Application Area Working Group	W. Chuang
Internet-Draft	N. Lidzborkski
Expires: April 18, 2014	E. Bursztein
	Google, Inc.
	October 15, 2013

MSMD: Mandatory Secure Mail Delivery

draft-wchuang-msmd-00

Abstract

Opportunistic SMTP TLS does not enforce electronic mail delivery using TLS leading to potential loss of privacy and security. We propose an optional mail header extension "mandatory-secure-mail-delivery:" and SMTP EHLO response extension "MSMD" that indicates mail must be delivered privately using TLS and with integrity using DKIM, and thereby provide a security guarantee to the user. When mail is sent with the header indicating privacy and integrity and if the receiving party does not support this, the mail is instead bounced. To protect the mail after delivery, the destination SMTP server must advertise its capabilities as part of the EHLO response, and the sender can choose whether the destination is able to honor the privacy requirements specified on the mail header.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction
2. Mandatory Private Delivery Specification
2.1. MSMD Mail Header
2.2. MSMD SMTP Extension
2.3. Domain Keys Identified Mail
2.4. Deployment Concerns
2.5. Delivery Failure Notification
2.6. Mail User Agent Support
2.6.1. IMAP Extension
2.6.2. POP3 Extension
2.7. Compatibility
3. TLS
4. Optional Requirements and Capabilities
4.1. Mail Header Requirements
4.2. SMTP MSMD Capabilities
4.3. Example with Requirements and Capabilities
5. Recommendations
6. Security Considerations
7. Acknowledgements
8. References

Authors' Addresses

1. Introduction

Opportunistic SMTP TLS [RFC3207] does not enforce electronic mail delivery using TLS. This means that even if a user wishing to send mail has a provider that supports SMTP TLS, the provider does not necessarily deliver mail over TLS depending on whether the peer supports TLS or if the TLS negotiation fails. Unfortunately our observation is that most mail providers do not support SMTP TLS. These deployment problems are made worse because users typically do not have control over delivery and do not know whether their message will be delivered securely. Also users have an assumption that their mail content will be delivered without modifications though in fact with traditional SMTP [RFC5321] there is no assurance of this. This combination diminishes trust amongst electronic mail users.

We believe that finding a solution will become more important as recent news indicates that eavesdroppers will increasingly targeting electronic mail delivery as a likely point of attack (if not already). This is motivated by the observation that web client side security has steadily improved due to increasingly effective and deployed encryption while most mail delivery remains unencrypted. Moreover SMTP TLS is more susceptible to active attack such as Man-in-the-Middle (MitM) than web client security. Attacks such as certificate forgery or TLS downgrade have been mitigated with certificate pinning and HSTS [RFC6797]. No such mechanism exists today with SMTP TLS.

The main alternative is to use encrypted mail such as PGP/MIME and S/MIME which keep the mail body private during delivery. However they do nothing to guarantee private mail delivery, and traditional SMTP transmits the sender and the recipient in the clear. Thus even PGP/MIME [RFC3156] or S/MIME [RFC5751] encrypted mail is susceptible to meta-data surveillance.

We propose optional mail extensions that indicates that mail delivery must be delivered privately using TLS, and must be signed with DKIM to indicate authenticity and integrity. The user or their provider can choose to set a new mail header "mandatory-secure-mail-delivery:" that indicates private mail distribution as described in Section 2.1. When mail is sent with this header, the SMTP client (sending SMTP server) checks if the receiving side supports encrypted and signed private delivery. If not, the mail is bounced. The bounce message must be returned to sender securely as well, otherwise the message must be dropped. A courtesy message may be sent to the recipient depending on the sender's settings. Another concern is maintaining the security of the mail at the destination and beyond. The SMTP server (destination SMTP server) must advertise it supports and enforces this protocol as part of the EHLO response using the extension "MSMD" which the sender will verify as described in Section 2.2. Mail providers that honor this protocol should ensure that

derivative mails created such as when forwarding and replying maintain the requested private mail delivery property by copying the security header to the new message. This is not mandatory in the base protocol because honoring it would impose restrictions for all Mail User Agent interfaces that would be a substantial barrier to entry. In the following sections, we show how we can elevate the "should" to "must" for securing derivative mails. Thus with the base protocol at minimum a sender can be certain that a sent mail will be delivered privately and unaltered.

The protocol can impose additional security requirement options via arguments on the message-header as described in Section 4.1 that the SMTP client can verify. As part of this verification process, the SMTP server can also advertise domain security capability options as part of the EHLO response as described in Section 4.2. They can help insure the recipient's mail provider is just as capable of verifying the security requirements as the sender's provider. Options enable the protocol to flexibly extend the definition of security to meet different needs, and to evolve as the security standing of the ecosystem improves.

Options are particularly useful if the sender wishes to mandate enforcement of the protocol for the entire conversation meaning through all derived forwarded and replied mails. We propose a Secure Conversation and Access (SCA) option that mandates for all mail messages so marked, all derived mail must also be so marked. Any mail agent in the domain must honor and propagate the additional SCA protocol in addition to the base protocol. This imposes a powerful and stringent requirement over all MUA such as IMAP or POP3 to support the protocol as described in Section 2.6, or not be able to access these messages. SCA also requires that these clients must access these messages privately over an encrypted channel. Thus a sender can be sure that a sent mail and its derivatives will be sent and accessed securely.

Via options we can also specify TLS settings to improve security. Of the threats to TLS, perhaps the most difficult to protect against is MitM since the adversary impersonates the endpoint. To defend against this, we bundle various TLS settings into tiers corresponding to the capabilities of ecosystem that improves the ability to reject MitM peers.

This example illustrates the message declaration, and that of the recipient i.e. the base protocol. It also illustrates the verification by the sender. We use the convention "C:" for client and "S:" for server.

```
Some mail message has a private delivery request, which specifies the
mail header:
mandatory-secure-mail-delivery:

The SMTP exchange protocol appears as follows:
S: <waits for connection on TCP port 25>
C: <opens connection>
S: 220 mail.server.org SMTP service ready
C: EHLO mail.client.com
S: 250-mail.server.org welcomes you
S: 250-STARTTLS
S: 250-MSMD
S: 250 DSN
C: STARTTLS
S: 220 Go ahead
C: <starts TLS negotiation using default TLS requirements>
C & S: <negotiate a TLS session>
C: EHLO mail.client.com
S: 250-mail.server.org welcomes you
S: 250-MSMD
S: 250 DSN
C: <Perform MSMD Requirements Check: On failure close the connection>
C: DATA
C: <Send mail with DKIM signature>

Note: 1) TLS requirements are passed to the TLS setup 2) after the
second EHLO, that there is a MSMD Requirements Check. The latter
verifies that the TLS connection has started, and that the MSMD
SMTP extension was presented on the second EHLO.
```

Figure 1: Basic MSMD Protocol Example

2. Mandatory Private Delivery Specification

2.1. MSMD Mail Header

We propose a new optional mail header: [RFC5322] "mandatory-secure-mail-delivery:" (MSMD) in accordance with [RFC3864]. The MSMD header indicates that this message must be privately distributed using encryption. It may have associated with it parameters that impose additional requirements on the delivery security and destination security. The parameters will be discussed further in Section 4.1. Requirements may only be applied if the user's mail provider can support those requirements.

Mail messages derived from a mail message with the MSMD message header should copy this header along with any options, and those derived messages must honor the security protocol. Examples of deriving the message are replying and forwarding the mail message. If the Secure Conversation and Access option is specified, the derived message must copy the message header. The user is free to download the content, or distribute the contents via GUI means such as cut or copy and paste without the header. The distinction is meant to protect mail content against accidental exposure via unencrypted mail delivery, but not stymie legitimate every day use of mail content. Derived mail may add additional security requirements to the MSMD message header but must not remove any requirements.

2.2. MSMD SMTP Extension

When delivering the mail message, the SMTP client must verify that the destination will honor maintaining the MSMD protocol once the message is delivered. To support this, the SMTP server advertises that it supports the protocol via a MSMD SMTP extension [RFC5321]. This means that during the EHLO response, there will be a "MSMD" keyword, and that it may be followed by parameters describing additional capabilities of the server. The capabilities are described in Section 4.2. The SMTP client verifies that the SMTP server has both MSMD and STARTTLS extensions, then establishes the TLS connection. As the connection at this point is encrypted and private, it redoes the EHLO, gathering any capabilities specified by the server. Then it does the MSMD requirements check using the securely communicated capabilities values. In the basic configuration meaning without any additional requirements or capabilities, SMTP client just verifies the connection is in TLS, and re-verifies that there is a MSMD SMTP extension. Success allows mail delivery to continue, while failure causes a notification to be sent as described Section 2.5 and the TLS and SMTP connection is closed and reset. Any other Mail Transfer Agent must support this protocol in a similar fashion or be prevent from transferring mails marked by the MSMD header.

2.3. Domain Keys Identified Mail

All MSMD market mails must be sent with DKIM [RFC6376] signatures to provide assurance for the integrity and authenticity of the mail message. Similarly MSMD SMTP Servers must verify received DKIM messages. In the MSMD context, how DKIM handles failed signature verification is up to the mail provider. MSMD mandates that at least the message body and "mandatory-secure-mail-delivery:" header be signed leaving the rest as implementation dependent.

2.4. Deployment Concerns

Though much of this proposal is concerned with security of SMTP mail delivery, the base protocol recommends that all other means of accessing the protected mail via Mail User Agents should be similarly protected. This means all mail fetch mechanism should similarly honor the encrypted delivery requirement and should propagate the MSMD header to derived mails.

If the Secure Conversation and Access option is given, the restrictions on MUA become mandatory. Also to prevent eavesdropping during fetch, SCA places a further requirement to encrypt access. If the MUA cannot insure propagating the MSMD protocol or provide private access, that they must be prevented from accessing user data marked with the MSMD security header. Upon access failure these clients should use whatever error reporting mechanism built into the protocol to provide a notification indicating the cause and an alternative means of accessing the mail securely.

Mailing list forwarders supporting the MSMD protocol must also check if each list recipient honors the protocol in the same way as a single recipient. Failure to meet the security requirements results in the message not being delivered and may result in notification of delivery failure. These forwarders must also insure that DKIM verification will succeed either by maintaining the exact same MSMD header and message body, or by re-generating the DKIM signature.

2.5. Delivery Failure Notification

For message delivery failure due to insufficient security, the sender will be notified via a non-delivery notification message [RFC3461]. For diagnostics, these security bounce messages should include at least a description of the failure including which step of the SMTP handshake that failed, any missing requested options, and identifying information such as destination and subject. Security bounce messages must be delivered with the same security guarantees as the originating message. This means that the MSMD header is copied to the bounce message, and honored as with the original, and a DKIM signature placed on the message. To prevent an endless notification loop, delivery failure of these security bounce messages due to MSMD protocol failure causes the message to be silently dropped.

Upon security failure on delivery, the SMTP client may still sent a courtesy message to the recipient depending on the header options, to let them know that mail is missing due to security failure at delivery time. The courtesy message is optional in case the sender insists on keeping private all meta-data that might be leaked by courtesy messages. It differs from security bounce messages in that the security header is not propagated, and the information passed is more restricted. The courtesy message should just provide enough to identify the message and cause but no more i.e. identifying information such as from: and subject: but not the body or any additional user identifying header information.

2.6. Mail User Agent Support

Assuming that the Secure Conversation Mail option applies to the mail message, Mail User Agents such as IMAP or POP3 Clients must honor the MSMD security protocol, and clients must be able to prove to the servers they are capable of honoring the protocol. Such MUA servers must be able to display the MSMD extension, and the MUA clients must be able present to the server its ability to honor the MSMD protocol. If MUA clients does not display support for MSMD, then MSMD MUA servers must prevent access to MSMD marked and protected mail.

2.6.1. IMAP Extension

In the case of IMAP [RFC3501], it provides a CAPABILITY command to describe the server extensions. We propose a new MSMD capability to IMAP where "MSMD" appears in the CAPABILITY response, and a new "MSMD" command that announces to the server that the client supports the MSMD protocol.

2.6.2. POP3 Extension

Like IMAP, POP3 [RFC1939], it provides a CAPA [RFC2449] command to describe the server extensions. We propose a new MSMD capability where "MSMD" appears in the CAPA response, and a new "MSMD" command so that the client can indicate to the server it supports the MSMD protocol.

2.7. Compatibility

Specification of the MSMD security header is on a per message basis. This allows the sender to fallback to clear-text delivery for backwards compatibility for recipients that don't support the protocol and don't need the security it mandates. In the expected case, when the sender starts composing the mail, they can check the GUI to see if the recipient can support MSMD, and if not then elect to send mail insecurely or not at all. That said, we expect that over time as the ecosystem will upgrade its security capability meaning that MSMD support becomes the norm. Once that happens, it is easy to imagine that a mail provider would make MSMD mandatory.

The Secure Conversation and Access option also provides a significant increase in privacy by mandating restricted access after delivery. This poses a policy issue, as the recipient has mandatory restriction placed on mail in their inbox that they may not want. Thus we imagine that this option be used only in conversations where privacy is required such as by business need, and where all parties would not object to the restrictions. With other conversations the sender could use the base MSMD protocol or even clear text delivery. Also these restrictions are caused by infrastructure limitations that should improve over time. Once MSMD and SCA support becomes universal then this restriction become moot.

3. TLS

TLS can be configured to significantly degrade eavesdropping and MitM. They are organized into tiers corresponding to ability of the ecosystem to support the features, and are meant to evolve and be upgraded to draw upon improved techniques for stopping these threats. The following list are the current TLS parameters that MSMD controls.

TLS Version
: Describes the TLS version. Currently this is restricted TLS only and version "1.0" (SSL version 3,1) or greater must be supported.

CipherSuite
: Describes the TLS cipher as a selector based on the values from IANA registry for TLS [RFC5246]. Anonymous Diffie-Hellman key exchange and RC4 cipher shall not be allowed.

Public Key Size
: Describes the SMTP client/server minimum TLS certificate CA certificate public key size. MSMD requires that key sizes of 1024, 2048, 4096 be supported for RSA, DSA and Diffie-Hellman based algorithms, and 250 for Elliptic-Curve Cryptography if supported. It is recommended that larger sizes be supported as well.

Public Key Type
: Describes the SMTP server TLS [RFC5246] public key type. MSMD requires that RSA be supported, and others are optional.

Symmetric Key Size
: Describes the SMTP server TLS [RFC5246] symmetric cipher key size. Both 128 and 256 bits must be supported.

Check Server Certificate
: This specifies the mechanism by which the SMTP TLS server certificate are verified. Ability to use PKI verification of X.509 certificate [RFC5280] using "CA" (Certificate Authority) must be supported, though actual verification is recommended. We recommend that certificate naming standardize [RFC6125] to setting the DNS-ID from the MX record host name. We also recommend using improved PKI such as Certificate Transparency [RFC6962].

To reiterate, the baseline TLS constraints for all MSMD implementation with or without specifying the "tls" requirement option is the following:

- TLS version >= "1.0"
- MSMD Public key exchange must support RSA.
- Public key exchange shall not use anonymous Diffie-Hellman.
- Cipher shall not use RC4.
- RSA/DSA/DH Public Key Size >= 1024 or ECC Public Key Size >= 250
- Symmetric Key Size >= 128
- Support for CA PKI verification of X.509 certificate.

If the "tls" requirement option is specified, then it must specify one of two tiers. The tiers corresponding to what can be deployed today (circa 2013) with reasonable support from ecosystem.

- Tier 1- Baseline strong TLS. Assumes TLS baseline.
    - TLS version >= "1.2"
    - Public key exchange must use ephemeral Diffie-Hellman for perfect forward secrecy.
    - RSA/DSA/DH Public Key Size >= 2048 or ECC Public Key Size >= 250

- Tier 2- Resist MitM- Assumes capabilities of tier 1.
    - Check Server Certificate- should check for: path, expiry, and trusted CA root. Self-signed certificates shall not not verify.
    - DNSSEC- DNS name look up must look up and verify security extensions. In other words the "ds" DNSSEC requirement option is implicitly set, though it is recommended the "ds" requirement should still be explicitly appended for readability.
    - Require that certificate naming standardize [RFC6125] to setting the DNS-ID from the MX record host name.

We anticipate more tiers being defined particularly to improve PKI.

```
    MX Record for gmail.com:
    gmail.com.  3600 IN MX 10 alt1.gmail-smtp-in.l.google.com.
    gmail.com.  3600 IN MX 5 gmail-smtp-in.l.google.com.

    SMTP X.509 Certificate recommended identifier bindings:
```

```
DNS-ID: gmail-smtp-in.l.google.com
DNS-ID: alt1.gmail-smtp-in.l.google.com.
```

Figure 2: Certificate Naming for SMTP

Certificate naming of identifiers for SMTP servers has been recently rigorously defined in [RFC6125]. MSMD recommends (and at Tier 2 requires) the use of this naming scheme for the certificate identifiers. The certificate DNS-ID identifier contents should be the SMTP MX host name, and if there are more than one host name than multiple DNS-ID names may be specified. The certificate common identifier CN-ID may also be set but with restrictions as noted in [RFC6125]. There are also restrictions for identifier wild-cards '*'. An example of the naming scheme is in Figure 2.

4. Optional Requirements and Capabilities

To allow for different security requirements and to allow for future extension, this protocol supports options. The MSMD message header may take additional arguments that act as requirements during the MSMD requirements check. The requirements specify expressions that have values bound to it during the TLS setup and the expression must evaluate to true. Requirements may be interpreted by other mail components outside of mail delivery e.g. SCA. The MSMD SMTP EHLO extension provides the means to display additional capabilities that the server can provide, and can bind values used for the requirements check. To summarize, options serve the following purposes:

- Constraints on TLS
- Constraints on recipient- Ensure the recipient mail provider can maintain security when the message is forwarded or replied-to from that provider.
- Settings describing the treatment of the message outside of the mail delivery

4.1. Mail Header Requirements

To simplify the treatment of MSMD requirements, they are all treated as expressions with the following rules for the evaluation. Requirement names are variables that may be bound to values from SMTP extension capabilities or bound to a default value as specified. Three value types are currently allowed: boolean, integer or string. They initially are bound to respectively False, 0, or "". All may also be set to "undef" meaning not assigned which is treated as False. Expressions support basic comparisons i.e. "==", "!=", "<", ">", "<=", and ">=". They also support composition with "AND", and "OR" and precedence group "()", as well as negation "NOT". Operator order of precedence follows the Python language. Requirement names and operators are case insensitive. One syntactic sugar is allowed:

1. If multiple requirements are presented in a sequence separated only by white-space, they are treated as composed with "AND".

The following list describes requirements that may be placed with "mandatory-secure-mail-delivery:" header. Each requirement has a short name and a longer description name. The short name should be used in the mail header to save transmission resources.

sca
: Secure Conversation and Access- as described in Section 2.4. (type: boolean)

ds
: DNSSEC- Use DNSSEC for name service look-up. Look-up must verify the signatures of DNS records. (type: boolean)

m
: Message- After MSMD security check failure, forward to the destination a courtesy notice as specified in Section 2.5. (type: boolean)

v
: Version- The version value for MSMD. This starts at 0. (type: int)

tls
: TLS Tier- As describes in Section 3 one can specify bundles or tiers of TLS settings. The allowed tiers are 1, and 2. (type: int)

4.2. SMTP MSMD Capabilities

As part of the EHLO response, additional optional capabilities are advertised. The capabilities names are variables bound effectively as key-value pairs. These results become available in the MSMD requirements check evaluation. Unadorned names are bound to "true", however if the name is followed by "=" and a value, then the name is bound to that value. Capabilities are also case insensitive.

sca
: Secure Conversation and Access- SMTP Server supports SCA.

ds
: DNSSEC- SMTP Server uses DNSSEC for name service look-up.

m
: Message- MSMD security check failure sends a courtesy notice.

v
: Version- The version number for MSMD.

tls
: TLS Tier- The version number for TLS tiers

4.3. Example with Requirements and Capabilities

The following illustrates how the options would be used. A company want mail delivery to their customers that mitigates eavesdropping and MitM attacks. To do so, this company would like to specify the use of perfect forward secrecy, DNSSEC, and X.509 CA PKI certificates verification during mail delivery. Currently the latter two requirements are not yet deployed for many mail providers, but will likely be available in the near future. So today this company just uses MSMD with baseline TLS encryption settings for mail delivery as described in the earlier example in Figure 1. Later when supported, the company specifies those three properties for new mail messages using the MSMD requirement option "tls>=2". This is illustrated in the following example in Figure 3

```
The mail message requests perfect forward secrecy, DNSSEC, and
X.509 certificate verification by setting the header to:
mandatory-secure-mail-delivery: tls>=2 ds

S: <waits for connection on TCP port 25>
C: <opens connection>
S: 220 mail.server.org SMTP service ready
C: EHLO mail.client.com
S: 250-mail.server.org welcomes you
S: 250-STARTTLS
S: 250-MSMD sca m tls=2 ds
S: 250 DSN
C: STARTTLS
S: 220 Go ahead
C: <starts TLS negotiation>
C & S: <negotiate a TLS session w/ECDHE-RSA-AES128-SHA>
C: EHLO mail.client.com
S: 250-mail.server.org welcomes you
S: 250-MSMD sca m tls=2 ds
S: 250 DSN
C: <Perform MSMD Requirements Check: On failure close the connection>
C: DATA
C: <Send mail with DKIM signature>

During the MSMD Requirements check the requirements expands as follows:
tls>=2 AND ds
2>=2 AND true

This evaluates true and passes, as does the baseline MSMD
requirements check.  The passing MSMD requirements check indicates
the recipient provider can maintain at least the same security as
requested for this message.
```

Figure 3: MSMD Options Example

5. Recommendations

Some recommendations that assist the effectiveness of this protocol:

- During composition of the mail message that the user be able to control setting of MSMD header and SCA option though not necessarily the other options.
- GUI should assist the user in predicting whether delivery will succeed when the MSMD feature is selected using provider modeling and other means.

6. Security Considerations

The ability of the protocol to maintain security of mail content depends on the effectiveness of peer mail systems implementation of this protocol. Since enforcement is voluntary and easily subverted by bad implementations, we propose a blacklist list of mail providers that advertise support but are known to have flawed implementations. This list would be maintained and adjudicated by a yet to be decided third party. Implementations should obtain the blacklist, and prevent delivery of MSMD messages to these mail providers.

7. Acknowledgements

The authors wish to acknowledge the very useful comments and suggestions from: Brandon Long, Adam Langley, Danesh Irani, Ian Fette, and Robert Chien.

8. References

[RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.

[RFC2449] Gellens, R., Newman, C. and L. Lundblade, "POP3 Extension Mechanism", RFC 2449, November 1998.

[RFC2971] Showalter, T., "IMAP4 ID extension", RFC 2971, October 2000.

[RFC3156] Elkins, M., Del Torto, D., Levien, R. and T. Roessler, "MIME Security with OpenPGP", RFC 3156, August 2001.

[RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.

[RFC3461] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)", RFC 3461, January 2003.

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[RFC3864] Klyne, G., Nottingham, M. and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R. and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.

[RFC5322] Resnick, P., "Internet Message Format", RFC 5322, October 2008.

[RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011.

[RFC6376] Crocker, D., Hansen, T. and M. Kucherawy, "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, September 2011.

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

[RFC6797] Hodges, J., Jackson, C. and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, November 2012.

[RFC6962] Laurie, B., Langley, A. and E. Kasper, "Certificate Transparency", RFC 6962, June 2013.

Authors' Addresses

Weihaw Chuang

Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
US
EMail: weihaw@google.com

Nicolas Lidzborkski
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
US
EMail: nlidz@google.com

Elie Bursztein
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
US
EMail: elieb@google.com