DANE Trusted Email For Supply Chain Management

Joseph Gersch
Secure64 Software Corporation
joe.gersch@secure64.com

Dan Massey
Colorado State University
massey@cs.colostate.edu

Scott Rose
NIST
scott.rose@nist.gov

Abstract

Supply chain management is critically dependent on trusted email mechanisms that address forgery, confidentiality, and sender authenticity. The IETF protocol 'Domain Authentication of Named Entities' (DANE) described in this paper has been extended from its initial goal of providing TLS web site validation to also offer a foundation for globally scalable and interoperable email security. Widespread deployment of DANE will require more than raw technology standards, however. Workflow automation mechanisms will need to emerge in order to simplify the publishing and retrieval of cryptographic credentials that are applicable for general audiences. Security policy enforcement will also need to be addressed. This paper gives a descriptive tutorial of trusted email technologies, shows how DANE solves key distribution logistics, and then suggests desirable automation components that could accelerate deployment of DANE-based trusted email. Pilot deployments are briefly described.

1. Introduction

Email is one of the most critical communication tools used in supply chain management. It is relied upon for a wide range of messages: partner-to-partner, customer-to-vendor, order processing and billing, and everyday intra- and inter-company communications. The inconvenient truth, however, is that email as typically used today *cannot* be relied upon.

It is difficult to tell if an email is fraudulent. An original email message can be modified by a man-in-the-middle attack; for example, to alter a bank routing number used for electronic payments. Phishing and spear phishing attacks are common and have become extremely sophisticated. Attackers are able to manipulate organizations for financial gain, espionage, or to launch malware.

Email is the preferred channel for launching targeted cyber attacks. Email is the weak link in government and enterprise security; it is hard to protect because email is not secure and is subject to social engineering. There are numerous examples of the abuse of email. A sampling of reports sorted from 2011 to 2016 shows a growing trend to targeted spear phishing:

- The 2011 *OMB Report to Congress* cites US CERT (The United States Computer Emergency Readiness Team) reporting 51.2% of 107,655 incidents reported by public agencies were phishing [1, 2].
- The Cisco[1] 2011 Security White Paper *Email Attacks: This Time It's Personal* illustrates the economic gain for attackers in moving away from mass attack phishing to targeted spear phishing attacks. In just one year the cyber criminal monetary benefit rose from $50 million to $150 million [3].
- Trend Labs 2012 Research Paper *Spear-Phishing Email: Most Favored APT Attack Bait* indicates that 65% of incidents were targeted to Government [4].
- The 2016 Verizon *Data Breach Investigation Report* states that 30% of phishing messages were opened by targets and 12% went on to click malicious attachments. The majority of phishing cases are used as a means to install persistent malware. Cyber-Espionage was found in 68 examples of phishing/social engineering attacks [5].
- As a specific example, Arrow Electronics, a major distributor, revealed that they were the victims of a $13 million theft in early 2016 based on a combination of social engineering and spear

[1] Certain commercial equipment, instruments, or materials (or suppliers, or software,...) are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

URI: http://hdl.handle.net/10125/41506
ISBN: 978-0-9981331-0-2

phishing in which an executive was impersonated [6].

Various approaches have been used to mitigate these problems. Application firewalls, Bayesian spam filters, email gateways and portals are common examples. The core solution, however, is to employ the inherent trust mechanisms contained in the email protocol itself. Email should be automatically encrypted and digitally signed to ensure message integrity and sender authentication to eliminate spear phishing attacks.

Trust mechanisms for email have existed for decades, but unfortunately these remain mostly unused or misunderstood. Barriers to use include the lack of a globally scalable publishing & retrieval mechanism for end-user cryptographic certificates and the complexity of current email security solutions. Ease-of-use is a common objection from anyone who has set up or renewed personal email certificates in laptops and mobile devices. Automated policy enforcement is also lacking.

Extensions to the DNS-based DANE protocol have been published [7, 8, 9] to address improvements for email security. In a nutshell, these DANE extensions use the existing infrastructure of DNS and DNSSEC to create a secure global repository of end-user X.509 certificates and the cryptographic credentials that authenticate email servers.

By itself, however, it would be unlikely for DANE to be widely deployed for the same reasons that S/MIME is not widely used; the lack of simple-to-use end-user solutions. The current email security ecosystem has multiple interdependent components that involve PKI certificate authorities, DNS provisioning systems, email host servers, and email client programs that run in a variety of end-user devices such as laptops, tablets and smartphones. Consumers today are used to a world where entire solutions are available by simply "downloading an app", not by having to integrate pieces from multiple sources using a complicated set of installation instructions from a variety of vendors.

The benefits of DANE will not be realized without catalyzing its technology within a broader approach for ease-of-use. Due to email's history, evolution, and the wide variety of vendors, it is also unlikely that there will be a day-one event in which all components are simultaneously interrelated.

This paper suggests methods to overcome these usage barriers via an incremental approach towards ease of use. We advocate automation techniques to manage the provisioning, maintenance and policy directives for credentials. Each step is useful in its own right; combined together they bring us closer to a more complete and deployable solution for end consumers. We also describe current pilot implementations of DANE email extensions and proposed international government mandates.

The remainder of this paper is organized as follows. Section 2 gives background regarding basic email security mechanisms. Section 3 describes the IETF DANE email extensions. Section 4 suggests components for automation and ease-of-use that would enable wider deployment of DANE. Section 5 concludes the paper.

2. Background and Related Work

2.1. S/MIME and OpenPGP: Use and Limitations

The email protocol [10] is over 30 years old and was originally restricted to text-only messages. It was later enriched with Multipurpose Internet Mail Extensions (MIME) for attaching files, formatted text, HTML audio, video, applications and graphics [11, 12, 13]. This extended its usefulness beyond measure. Trust mechanisms for confidentiality, authentication and data integrity were addressed by extending email with Secure/MIME (S/MIME) [14, 15] and with an alternative method, OpenPGP for MIME [16]. Both S/MIME and OpenPGP use public key cryptography to digitally sign and encrypt email messages.

Public Key Cryptography is a method in which an email user generates a public/private key-pair that is either signed by a Certificate Authority (CA) or encoded into a self-signed certificate. The added value the CA brings is that it is a third party that is vouching for some portion of the identity metadata stored in the certificate along with the public key. The public key certificate is meant to be globally available to anyone so that they may use S/MIME to encrypt email. The private key, held only by the email recipient, is used to decrypt these messages. The private key is also used to generate digital signatures for email. Since only the sender has the private key, this mechanism ensures authenticity of the email sender and additionally ensures that no changes were made to the message (data integrity). Fraudulent email will not be able to be signed.

Unfortunately, use of S/MIME today is spotty at best. The trust mechanism is cryptographically sound, but operational issues have stalled its use. These include:

- Creation of user key-pairs and installation of private keys onto multiple devices.
- Global Distribution of public key certificates.
- Lack of a name-space to authenticate public key certificates

- Resisting spammer techniques such as ”cousin domains”
- Lack of enforced policy and feedback mechanisms

The manual steps involved in generating and installing personal cryptographic keys can be difficult and time-consuming, therefore most users simply don't do it. Furthermore, users have multiple devices and multiple email identities. Transporting private keys from a laptop to a smartphone or tablet is a possible but confusing process. The end result: no keys, no trusted email, increased risk.

Assuming a user has mastered the art of key installation and management, the next step is to distribute the public key certificates. Unfortunately, there is no global key repository in which one can publish and retrieve the public key of an individual. Instead, it is usually done by S/MIME users manually distributing keys to desired recipients by sending them a digitally signed email. OpenPGP distributes keys using a web of trust via ”key-exchange parties” and a limited set of well-known key exchange servers. Neither S/MIME nor OpenPGP scale well and this limits use. A vendor cannot send encrypted email to a customer for whom the key is unknown.

Another operational problem is the existence of fraudulent certificates. It is possible for rogue CAs to generate fake server or email certificates. Recipients don't normally examine email certificates to see if they are correct. They assume that if a certificate exists, it must be valid. To avoid using malicious credentials, it is desirable to link the authorized certificates into a global managed name space such as the Domain Name System (DNS). This is described further in the next section.

Related to fake certificates is the use of “cousin domains”, defined by Steve Crocker as “*a registered domain name that is deceptively similar to a target domain name. The target domain is familiar to many end-users, and therefore imparts a degree of trust. The deceptive similarity can trick the user by embedding the essential parts of the target name, in a new string, or it can use some variant of the target name, such as replacing 'i' with 'l'.*” As an example, an email from `someone@examp1e.net` (using a “one” character instead of the letter “l”) might easily be mistaken for the legitimate `someone@example.net` even if digitally signed by the fraudulent domain owner.

S/MIME by itself has no policy directives or feedback mechanisms. Automated policy enforcement could tighten the controls on acceptance or rejection of emails and provide feedback on failure mechanisms. A simple example would be to create a mailbox for a user that *only* accepts digitally signed email. All others (e.g. spear phishing messages) would be rejected. Another policy could be to enforce sender signing and encryption.

Sections 3 and 4 will describe methods to overcome these obstacles to make trusted email pervasive.

2.2. SPF, DKIM and DMARC

Because of the enormous growth in spam and phishing, various methods have been developed to limit their propagation. All of these methods use the DNS to publish and retrieve IETF standard records that dictate policy to an email server. Organizations such as the Anti-Phishing Working Group (APWG) and the Mobile, Messaging and Mail Anti-Abuse Working Group (M3AAWG) have encouraged their adoption. Although very useful in the context of spam, note that these do not constitute a full trust model for email. However they do complement S/MIME and DANE and would be incorporated in a comprehensive email solution. They are described here for completeness.

Sender Policy Framework (SPF) [17, 18] is a simple method to detect email spoofing by letting a sending domain identify and assert the authorized mail senders for a given domain. SPF removes guesswork as to the authenticity of a sending email server. This benefits receivers by allowing greater accuracy in quarantining and blocking.

DomainKeys Identified Mail (DKIM) [19] is a method to detect email spoofing by checking whether incoming mail from a domain has ben actually sent from that domain. Authorized sending email servers cryptographically sign all email headers (and email bodies) with the domain's private key. This signature allows the receiver to verify that email purported to come form a specific domain is authorized by the owner of the domain. It also allows verification as to whether headers or the message body was tampered with after it left the sending email server. The private key used to generate signatures is common to all email messages from that server. This means that DKIM does not offer true end-to-end digital signing, as the sending MTA generates the DKIM signature, not the original sender of the message. Verification is carried out at the receiving MTA using the domain's public key that is published in the DNS.

A problem with SPF and DKIM is the lack of feedback regarding its effectiveness. How many emails were blocked? Were mistakes made in setting policies or have all authorized senders been accounted for? Can a domain test the effectiveness of DKIM before fully turning it on?

Domain-based Message Authentication, Reporting, and Conformance (DMARC) [20] was defined to address these issues. DMARC was conceived to allow email senders to specify policy on how their mail

should be handled, the types of reports that receivers can send back and the frequency of reports. DMARC allows domain owners to know the extent to which unauthorized senders are using their domain.

2.3. Proprietary Systems

A number of commercial and open source products have been created to fill the void in email security. These can be appliances or cloud-based SaaS (Software as a Service). Systems include firewall products from FireEye, Cisco, SonicWall and others. SpamHaus, Sophos and Barracuda produce real-time query systems to determine if email is coming from a non-trusted source. AntiSpam protection and email security gateways are available from MXLogic (acquired by Intel), TrendMicro, FortiNet and others.

Proprietary email encryption products have been created due to the S/MIME limitations outlined earlier. Zix and ProofPoint are example products used by companies that need a fully functional email encryption solution. These are closed systems, however, and all parties have to use the same solution environment. Typically used in the financial sector, these proprietary solutions can be complex, and are neither universally available across diverse groups nor interoperable due to their walled-garden nature.

3. DANE

Supply Chain Management is a global process. Its diverse community of suppliers, customers and integrators typically use differing processes and systems. Interoperability of trusted email across this community is an absolute requirement. Proprietary solutions are inadequate due to their closed nature. A standards-based approach to trusted email, on the other hand, achieves universality and interoperability.

This leads to using standard S/MIME; it already exists and is available across all mail servers and clients. In fact, S/MIME can be and is used today, but the challenge in managing key distribution makes global scaling difficult. This limitation can be overcome, however, by means of the DANE protocol. DANE uses the global DNS infrastructure to overcome key distribution issues. It also solves problems in securing communication between mail exchange servers. Its use of the *existing* DNS infrastructure implies that solutions are readily deployable and affordable.

3.1. The DANE Mechanism

DNS-based Authentication of Named Entities (DANE, RFC6698) [7, 8] is a mechanism used to bind X.509 certificates into the DNS. The records are made cryptographically secure via the DNSSEC security extensions [21, 22, 23]. DANE can be used to store self-signed certificates, or to authorize specific X.509 certificates from a registered CA. It does this by publishing the X.509 certificate (or fingerprint thereof) in the appropriate specialized DANE resource record according to its usage: `TLSA` for certificates used to support TLS in applications, `OPENGPGKEY` or `SMIMEA` to support OpenPGP and S/MIME respectively.

One motivation for creating DANE was to solve issues with the existing X.509 Public Key Infrastructure (PKI). DANE, for example, addresses rejection of fraudulent certificates, permits simpler handling of certificate revocation, creates a mechanism for global publishing and retrieval of certificates, and allows the authorization of self-signed certificates.

DANE achieves these goals by using the *delegation property* of the DNS name space, meaning that only authorized domain owners can place records in their DNS domain. As an example, only the "example.com" corporation can place records in the `example.com` DNS name space. No one else can do so because they do not have access to the delegation. Delegation enables the creation of an authorization mechanism.

The first application of the DANE protocol was for the authentication of TLS certificates used by web servers. Consider a web site `www.example.com`. Assume that multiple certificates exist for that site, a real one and several fraudulent ones used by attackers for man-in-the-middle attacks (MITM). How can `www.example.com` protect itself? The solution is for the domain owner to insert a DANE `TLSA` record in the `www.example.com` DNS namespace to authorize only the genuine certificate. Web clients that retrieve certificates from a server can also retrieve the DANE record and match it against the certificate. If the DANE record exists and matches, the certificate is authorized and the connection is accepted. If the record does not match, the certificate is rejected and the connection is denied.

The DANE protocol is meant to be generic and multi-purpose. Application-specific use of DANE is defined in separate RFCs. Email usage is defined in two documents: RFC 7672 [24] defines `TLSA` records to secure the SMTP protocol for email servers, and an IETF draft document [9] defines `SMIMEA` records to secure end-user email certificates. We will explore each of these in turn.

3.2. DANE for MTA-MTA Security

A simplified email architecture is illustrated in figure 1. Email clients are programs such as Outlook, Apple Mail or Thunderbird that run in user devices (smartphones, tablets, laptops) to compose, send and retrieve email. Email is sent from these clients to Mail Transfer Agents (MTA) that store and forward the messages among themselves and finally to the recipient email client.

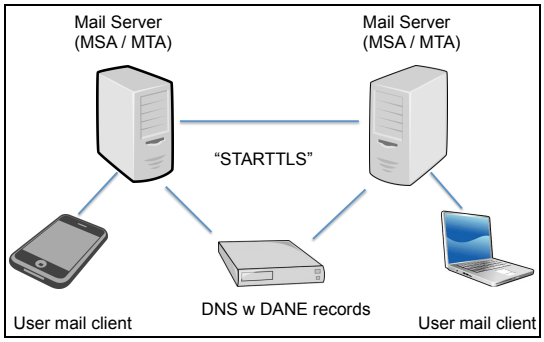

Figure 1: Simplified Email Architecture

Mail Transfer Agents will encrypt data sent from one MTA to another if TLS is available. This is a privacy measure for data-in-motion only. Once transferred, the data-at-rest is in plaintext.

Unfortunately, The original SMTP protocol did not accommodate TLS. To fix this, a new command, `STARTTLS`, was added to the protocol. `STARTTLS` modifies an existing insecure connection and upgrades it to a secure connection using SSL/TLS. The `STARTTLS` implementation, however, employs *opportunistic TLS*; that is, the receiving server can refuse the command and data communications between the two servers will continue in plaintext.

Opportunistic TLS creates vulnerability. An attacker can use a man-in-the-middle *downgrade* attack by simplify refusing the `STARTTLS` request. This allows eavesdropping and potential message modification by an attacker.

DANE eliminates this vulnerability as illustrated by the block diagram in figure 1. Before issuing a `STARTTLS`, the sending mail server will query the DNS for the DANE `TLSA` record associated with the receiving server. If a record exists, `STARTTLS` becomes *mandatory*. If a server refuses the `STARTTLS` request or if the certificate does not match the DANE `TLSA` record, communication between the servers will cease and the email server will wait to send the message at a later time. If a `TLSA` record does not exist, opportunistic TLS is still used. The absence or presence of a `TLSA` record permits incremental deployment of this DANE security mechanism.

DANE therefore achieves two goals for MTAs: it authenticates the receiver (certificate match), and enforces confidentiality via encryption between MTAs. Several email servers have already been modified to take advantage of this capability, including the popular open-source *Postfix* server.

3.2.1 Current Deployment of MTA-MTA Security Using DANE

The use of DANE for SMTP was specified in 2015 so deployment has been sparse as developers add the functionality to their implementations. There has been a sizable deployment within Germany and some experiences have been documented [25]. Using TLSA RRs to publish certificate information has been called out by the German Federal Office of Information Security as mandated for deployment as part of the "Email Made in Germany" initiative [26].

3.3. DANE for End-User Email Security

As mentioned, DANE for MTAs protects data-in-motion only. It does nothing for end-user authentication, digital signatures or data-at-rest encryption. For this we must use S/MIME. But the challenge has always been key management and distribution.

Assume employees in two organizations, `purple.com` and `green.com` need to communicate with each other using confidential and authenticated email. The employees have already obtained X.509 certificates. But how do personnel at either company obtain access to the public certificates of employees from the other company? There is no global public repository or "certificate phone book", where one can easily look up this information. As we explain below, however, DANE does provide just such a capability by publishing records in the global DNS.

Internet draft [9] extends DANE by defining the `SMIMEA` record. `SMIMEA` follows the same format as a `TLSA` record, but is used to store X.509 certificate data for individual users. The draft also defines a method to convert an email address, `john.doe@purple.com` into a domain name. The domain name uses a truncated SHA-256 hash of the user name to provide rudimentary privacy. The data stored in the `SMIMEA` record could be a complete X.509 certificate or a fingerprint. The DNS, secured by DNSSEC, is now a trusted repository or an authentication method for end user email certificates.

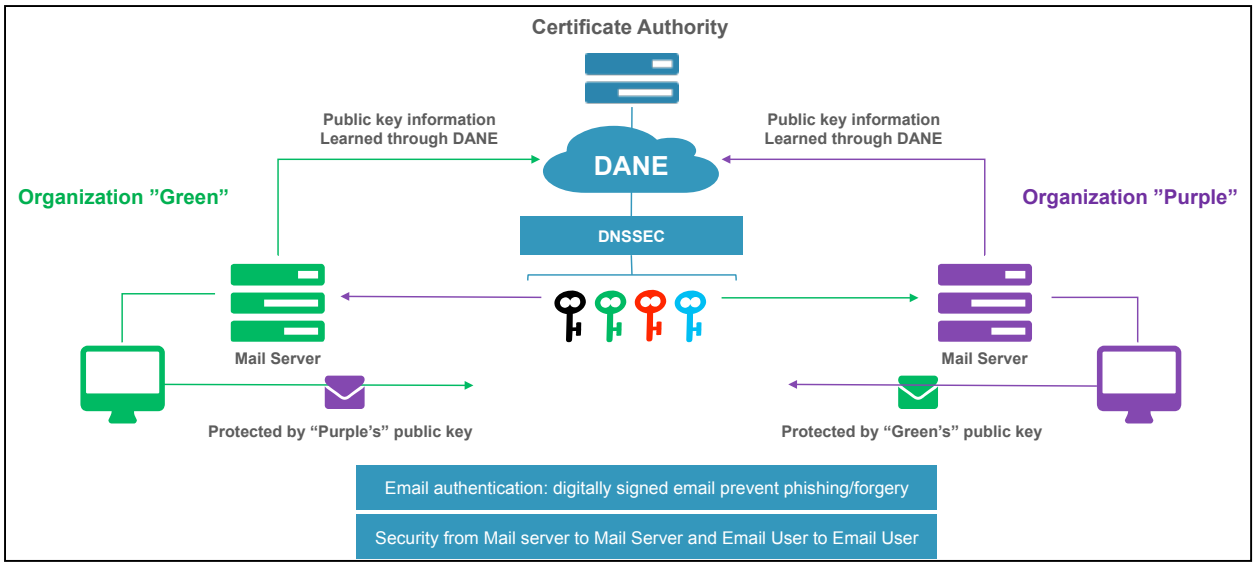

Figure 2: DANE system showing certificate retrieval for MTA and end-users (source: NIST [27])

The process for end-user security is illustrated in figure 2. A user at `green.com` digitally signs a message with her private key. This is done directly in the email client on her device. Next, in order to encrypt the message, a query to the DNS is made to retrieve the recipient's public key certificate. This certificate is cached by the user for future use and used to encrypt the message. Using the public key certificate ensures that only the recipient can decrypt the message. Performing the operation in the recipient's device ensures data-at-rest confidentiality. The signed and encrypted message is then transferred to the recipient through another encryption layer at the MTA to MTA level.

When the employee at `purple.com` receives the message, it is decrypted on his device via his private key. The user now needs to authenticate this message; did it really come from the sender at `green.com`, or is it a cleverly crafted spear phishing message? To confirm authenticity, the email program must check the veracity of the digital signature. This is done by performing a DNS lookup of the sender's public key certificate. The public key is used to decrypt the digital signature and perform a data integrity check. If the signature validates correctly, the message is authentic. It has not been altered in transit and the originator has been confirmed.

3.4. Policy: DMARC applied to DANE

As described in section 2.2, DMARC defines policy directives dictating the behavior of DKIM. It also provides a feedback mechanism to report on actual behavior. DANE can benefit from a similar mechanism. To that end, a draft proposing DMARC extensions for DANE [28] is currently a work-in-progress at the IETF. Sample policy directives include

- Receiver mail must be signed
- Receiver mail must be encrypted
- Sender mail must be signed
- Sender mail must be encrypted

Without these policies, users have to be attentive as to whether a received email has been digitally signed, typically indicated by an icon somewhere in the message. These indicators are easy to miss. Unsigned spear phishing messages without the icon could arrive unnoticed and potentially be acted upon.

To build strong protection, an organization could construct two inboxes for users. The "protected" inbox would enforce strict policy dictating all incoming email must be signed. The "unsafe" inbox accepts all mail, signed or unsigned. Official company business would be conducted within the protected mailbox. Other business could still be handled in the unsafe mailbox, but users now have the burden of checking for signatures.

Feedback mechanisms are currently being defined, but typically would report on various metrics such as failure counts, etc.

3.5. Objections and Alternatives to DANE

While this paper advocates the usage of DANE, there are several criticisms of the method. The Internet blog articles [29, 30] discuss its dependency on

DNSSEC and prompted many pro and con arguments. It should be noted that [30] limits its discussion to DANE for web site validation, as SMIMEA had not yet been introduced.

A counter-proposal for securing MTA-to-MTA communication has been proposed in a working draft at the IETF [31] for SMTP Strict Transport Security (STS). Like DANE, DKIM and DMARC, this protocol also publishes records in the DNS, however STS does not require DNSSEC. SMTP-STS is similar to STS for web servers, but modified for relevancy to SMTP. It works by having the receiving domain publish its security policy at a well-defined URL, which a sender accesses using HTTPS. Advantages are that it defines policies and feedback reporting and does not mandate the use of DNSSEC. Disadvantages are that it can be spoofed or DDoSed (Distributed Denial of Service) to make it appear that a policy is nonexistent. In addition, sending MTAs must now use HTTPS to insure that a secure channel exists. In contrast, DANE with DNSSEC has secure responses and proof of nonexistence built in.

STS is a trust mechanism for MTA to MTA only. It does nothing for client certificates used for end-to-end encryption and digital signing. SMIMEA remains as a viable key distribution method.

Research on the robustness, security, resilience and efficiency of DANE are only beginning at this time. This is a topic for future development. Current pilot programs are focusing on interoperability and core features.

3.6. Deployments & Government Programs

DANE can be deployed today and multiple organizations have already done so. The Internet Society Deploy360 Programme has created a website [32] listing some current deployments.

Of particular note is the trusted email showcase and testbed at NIST's National Cybersecurity Center of Excellence (NCCoE) [33]. The purpose of this NCCoE project is to demonstrate interoperability among commercially available DANE technologies from various suppliers. The use and setup of these technologies is being prepared to help government and private enterprise deploy DANE on their own.

The testbed has several environments contributed by Microsoft, Secure64, NLNetLabs, and ISC-Bind. Each environment contains DNSSEC servers, email servers, and email clients making use of DANE. Email can be exchanged between the environments to demonstrate interoperability in MTA-to-MTA security as well as end-user security with DANE S/MIME.

NIST has also published an excellent reference document to describe the principles and techniques currently available for secure email: *Trusted Email (Special Publication SP-800-177)* [27].

Other government involvement includes the drafting of proposed mandates that require DANE. The German government has published *BSI TR-03108 Secure E-mail Transport* [26], dated August 2015, requiring the use of DANE.

4. Workflow Automation

DANE removes the biggest limitation to using S/MIME on a global scale by creating a secure public repository of email certificates. The other limitations listed in section 2.1 still need to be addressed, as well as methods to make DANE easier to use. There are complexities in its use that begs for a more complete and automated solution.

As an example, an organization could use DANE as it exists today, however deployment would likely be limited to a small scale. This is because `TLSA` and `SMIMEA` records have to be manually generated and maintained. Mistakes are easy to make. Managing a trusted email environment will be difficult without proper tools and processes.

The objectives for managing a trusted email environment include the ability to automate DNS provisioning, integrate company workflows, simplify end-user activities, and manage company policy. To that end, the following items are being developed or already exist to assist DANE deployment and operations:

- Automated DNS zone file provisioner for TLSA, SMIMEA, SPF, DKIM and DMARC records.
- Automated DNSSEC signing appliances (e.g. Secure64, OpenDNSSEC) or DNSSEC enabled managed DNS services.
- Interfaces to Human Resource credential databases (e.g. Active Directory, etc.)
- Means to acquire or generate X.509 certificates either with an API to commercial CA accounts, an enterprise local CA, or tools to generate self-signed certificates.
- GUI objects and wizards to manage trust policies expressed as DMARC and DMARC/DANE records. Interfaces to legacy DMARC generation systems.
- Mail servers with milters for handling DANE and DMARC policy directives.
- Blacklist managers to block cousin domains.
- Integration with legacy email gateways and DMARC data collectors / report generators (e.g. Agari and others).
- Mobile Device Managers that provision personal devices with X.509 certificates and interface to

the DNS provisioner to exchange publishable trust data.

- Key escrow as an option for organizations whose policy requires that they have copies of end-user's private keys.
- Logging and Auditing with interfaces to SIEM systems.

A critical solution component is a DNS Provisioning System. The basic function of DNS record provisioning is to manage the workflow for creation and maintenance of DNS records and zone files. This task must be done with little or no manual intervention. Software API's link the provisioner to other components such as employee databases (e.g. *Active Directory*), mobile device managers (MDM), and Certificate Authority APIs. The provisioner could run locally or as a service in the cloud. The workflows to be managed are:

- *MTA management*: Automate discovery of mail servers and provisioning of TLSA, DKIM, DMARC and SPF records. Dynamically Maintain records due to external events such as server changes or certificate expiration or revocation. Interface with commercial systems for DMARC feedback (e.g. Agari). Software wizards and GUI objects would be needed to assist in policy definition for DANE and DKIM as well as interfaces to commercial systems for this function.
- *Employee Credential Management*:
  - *Initial setup*: Scrape the employee database to create S/MIMEA records. Company policy would dictate if private keys are owned by the corporation or by the individual. If the corporation owns the keys, certificates can be generated by a central system. If the individual owns the private keys, then only the individual's end user device should create key-pairs and the provisioner device API or MDM API will fetch this from user devices.
  - *Employee hire or termination*: This is best handled directly in the HR department through an API. Credentials would be established or revoked and the provisioning system would update its local database and DNS zone files on demand.
  - *Certificate renewal or revocation*: Manage the integration with HR databases, Certificate Authorities, and MDM to end-user devices; update records in the DNS.

Provisioning handles the supply side of certificate publication. The retrieval side is also in need of automation and simplification. Modern email clients such as *Outlook, Office 365, Apple Mail,* and *Thunderbird* have built-in encryption and digital signature verification using S/MIME. What they do not currently have is the ability to automatically retrieve DANE-formatted public keys from the DNS as mail is being composed and sent.

End-users should be able to use email with little or no change to existing email usage. This requires transparent integration of DANE into the end-user devices and easier methods to install user credentials.

It is expected that vendors of mobile and PC-based email clients will add this capability. In the temporary absence of such systems, simple standalone applications can fetch credentials from the DNS to store public certificates in the end-user device. Plugins to mailers such as *Thunderbird* have already been developed to make this step automatic.

4.1. Incremental Deployment

These suggestions for automation and usability may take some time to be fully realized. The authors wish to emphasize, however, that the core elements to build a DANE-enabled email system using manual steps is immediately available. Additional functionality can be implemented incrementally over time as new tools become available. A possible sequence of events is as follows:

1) Manual provisioning of DANE records: Early adopters are demonstrating the benefits of DANE, but manual implementation is impractical for the wider audience. Nevertheless, scripts are currently available for constructing TLSA and SMIMEA records and these can be installed in an organization's authoritative DNS servers immediately.

2) Manual retrieval of certificates: some standalone apps and email client plugins are available and more are under development. These programs access email contact lists from user devices to fetch email credentials and insert them into the device's keystore. These tools require manual intervention by the user rather than the more desirable goal of transparent fetching of credentials within an email client.

3) Mail Filters (Milters) to automatically fetch encryption certificates at the mail server. SMILLA [34] is an existing example milter that encrypts mail at the server instead of at the sending email client. This provides a "90% solution"; that is, it provides data-at-rest protection at the end user device with messages

uniquely encrypted for the recipient, but does not perform encryption at the first mile between the sending device and the mail server. The solution is still useful, however, since this first mile communication is typically encrypted with TLS.

4) Future automated provisioning of SMIMEA records: organizations will be able to publish their employee certificates more easily, but the recipients of email will still have no client applications (e.g. Thunderbird, Exchange, Apple Mail) to automatically retrieve these certificates.

5) Future fully transparent email client integration: No manual intervention required by the end-user to retrieve public keys.

6) Future security policy Enforcement on servers and end-user devices: creates the possibility of inboxes that only accept signed and/or encrypted email.

5. Conclusion

The authors have demonstrated the need for trusted email in supply chain management. Spear phishing, forgery, and other attacks can result in data breaches, industrial and government espionage, installation of malware, and financial theft. DANE email extensions are then posited as a solid foundation for global trusted email. DANE creates a secure repository for publishing and retrieving email credentials and policy directives on a globally scalable basis.

Research on the efficiency, security and robustness of DANE email, as well as in-depth comparisons to other technologies is only in the starting phases. Current pilots are focusing on interoperability and core functionality. This is a topic for future development.

Despite its promising capabilities, however, the basic DANE protocol is nothing more than an enabling technology at this time. A complete trusted email solution will require the development of an ecosystem of automated tools, procedures and the incorporation of new DANE features into existing popular email client programs.

To this end, supplementary automation tools to manage the workflow of DANE are proposed. The tools discussed are for both sides of the equation: the automated publishing of email certificates as well as email client features and plugins to simplify the retrieval of certificates and make user interaction with trusted email as transparent as possible. Some components are under development; some already exist.

Finally, the authors encourage supply chain managers to reduce risk by protecting their email with basic DANE technology as soon as possible. Implementing DANE trusted email manually with the existing infrastructure components is definitely possible using published scripts and tools. The basic functionality can then be expanded into a more robust, automated, and easier to use solution for a more general audience as additional tools become available.

6. References

[1] McCaney, K. "To hackers, government users are phish in a barrel". GCN, March 19, 2012. https://gcn.com/articles/2012/03/19/phishing-goverment-cyber-attacks-us-cert.aspx.

[2] OMB. Fiscal year 2011 report to congress on the implementation of the federal information security management act of 2002, 2011. https://www.whitehouse.gov/sites/default/files/omb/assets/egov docs/fy11 fisma.pdf.

[3] Cisco. Email Attacks: This Time It's Personal. June 2011. http://www.cisco.com/c/dam/en/us/products/collateral/security/email- security-appliance/targeted attacks.pdf.

[4] TrendLabs APT Research Team. Spear phishing email:Most favored apt attack bait, 2012. http://www.trendmicro.com/cloud- content/us/pdfs/security-intelligence/white-papers/wp-spear phishing- email-most-favored-apt-attack-bait.pdf.

[5] Verizon. Data Breach Investigation Report, 2016. http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016/

[6] M. Novinson, "Arrow was the target: Criminals impersonate executive, transfer money to outside bank". CRN, February 4, 2016. http://www.crn.com/news/security/300079601/arrow-was-the-target- criminals-impersonate-executive-transfer-money-to-outside-bank.htm.

[7] J. Schlyter and P. Hoffman. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. RFC 6698, August 2012. https://tools.ietf.org/html/rfc6698

[8] Shumon Huque, Dan James, and Viktor Dukhovni. TLS Client Authentication via DANE TLSA records. Internet-Draft draft-huque-dane-client-cert-02, Internet Engineering Task Force, January 2016. Work in Progress. https://datatracker.ietf.org/doc/draft-huque-dane-client-cert/

[9] J. Schlyter and P. Hoffman. Using Secure DNS to Associate Certificates with Domain Names For S/MIME.

Internet-Draft draft-ietf- dane-smime-10, Internet Engineering Task Force, February 2016. Work in Progress. https://datatracker.ietf.org/doc/draft-ietf-dane-smime/

[10] J. Klensin. Simple Mail Transfer Protocol. RFC 5321, October 2008. https://tools.ietf.org/html/rfc5321

[11] N. Freed and Dr. N. Borenstein. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. RFC 2045, November 1996. https://tools.ietf.org/html/rfc2045

[12] N. and Dr. N. Borenstein. Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. RFC 2046, November 1996. https://tools.ietf.org/html/rfc2046

[13] K. Moore. MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text. RFC 2047, November 1996. https://tools.ietf.org/html/rfc2047

[14] S. Turner and B. Ramsdell. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Certificate Handling. RFC 5750, January 2010. https://tools.ietf.org/html/rfc5750

[15] S. Turner and B. Ramsdell. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification. RFC 5751, January 2010. https://tools.ietf.org/html/rfc5751

[16] T. Roessler, M. Elkins, R. Levien, and D. Del Torto. MIME Security with OpenPGP. RFC 3156, August 2001. https://tools.ietf.org/html/rfc3156

[17] W. Schlitt and M. Wong. Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1. RFC 4408, April 2006. https://tools.ietf.org/html/rfc4408

[18] M. Kucherawy. Resolution of the Sender Policy Framework (SPF) and Sender ID Experiments. RFC 6686, July 2012. https://tools.ietf.org/html/rfc6686

[19] M. Kucherawy, D. Crocker, and T. Hansen. DomainKeys Identified Mail (DKIM) Signatures. RFC 6376, September 2012. https://tools.ietf.org/html/rfc6376

[20] M. Kucherawy and E. Zwicky. Domain-based Message Authentication, Reporting, and Conformance (DMARC). RFC 7489, March 2015. https://tools.ietf.org/html/rfc7489

[21] S. Rose, M. Larson, D. Massey, R. Austein, and R. Arends. DNS Security Introduction and Requirements. RFC 4033, March 2005. https://tools.ietf.org/html/rfc4033

[22] S. Rose, M. Larson, D. Massey, R. Austein, and R. Arends.. Resource Records for the DNS Security Extensions. RFC 4034, March 2005. https://tools.ietf.org/html/rfc4034

[23] S. Rose, M. Larson, D. Massey, R. Austein, and R. Arends.. Protocol Modifications for the DNS Security Extensions. RFC 4035, March 2005. https://tools.ietf.org/html/rfc64035

[24] V. Dukhovni and W. Hardaker. SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS). RFC 7672, October 2015. https://tools.ietf.org/html/rfc7672

[25] P. Koetter Sys4.de presentation at the 34th M3AAWG meeting "One year of DANE: Tales and Lessons Learned" June 2015 https://sys4.de/download/dane-maawg.pdf

[26] Federal Office of Information Security. BSI TR-03108-1: Secure E-Mail Transport (English Version). March 22, 2016. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03108/TR03108-1.pdf?__blob=publicationFile&v=3

[27] R. Chandramouli, S. Garfinkel, S. Nightingale, and S. Rose. Trusted Email. SECOND DRAFT NIST Special Publication 800-177, March 2016. http://csrc.nist.gov/publications/PubsSPs.html

[28] E. Osterweil and G. Wiley. DMARC Extensions for DANE. Internet-Draft draft-osterweil-dmarc-dane-names-00, Internet Engineering Task Force, January 2016. Work in Progress. https://datatracker.ietf.org/doc/draft-osterweil-dmarc-dane-names/

[29] T. Ptaceck and E. Ptacek. Against DNSSEC. Blog article, January 15, 2015. http://sockpuppet.org/blog/2015/01/15/against-dnssec/

[30] A. Langley. Why not DANE in Browsers. Blog article, January 17, 2015. https://www.imperialviolet.org/2015/01/17/notdane.html.

[31] D. Margolis, M. Risher, N. Lidzborkski, W. Chuang, B. Long, B. Ramakrishnan, A. Brotman, J. Jones, F. Martin, K. Umbach, and M. Laber. SMTP MTA Strict Transport Security. Internet-Draft draft-ietf-uta-mta-sts-00, Internet Engineering Task Force, May 2016. Work in Progress. https://datatracker.ietf.org/doc/draft-ietf-uta-mta-sts/

[32] ISOC Deploy360 Programme, DANE Test Sites. http://www.internetsociety.org/deploy360/resources/dane-test-sites/.

[33] W. Barker. Domain Name Based Security for Electronic Email, March 2016. https://nccoe.nist.gov/sites/default/files/library/project-descriptions/dns- secure-email-project-description-final.pdf.

[34] P.B. Koetter. Smilla – SMIMEA aware Milter. IETF DANE thread with github repository, July 2, 2015. https://www.ietf.org/mailarchive/web/dane/current/msg07865.html