Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 31, 2015

G. Wiley, Ed.
E. Osterweil, Ed.
D. Smith, Ed.
Verisign
A. Reiner, Ed.
D. Roark, Ed.
Armory Technologies

February 27, 2015

Using DANE to associate payment information with email addresses
draft-wiley-paymentassoc-00

Abstract

There is no standard, interoperable method for associating Internet service identifiers with payment information. This document specifies a means for retrieving information sufficient for a party to render payment using various payment networks given the recipient's email address by leveraging the DNS to securely publish payment information in a payment association record. A payment association record associates an Internet service identifier such as an email address with payment information such as an account number or Bitcoin address.

Status of This Memo

Copyright Notice

Table of Contents

```
   1.  Introduction  . . . . . . . . . . . . . . . . . . . . . . . .   3
     1.1.  Sending Money . . . . . . . . . . . . . . . . . . . . . .   3
     1.2.  Terminology . . . . . . . . . . . . . . . . . . . . . . .   4
   2.  The Payment Association Resource Record . . . . . . . . . . .   5
     2.1.  The PMTA RDATA wire format  . . . . . . . . . . . . . . .   5
       2.1.1.  Payment Network Selector Field  . . . . . . . . . . .   6
       2.1.2.  Preference Field  . . . . . . . . . . . . . . . . . .   6
       2.1.3.  URI Length Field  . . . . . . . . . . . . . . . . . .   7
       2.1.4.  Uniform Resource Identifier Field . . . . . . . . . .   7
       2.1.5.  Payment Association Data Type Field . . . . . . . . .   7
       2.1.6.  Payment Association Data Field  . . . . . . . . . . .   8
     2.2.  The PMTA RDATA presentation format  . . . . . . . . . . .   8
   3.  Locating the PMTA record  . . . . . . . . . . . . . . . . . .   9
     3.1.  Using Email addresses to Locate PMTA records  . . . . . .   9
   4.  Payment Association Data for ACH  . . . . . . . . . . . . . .  10
     4.1.  ACH Bank Routing Number . . . . . . . . . . . . . . . . .  10
     4.2.  ACH Account Number  . . . . . . . . . . . . . . . . . . .  10
     4.3.  ACH Receiving Name  . . . . . . . . . . . . . . . . . . .  11
     4.4.  Wire Format Payment Association Data Field for ACH  . . .  11
   5.  Payment Association Data for Bitcoin  . . . . . . . . . . . .  11
     5.1.  Bitcoin Addresses . . . . . . . . . . . . . . . . . . . .  11
     5.2.  Bitcoin Components  . . . . . . . . . . . . . . . . . . .  12
     5.3.  Wire Format Payment Association Data field for BTC/TBTC .  13
     5.4.  Constructed Script Components . . . . . . . . . . . . . .  13
   6.  Security Considerations . . . . . . . . . . . . . . . . . . .  14
     6.1.  Email address information leak  . . . . . . . . . . . . .  14
     6.2.  Inherent Risk . . . . . . . . . . . . . . . . . . . . . .  14
   7.  IANA Considerations . . . . . . . . . . . . . . . . . . . . .  14
     7.1.  PMTA RRtype . . . . . . . . . . . . . . . . . . . . . . .  14
     7.2.  Payment Network Selector Registry . . . . . . . . . . . .  15
     7.3.  Payment Association Data Type Registry  . . . . . . . . .  15
   8.  Acknowledgments . . . . . . . . . . . . . . . . . . . . . . .  15
   9.  References  . . . . . . . . . . . . . . . . . . . . . . . . .  15
     9.1.  Normative References  . . . . . . . . . . . . . . . . . .  15
     9.2.  Informative References  . . . . . . . . . . . . . . . . .  16
   Authors' Addresses  . . . . . . . . . . . . . . . . . . . . . . .  16
```

1. Introduction

In order to receive payment for goods or services a seller must provide account information or a payment address. Finance systems currently leverage a number of different approaches for providing information required to affect a transfer of money between parties to a transaction. The risk of cut and paste errors and vulnerability to MITM attacks is significant. How can a buyer be certain that they are sending money to the right person?

In the case of cryptocurrency, such as Bitcoin, transactions are not reversible; if Bitcoin is sent to the wrong address the funds are permanently lost. In the case of interbank transfers there may be some recourse if funds are sent to the wrong recipient, however recovering the funds can be difficult. It is important that a payer be confident in the association between the recipient and the payment information.

This association is most useful in cases where payments are made out of band from a transaction negotiation. For example when buying a product or service you could simply look up the payment information by leveraging an email address. Another example would be when negotiating a transaction over the phone - the seller could provide a mnemonic (such as an email address) that could be reliably retrieved via the DNS.

The payment association record ("PMTA") addresses these issues by providing a secure and reliable association between service identifiers and payment information via a new RR type: PMTA.

If multiple PMTA records are present in an RRSet then any one of them is valid (provided the preference field is not set to REJECT); which facilitates the use of multiple payment networks and allows the payer to select which network they would like to use.

This draft leverages two use cases as practical implementations of the payment association record to demonstrate the use of a payment association. A simple use case provides Automated Clearing House (ACH) information. A more complex use case demonstrates the use of a cryptocurrency (Bitcoin). Other payment networks can also use the PMTA record, which uses should be captured in separate drafts including changes to the IANA registries referenced in this draft.

1.1. Sending Money

In a retail setting there is infrastructure in place to provide a payer the means for sending money to a payee. In the simplest case the payer hands the payee cash. In a slightly more complicated case

the payer uses a credit card terminal to transfer money from a credit card to the payee. Transactions become less secure and reliable once we consider the use cases outside the retail setting, particularly in cases where there is a deferred payment. In the current financial model payers prefer to use either cash or to rely on familiar infrastructure to transfer money to a payee. In the absence of cash or familiar infrastructure (which varies from locale to locale) a payer will defer payment until they can send money via a wire transfer, mail a bank check or purchase a money order. This imposes significant friction on the transactions that occur outside retail or online settings.

The security of different payment mechanisms varies widely with respect to vulnerability to unauthorized transfers. In the case of a cryptocurrency like Bitcoin that vulnerability is very low (provided the payer uses their wallet properly). In many parts of the world the bank account and routing numbers are the most common means for affecting transfers.

This draft addresses use cases that satisfy two conditions: 1) they occur outside the typical retail setting that provides familiar payment infrastructure and 2) they rely on a source of payment that can be provided to a payer without exposing the payee to the risk of unauthorized transfers.

Although Bitcoin provides an almost ideal use case for payment associations some more traditional payment mechanisms are also good candidates for PMTA. Consider account numbers in cases where authentication mechanisms prevent unauthorized transfers from the account (such as shared secrets). It would be very useful for payees to make the account information available to payers in order to reduce the friction in transactions outside a typical retail setting.

Companies in the finance industry have spawned an impressive number of "safe" mechanisms for transferring money. Most of which share the same problem we have identified earlier: How do I know that I have the right account number to which to send money? The sheer volume of solutions is a strong indication of the need for making associations between payment information and service identifiers. The model used by Paypal offers probably one of the strongest cases that demonstrates the utility of being able to provide something like an email address as a key to payment information for payers.

1.2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

This document also makes use of standard DNSSEC and DANE terminology. See DNSSEC [RFC4033], [RFC4034], [RFC4035], and DANE [RFC6698] for these terms.

2. The Payment Association Resource Record

The Payment Association (PMTA) DNS resource record (RR) is used to associate payment information with an email address or other internet service identifier, thus forming a "payment association". The goal is to provide sufficient detail to allow for unambiguous and secure delivery of a payment to the requester who has published the PMTA record.

A value for the PMTA RR type will be assigned (temporarily use type code 65337 from the block of experimental RR type codes).

The PMTA RR is class independent.

2.1. The PMTA RDATA wire format

The RDATA for a PMTA RR consists of a series of header fields followed by the payment association data:

- Payment Network Selector field that indicates which payment network the payment information is relevant to. The value is defined in a new IANA registry in order to make it easier to add new payment networks.

- Preference field which indicates the relative preference for this PMTA record compared to other PMTA records in the RR set.

- URI Length field that indicates the length of the URI String field.

- URI String field that indicates an alternate service that will provide the payment information.

- Payment Association Data Type field which indicates the type of the data in payment association data field

- The Payment Association Data field is interpreted based on the Payment Network Selector and Payment Association Data Type fields

```
                        1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Payment Network Selector      | Preference                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | URI String Length             | URI String                    /
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               /
   /                                                               /
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Association Data Type         |    Payment Association Data   /
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               /
   /                                                               /
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

2.1.1. Payment Network Selector Field

The Payment Network Selector field indicates which payment network will be used for the transfer of funds. The value of the payment network selector affects the way the Payment Association Data field is interpreted. The value is specific to the payment network and is defined in a new IANA registry in order to make it easier to add new payment network types and usages. This document populates the registry with the following initial values:

0 or ACH, Automated Clearing House

1 or TBTC, Bitcoin blockchain test network

2 or BTC, Bitcoin blockchain network

2.1.2. Preference Field

The Preference field is a two-octet value interpreted as a 16-bit unsigned integer that indicates the relative preference for this PMTA record compared to other PMTA records in the RR set. A lower value in this field indicates a higher preference. The record with the lowest preference value that the payer can support should be used by the payer.

A value of 65535 (2^16-1) in the Preference field indicates that this PMTA RR MUST be considered invalid. This is a way of asserting that the payment information in a previously valid record is no longer valid.

2.1.3. URI Length Field

The URI Length field indicates the number of characters that follow that should be interpreted as a URI. If the URI Length is greater than 0 then the URI String field specifies an alternate means for retrieving the Payment Information.

If the URI Length field is 0 then the URI String field is not populated or used for this record and the Payment Association Data contains the payment information.

2.1.4. Uniform Resource Identifier Field

The Uniform Resource Identifier (URI) String field is a string encoded as hex digits that specifies an alternate means for retrieving payment information. It describes an alternative location for the payment information. Some domain owners may not want to publish payment information via the DNS but may want to use the DNS to advertise the means to access it.

The URI includes a scheme (such as "http", "https", "ftp", etc.) followed by a colon character and then a scheme-specific part as defined in [RFC3986].

The data returned by the service addressed by the URI is interpreted based on specifications provided by the service that responds to the URI. If the URI String field length is greater than 0 then the Payment Association Data field contains cryptographic material that is used to sign the data that is returned by the service addressed by the URI.

It is important to note that a URI may specify protocols or service locations that are not universally reachable to relying parties, and administrators should be conscious of this when deciding to store payment information off-axis.

2.1.5. Payment Association Data Type Field

The Payment Association Data Type field is a two-octet value interpreted as an unsigned integer that specifies how to use the Payment Association Data field. These values will be defined in a new IANA registry. This document populates the new registry with the following initial selector values:

    0 or ADDR, Payment Association is a static payment address

1 or SPKI, if the URI Length field is non-zero then the Payment Association Data contains subject public key info in a DER-encoded binary structure as defined in [RFC5280]

2 or CERT, if the URI Length field is non-zero then the Payment Association Data contains a full certificate as defined in [RFC5280]

2.1.6. Payment Association Data Field

The previously described fields indicate what the payment association field is used for. Although ACH and Bitcoin payment networks are used to demonstrate the PMTA record, other payment networks will require interpretation of the data in this field according to specifications written to describe the indicated payment network.

2.2. The PMTA RDATA presentation format

The RDATA Presentation Format, as visible in textual zone files is defined as follows:

o The Payment Network Selector must be represented as a payment network selector mnemonic or a 16-bit unsigned integer.

o Preference must be represented as a 16-bit unsigned integer.

o The URI Length field must be represented as a 16-bit unsigned integer.

o The URI String field representation will be determined by future work, the description in draft-faltrsom-uri appears to be a reasonable approach.

o The Payment Association Data Type field MUST be represented either as a Payment Association Data Type field mnemonic or an 16-bit unsigned integer.

o Payment association data MUST be represented as a string of hexadecimal characters. White space is allowed within the string of hexadecimal characters as described in [RFC1035].

Where practical the mnemonic form SHOULD be used in order to provide clarity.

3. Locating the PMTA record

One of the valuable use cases for payment association records in the DNS is the ability for a user to communicate a predictable and simple anchor to a payer without weakening security. Email addresses are a ubiquitous means for identifying users and make a logical choice as a way of providing payment associations.

While email addresses provide a very simple and predictable means for locating a payment association, there is also a need for more flexible mechanisms. Operators may choose to provision a PMTA record with any label in a zone. In these cases the operator simply offers a valid DNS name to payers from which they can retrieve the PMTA records.

3.1. Using Email addresses to Locate PMTA records

Email addresses are mapped into DNS using the following method:

1. The user name (the "left-hand side" of the email address, called the "local-part" in the mail message format definition [RFC2822] and the "local part" in the specification for internationalized email [RFC6530]), is hashed using the SHA2-224 [RFC5754] algorithm represented as hex to become the left-most label in the prepared domain name. This does not include the at symbol ("@") that separates the left and right sides of the email address.

2. The DNS does not allow the use of all characters that are supported in "local-part" of email addresses as defined in [RFC2822] and [RFC6530] . The SHA2-224 hashing of the user name ensures that none of these characters would need to be placed directly in the DNS.

3. The string "_pmta" becomes the second left-most label in the prepared domain name.

4. The domain name (the "right-hand side" of the email address, called the "domain" in RFC 2822) is appended to the result of step 2 to complete the prepared domain name.

For example, to request a PMTA resource record for a user whose email address is "bob@example.com", a PMTA query would be placed for the following QNAME: "550c233eeabd0f03bb42b99956efa56cdadaef7d346a04e351a c1b7a._pmta.example.com" The corresponding RR in the example.com zone might look like (hash shortened for formatting):

```
55[..]7a._pmta.example.com.  IN PMTA <payment association>
```

4. Payment Association Data for ACH

Automated Clearing House (ACH) is ubiquitous as a means for transferring money between parties. In the United States bank checks are routinely used to make payments when a physical exchange is made between parties. The payer writes a check drawn on their bank, this check includes the ABA routing number (in some cases this is the same as the ACH routing number), account number and authorization to withdraw funds using a hand written signature to authenticate the payer.

In many other parts of the world the bank routing number and account number are used to affect electronic transfers between parties without the use of hand signed checks. In these cases the payer initiates the transfer once the recipient of the funds provides their bank routing number and account number.

A payer needs very little information to affect a transfer of funds to the recipient, namely:

- ACH Bank Routing Number

- ACH Account Number

- ACH Recipient Name

Although there are some additional requirements for an ACH transfer, these can be populated by the payer and do not need to be identified in the payment information.

4.1. ACH Bank Routing Number

A nine digit number (the ACH Bank Routing Number) encoded as a string of nine octets containing ASCII digits that uniquely identifies the recipient's bank.

4.2. ACH Account Number

The recipient's account number can be up to thirty five octets long and uniquely identifies the recipient's account at the bank indicated by the routing number. The account number is encoded as a left justified string of ASCII digits, the unused positions are filled with the null character.

4.3. ACH Receiving Name

The receiving company or individual name must be provided in an ACH transaction as it will appear in the ACH transfer is up to thirty five characters long. The receiving name is left justified and encoded as hexadecimal characters in 70 octets with the unused positions set to 0.

4.4. Wire Format Payment Association Data Field for ACH

If the Payment Network Selector is ACH then the Payment Association Data Type Field is ADDR and the Payment Association Data field contains the static payment information sufficient to send money to the recipient.

```
                        1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Routing        |  Account                                    /
   +-+-+-+-+-+-+-+-+-+     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-/
   /                       |  Name                                 /
   +-+-+-+-+-+-+-+-+-+-+-+-+                                       /
   /                                                               /
   /                                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-/
   /                                   | unused                    /
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5. Payment Association Data for Bitcoin

The Payment Association Data Type field shall have the value ADDR for the BTC and TBTC payment networks. Some of the implementation details are left to future work.

5.1. Bitcoin Addresses

The technical side of Bitcoin causes addresses to have ambiguous meanings. Therefore, the word "address" is not used in this specification without giving a specific definition for the address; an address indicates particular use types that are strictly defined.

A Bitcoin address provided to receive payment is simply a chunk of data to be used in a particular script template type. Such templates are known as TxOut script templates. A TxOut script specifies who exactly will receive payment. As of Jan. 2015, there are five standard TxOut types on the Bitcoin network. The TxOut types are not of specific interest except to help explain what this specification is attempting to accomplish.

For example, as of Jan. 2015, the most common TxOut type on the Bitcoin network is known as the pay-to-public-key-hash (P2PKH) template. P2PKH uses addresses starting with 1. In P2PKH, the payer must insert a 20 byte payload, based on a secure hash of a public key, into the accompanying P2PKH script template. The P2PKH template is defined as follows:

```
    OP_DUP OP_HASH160 <20 byte payload> OP_EQUALVERIFY OP_CHECKSIG
```

Another address example is an address starting with 3. For such addresses, the payer must insert the inner 20 byte payload into what is known as a pay-to-script-hash (P2SH) template, as defined in BIP 16 [BIP16]. The P2SH template is defined as follows:

```
    OP_HASH160 <20 byte payload> OP_EQUAL
```

This specification focuses on that fact that the Bitcoin wallet software, at the time of payment, is ultimately constructing a destination, or destinations, for the coins. This specification defines a way for Bitcoin wallet software to communicate both the TxOut script to be paid, as well as a secure authentication chain for verifying that a TxOut script is associated with a given recipient.

5.2. Bitcoin Components

Public Key Source (PKS): Communicates the simplest of ID information, typically about a single BIP32 key tree that would be used as a single signature wallet to manage funds. A PKS can also be used as a placeholder in a more complex, constructed script. A public key source can also be a reference to another resource for getting the public key source.

Constructed Script (CS): This is a data structure which includes a script template and a list of public key sources.

Raw Payment Script (RPS): This is the final, complete script expected to receive payment as part of a given payment association. The payer may ignore any attached proofs and simply use this script to pay. This will usually be provided with a PKRP and/or SRP (see below). However, if a PKS is a stealth address, requires a user-supplied key, or uses an external source, an RPS cannot be included. There may be other conditions when an RPS cannot be provided. Encoding and Implementation details are left to future work.

Public Key Relationship Proof (PKRP): This is a list of multipliers to apply to a given PKS. If the PKS specifies a root-level public key, but the payment script uses a 3rd-level public key, then the

proof will actually be three 32-byte multipliers. Encoding and Implementation details are left to future work.

Script Relationship Proof (SRP): This is a list of PKRPs to be applied to a Constructed Script, one per public key placeholder in the script template. Encoding and Implementation details are left to future work.

5.3. Wire Format Payment Association Data field for BTC/TBTC

In the example cases using the simplest bitcoin addresses the wire format for the Payment Association Data field includes a few fields.

The Script Length field is encoded as a 16-bit unsigned integer (two-octets) and specifies the length in octets of the Script/Address field.

The Script/Address field is encoded as a

```
                     1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Script Length                | Script/Address                /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               /
/                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5.4. Constructed Script Components

Some use cases will leverage payment information more complex than a static payment address. If the selector field specifies a constructed script then the payment association data is further described as:

```
                     1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Tx Method    |  Script Template                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                 PKS entries                                   /
/                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Tx Method is a one-octet value that specified the transaction method. Values include 0 or 1 to indicate whether a P2SH is to be used.

A Script template that contains specifically designed opcodes that use public key data to construct the final TxOut script.

PKS entries is a list of PKS entries used with the script template to construct the final script.

6. Security Considerations

PMTA usage considerations are published in a separate usage document.

6.1. Email address information leak

Using email addresses as a key to fetch payment information by including them in the DNS provides an undesirable opportunity for harvesting email addresses for attackers willing to "walk" the zone. While using NSEC3 increases the difficulty of harvesting email addresses by "walking" a zone it does not prevent this approach.

NSEC5 might be a more viable option than NSEC3 for preventing unintended leaks via the DNS however at the time of this draft it is still being discussed as a proposal. Note that this problem is shared by other proposed record types that create associations similar to those used by PMTA (SMIMEA, OPENPGPKEY).

6.2. Inherent Risk

The use of the PMTA RR type should be restricted to zones that are properly signed (DNSSEC). The use of the DNS for delivering payment information underscores the importance of keeping the keys used for signing the zone secure.

7. IANA Considerations

7.1. PMTA RRtype

This document defines a new DNS RR type, PMTA, whose value will be allocated by IANA from the Resource Record (RR) TYPEs subregistry of the Domain Name System (DNS) Parameters registry.

IANA is requested to create two other registries:

o Payment Network Selector

o Payment Association Data Type

7.2. Payment Network Selector Registry

This document creates a new registry, the "Payment Network Selector". The registry policy is "RFC required" and the initial entries in the registry are:

Value	Short Description	Mnemonic	Reference
0	Automated Clearing House	ACH	
1	Bitcoin blockchain test network	TBTC	
2	Bitcoin blockchain	BTC	

7.3. Payment Association Data Type Registry

This document creates a new registry, the "Payment Association Data Type". The registry policy is "RFC required" and the initial entries in the registry are:

Value	Short Description	Mnemonic	Reference
0	Simple Address	ADDR	
1	Subject Public Key for URI	SPKI	
2	Full certificate for URI	CERT	

8. Acknowledgments

Burt Kaliski provided significant direction and useful feedback to this document. Additional input from Scott Hollenbeck, Swapneel Sheth and Lynch Davis contributed to this document. Some text was taken from an early draft of the DANE SMIME proposal by Scott Rose and the DANE openpgp draft by Paul Wouters as well.

9. References

9.1. Normative References

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

[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 the DNS Security Extensions", RFC 4034, 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.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

[RFC5754] Turner, S., "Using SHA2 Algorithms with Cryptographic Message Syntax", RFC 5754, January 2010.

9.2. Informative References

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

[RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

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

[RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, February 2012.

[RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, June 2012.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

Authors' Addresses

Glen Wiley (editor)
Verisign

Email: gwiley@verisign.com

Eric Osterweil (editor)
Verisign

Email: eosterweil@verisign.com

David Smith (editor)
Verisign

Email: dsmith@verisign.com

Alan Reiner (editor)
Armory Technologies

Email: alan@bitcoinarmory.com

Douglas Roark (editor)
Armory Technologies

Email: doug@bitcoinarmory.com