.

A.2. Типы отпечатков DNSSEC

Поле Digest Type в записях DS идентифицирует криптографический алгоритм создания отпечатка (digest), использованный для этой записи. Определённые в настоящее время алгоритмы указаны в таблице.

Значение	*Алгоритм*	*Статус*
0	резерв	-
1	SHA-1	Обязательно
2 - 255	Не распределены	-

[1]Object Identifier.

Приложение B. Расчёт Key Tag

Поле Key Tag в записях типов RRSIG и DS обеспечивает механизм эффективного выбора открытого ключа. В большинстве случаев комбинация имени владельца, алгоритма и тега ключа могут эффективно идентифицировать запись DNSKEY. Записи обоих типов RRSIG и DS имеют соответствующие записи DNSKEY. Поля Key Tag в записях RRSIG и DS могут помочь в эффективном выборе соответствующей DNSKEY RR из множества таких записей.

Однако важно подчеркнуть, что тег ключа не является уникальным идентификатором. Теоретически возможно существование двух разных DNSKEY RR с совпадающими именами владельца, алгоритмом и тегом ключа. Тег ключа может применяться для ограничения числа рассматриваемых кандидатов, но не является уникальным идентификатором записи DNSKEY. Реализациям **недопустимо** предполагать, что тег ключа точно указывает DNSKEY RR.

Теги ключа определяются одинаково для всех алгоритмов DNSKEY, за исключением алгоритма 1 (для этого алгоритма определение тега ключа приведено в Приложении B.1). Тег ключа получается путём суммирования 2-октетных блоков DNSKEY RDATA в формате передачи. Сначала RDATA (в формате передачи) трактуется, как последовательность 2-октетных групп. Эти группы складываются с использованием по крайней мере 32-битового формата суммы с сохранением всех битов переноса. После этого биты переноса добавляются к результату из которого в качестве тега ключа используется только 16 младших битов. Следует отметить, что переносы, возникающие при добавлении при добавлении битов переноса, игнорируются. Это, в свою очередь, означает, что расчёт ключа зачастую (но не всегда) совпадает с сокращением по модулю 65535[1].

Ниже приведён образец реализации алгоритма создания тега ключа на языке ANSI C с использованием компоненты RDATA записи DNSKEY RR в качестве входных данных. Не требуется использовать именно приведённый код, но численное значение Key Tag **должно** быть идентично значению, которое для тех же входных данных будет возвращать приведённый образец кода.

Отметим, что алгоритм расчёта Key Tag почти (но не полностью) идентичен алгоритму расчёта контрольных сумм с дополнением до 1, используемому во многих протоколах Internet. Значения Key Tags **должны** рассчитываться с использованием приведённого здесь алгоритма, а не контрольной суммы с дополнением до 1.

Приведённый ниже код ANSI C служит для расчёта Key Tag. Этот образец реализации применим ко всем типа алгоритмов за исключением алгоритма 1 (см. Приложение B.1). Входными данными является компонента RDATA (в формате передачи) записи DNSKEY RR. Код оптимизирован в части ясности, а не эффективности.

```
/*
 * Предполагается, что размер int не менее 16 битов.
 * Первый октет тега ключа содержит 8 старших битов возвращаемого значения;
 * Второй октет тега ключа содержит 8 младших битов возвращаемого значения.
 */

unsigned int
keytag (
        unsigned char key[],  /* компонента RDATA записи DNSKEY RR */
        unsigned int keysize  /* RDLENGTH */
       )
{
        unsigned long ac;     /* предполагается размер не менее 32 битов */
        int i;                /* переменная цикла */

        for ( ac = 0, i = 0; i < keysize; ++i )
                ac += (i & 1) ? key[i] : key[i] << 8;
        ac += (ac >> 16) & 0xFFFF;
        return ac & 0xFFFF;
}
```

B.1. Key Tag для алгоритма 1 (RSA/MD5)

Теги ключей для алгоритма 1 (RSA/MD5) определяются не так, как для всех остальных алгоритмов в силу исторических причин. Для записи DNSKEY RR с алгоритмом 1 тег ключа определяется, как 16 старших из 24 младших битов в модуле открытого ключа (иными словами, 3-й и 2-й[2] с конца октеты модуля открытого ключа).

Следует отметить, что использование алгоритма 1 **не рекомендуется**.

Адреса авторов

Roy Arends

Telematica Instituut

Brouwerijstraat 1

7523 XC Enschede

NL

EMail: roy.arends@telin.nl

Rob Austein

Internet Systems Consortium

[1]Текст этого абзаца был переведён с учётом обнаруженной в нем ошибки. См. https://www.rfc-editor.org/errata_search.php?eid=4552 и https://www.rfc-editor.org/errata_search.php?eid=2681. *Прим. перев.*

[2]В оригинале ошибочно указаны 4-й и 3-й. См. https://www.rfc-editor.org/errata_search.php?eid=193. *Прим. перев.*

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

Подтверждение

Финансирование функций RFC Editor обеспечено Internet Society.