A New Internet Architecture for Secure Key Learning: DANE

Eric Osterweil

Principal Scientist, Verisign Labs

August 6, 2015

Something fundamental has been missing from security protections on the Internet

- Our Internet security has had a loophole for years
  - We have TLS, IPSec, S/MIME, SSH, etc.
  - They give us: privacy, encryption, integrity protection, & more

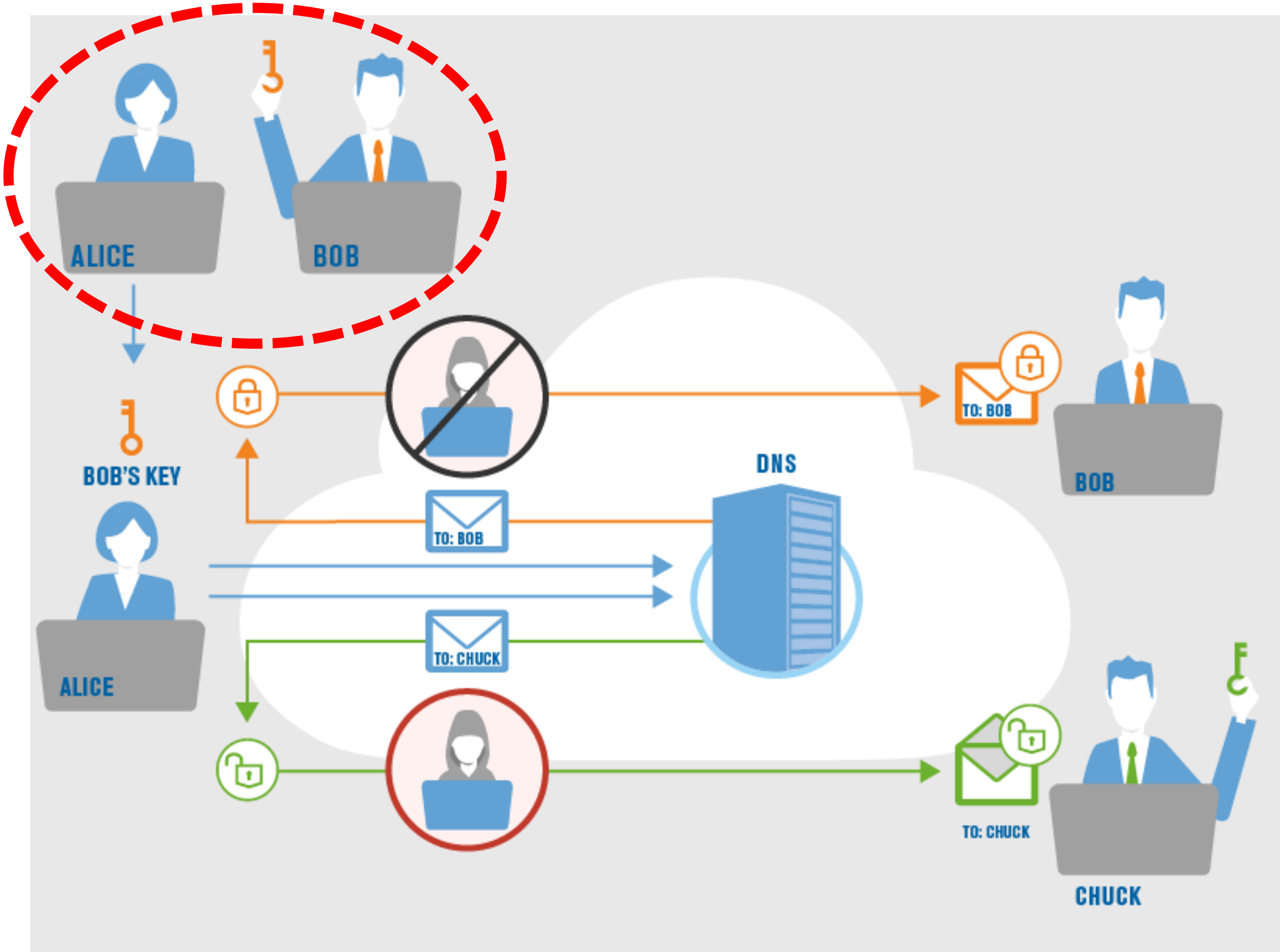
- Protections are mature, have extensive codebases, and are well understood

- But, almost all of them lack important protections during their startup phases (secure bootstrapping)

Examples of creating secure connections today (w/o DANE)

- Sending/receiving secure Inter-Administrative email (e.g. S/MIME)
  - We use out-of-band key bootstrapping to learn keys (user by user b/c we need to know ID-to-key bindings ahead of time). Then we lookup a mail domain in DNS, connect to and ask a server about an email identity and do verif/decrypting/etc w/ pre-learned keys
  - Because we can't securely learn the keys without out-of-band trust

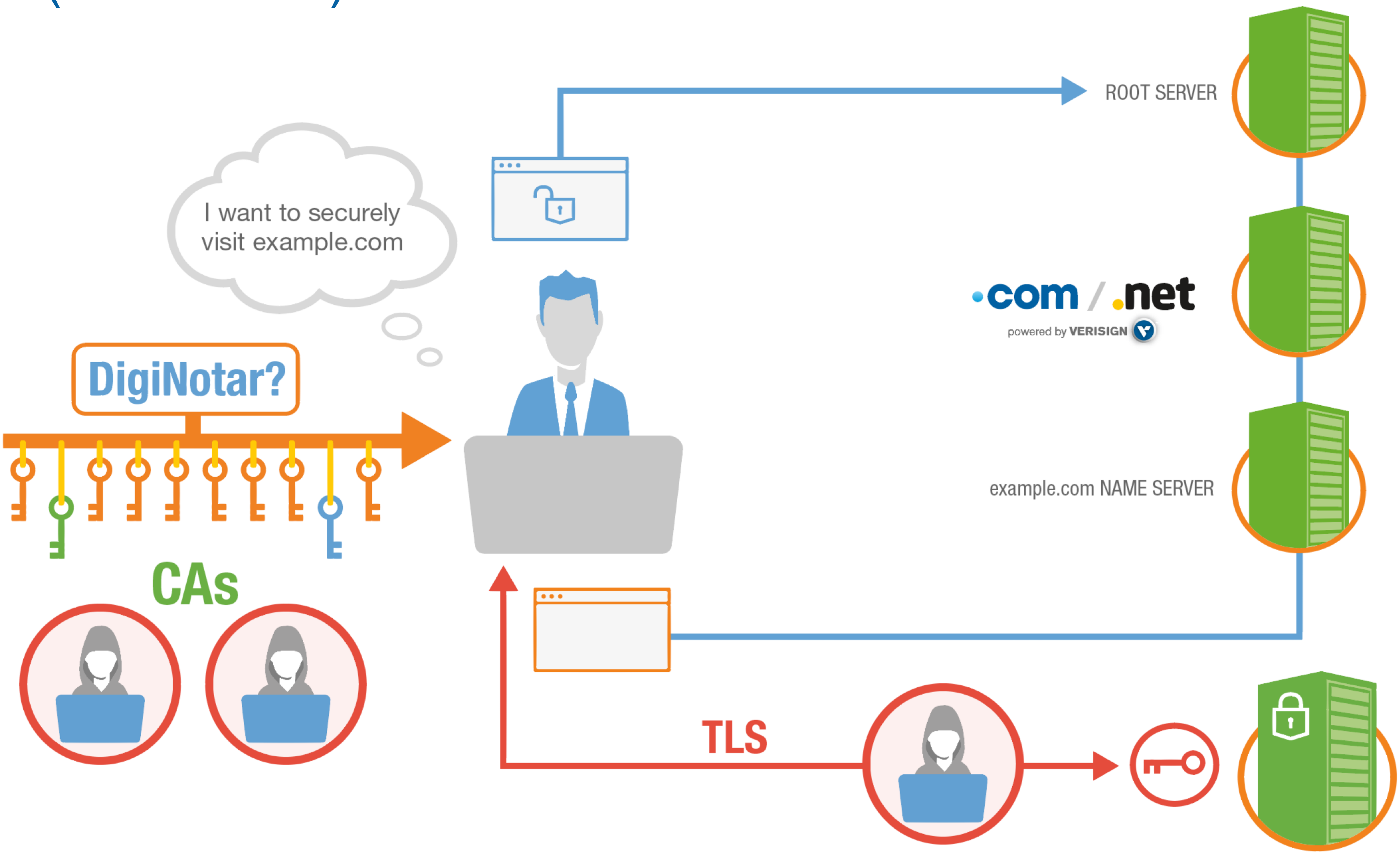
Examples of creating secure connections today (w/o DANE)



Examples of creating secure connections today (w/o DANE)

- Connecting to secure websites (e.g. HTTPS over TLS)
  - We use out-of-band key bootstrapping to get a list of globally trusted CA keys. Then we lookup a website's IP address(es) through DNS, fetch a crypto key over an insecure TCP connection, and validate its key using CA keys
  - We learn CA keys using out-of-band trust

Examples of creating secure connections today (w/o DANE)



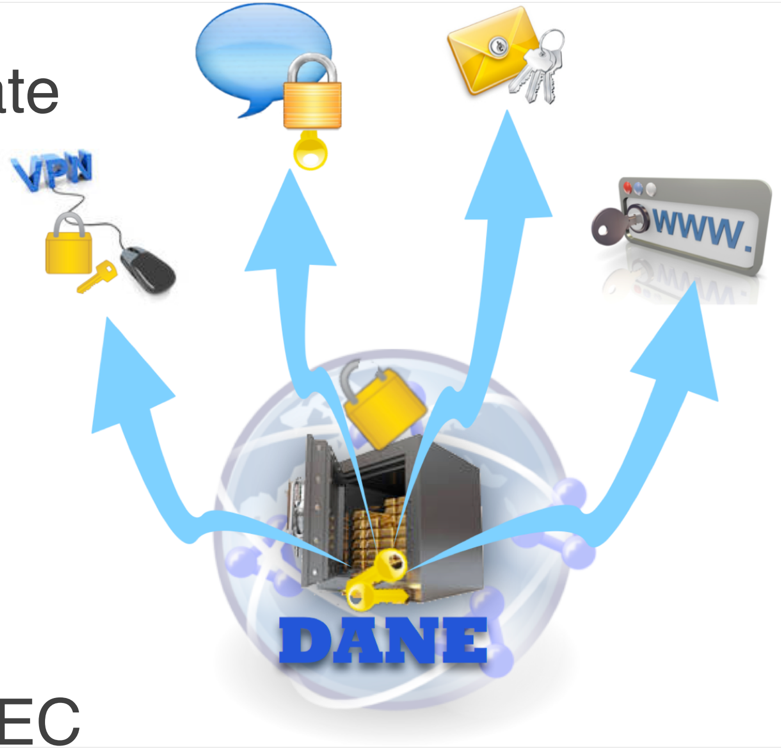
Examples of creating secure connections today (w/o DANE)

What's missing is *secure key learning*

DANE uses DNSSEC for secure key learning

- DANE: DNS-based Authentication of Named Entities

- DANE is an *architectural* substrate for Internet key learning in
  - TLS, S/MIME, PGP, IPSec, etc.



- Don't do key learning *after* DNS, do it *with* DNS

- DANE is the killer app for DNSSEC

- DANE aligns costs with incentives so that there's a reason to *ASK* for DNSSEC!

Outline

- A brief overview of DNSSEC (DANE's substrate)
- How DANE works
- Security using DANE
- Verification with the WebPKI and with DANE
- Examples of DANE protocols and open tools

Why we need DNSSEC

- DNS cache poisoning has been a known attack against DNS since the 1990s [1]

- DNSSEC was designed to *cryptographically* ensure data's origin authenticity and integrity

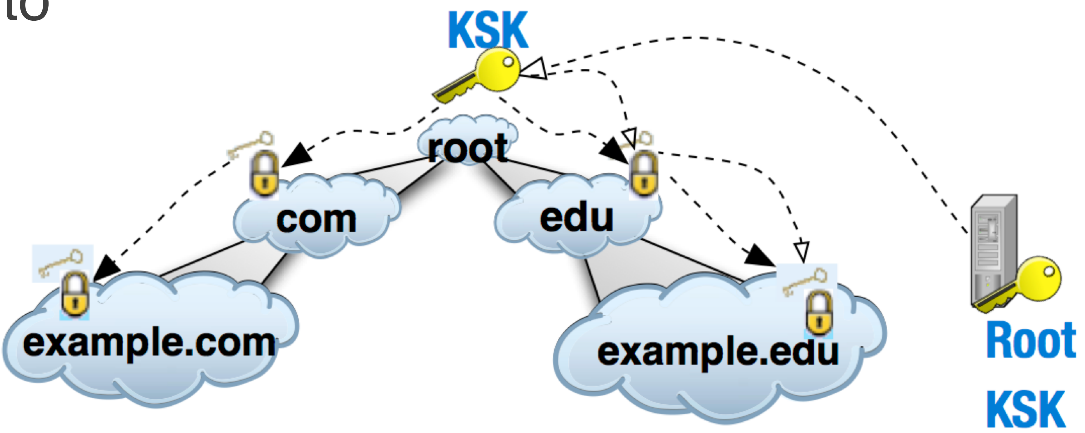
- Then came the "summer of fear" – '08
  - The Kaminsky attack
  - Patches (source port randomization) helped in the short-term

[1] Bellovin, S. M. 1995. Using the domain name system for system break-ins. USENIX UNIX Security Symposium 1995

How DNSSEC works

- First attempt to secure a core Internet protocol w/ crypto

- DNSSEC zones create pub/priv keys
  - Public key is DNSKEY

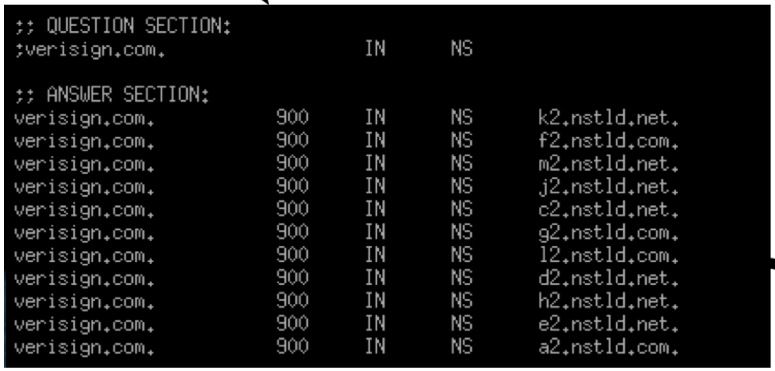


- Zones sign all RRsets and resolvers use DNSKEYs to verify them
  - Each RRset has a signature attached to it: RRSIG

- Resolvers are configured with a *single root* key, and *all* trust flows recursively down the hierarchy

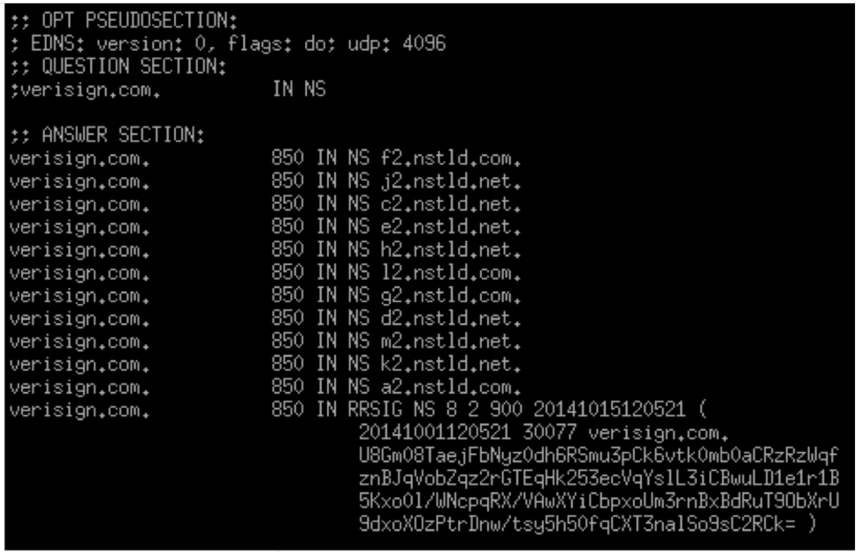
Data Signing Example

Using a zone's key
on a standard RRset
(the NS)



```
;; QUESTION SECTION:
;verisign.com.                  IN      NS

;; ANSWER SECTION:
verisign.com.           900     IN      NS      k2.nstld.net.
verisign.com.           900     IN      NS      f2.nstld.com.
verisign.com.           900     IN      NS      m2.nstld.net.
verisign.com.           900     IN      NS      j2.nstld.net.
verisign.com.           900     IN      NS      c2.nstld.net.
verisign.com.           900     IN      NS      g2.nstld.com.
verisign.com.           900     IN      NS      l2.nstld.com.
verisign.com.           900     IN      NS      d2.nstld.net.
verisign.com.           900     IN      NS      h2.nstld.net.
verisign.com.           900     IN      NS      e2.nstld.net.
verisign.com.           900     IN      NS      a2.nstld.com.
```



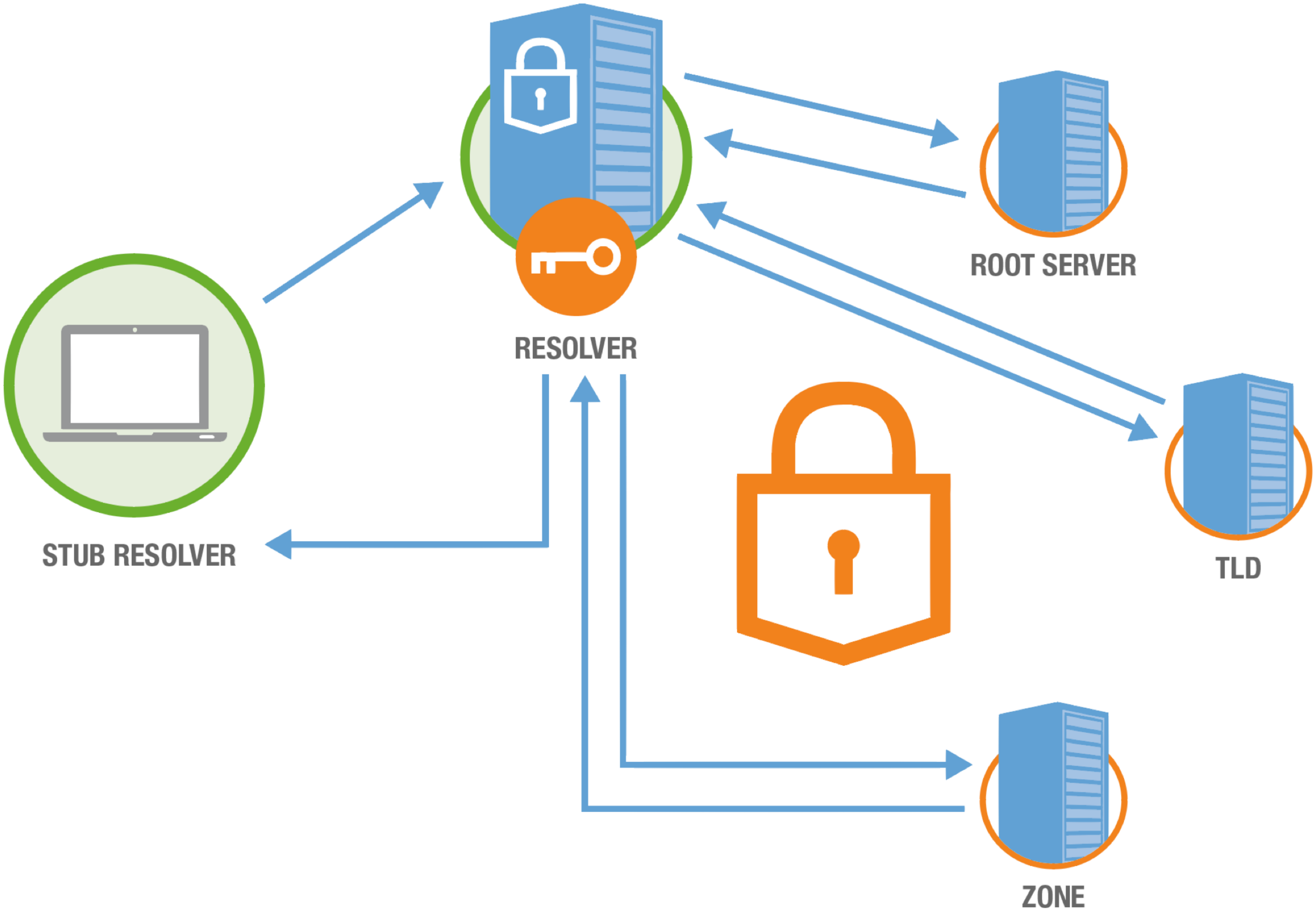
```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;verisign.com.          IN NS

;; ANSWER SECTION:
verisign.com.           850 IN NS f2.nstld.com.
verisign.com.           850 IN NS j2.nstld.net.
verisign.com.           850 IN NS c2.nstld.net.
verisign.com.           850 IN NS e2.nstld.net.
verisign.com.           850 IN NS h2.nstld.net.
verisign.com.           850 IN NS l2.nstld.com.
verisign.com.           850 IN NS g2.nstld.com.
verisign.com.           850 IN NS d2.nstld.net.
verisign.com.           850 IN NS m2.nstld.net.
verisign.com.           850 IN NS k2.nstld.net.
verisign.com.           850 IN NS a2.nstld.com.
verisign.com.           850 IN RRSIG NS 8 2 900 20141015120521 (
                                20141001120521 30077 verisign.com.
                                U8Gm08TaejFbNyz0dh6RSmu3pCk6vtk0mb0aCRzRzWqf
                                znBJqVobZqz2rGTEqHk253ecVqYslL3iCBwuLD1e1r1B
                                5Kxo0l/WNcpqRX/VAwXYiCbpxoUm3rnBxBdRuT90bXrU
                                9dxoX0zPtrDnw/tsy5h50fqCXT3nalSo9sC2RCk= )
```

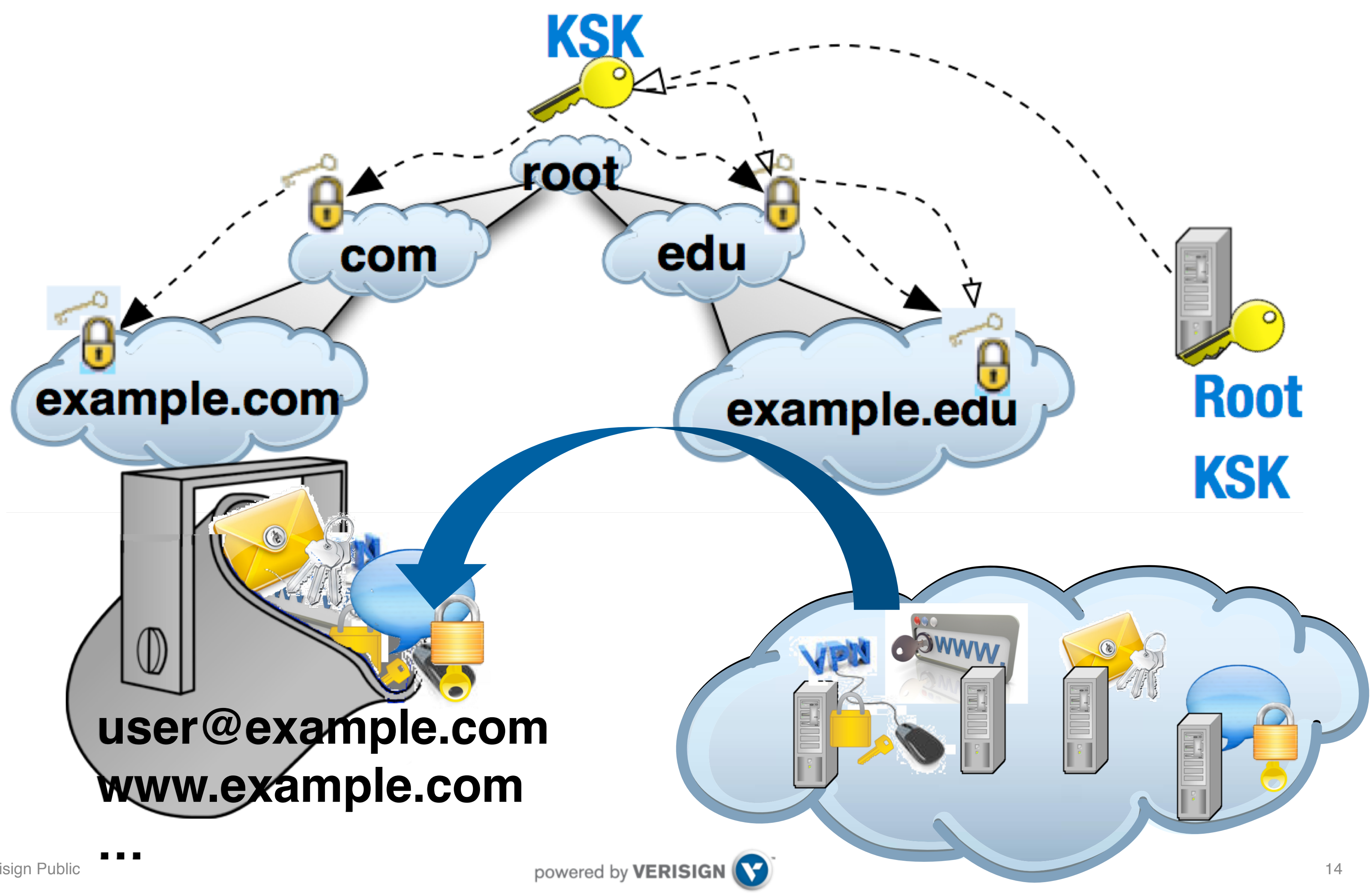
Signature (RRSIG) will
only verify with the
DNSKEY if *no*
data was modified

DNSSEC: Validating

A ***Validating Recursive Resolver*** uses the root's public key to verify (validate) delegations



DANE’s architecture



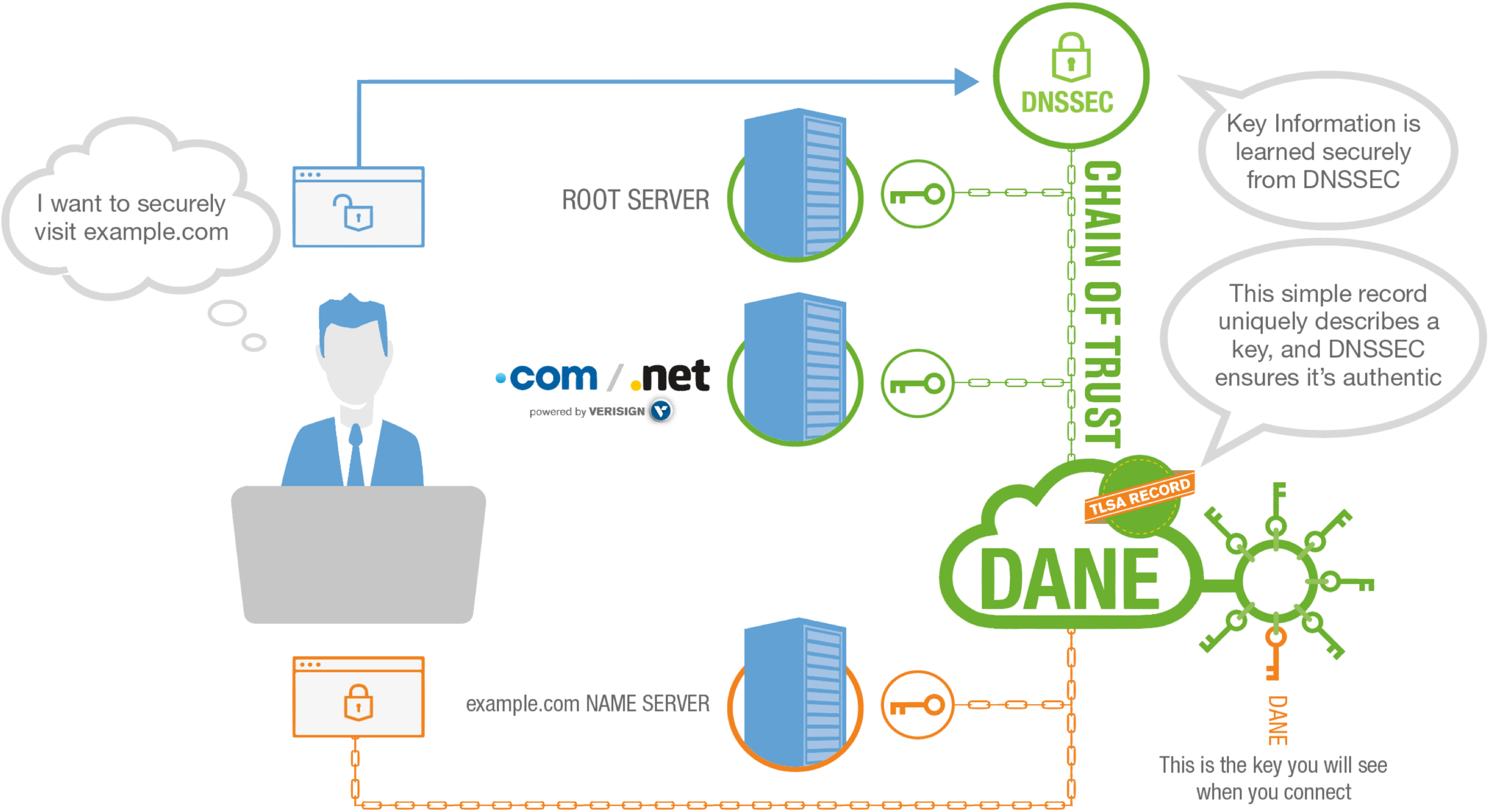
user@example.com
www.example.com
...

Application uses of DANE

- Allow applications to securely obtain (authenticate) those keys and use them in application security protocols

- Some possible applications: SSH, SSL/TLS, HTTPS, S/MIME, PGP, SMTP, DKIM, and many others ..
- DANE records:
  - TLSA
  - Upcoming: OPENPGPKEY, SMIMEA, IPSECA, …
- DANE-like legacy records:
  - SSHFP, IPSECKEY, DKIM TXT record, …

Security for TLS: Using DANE

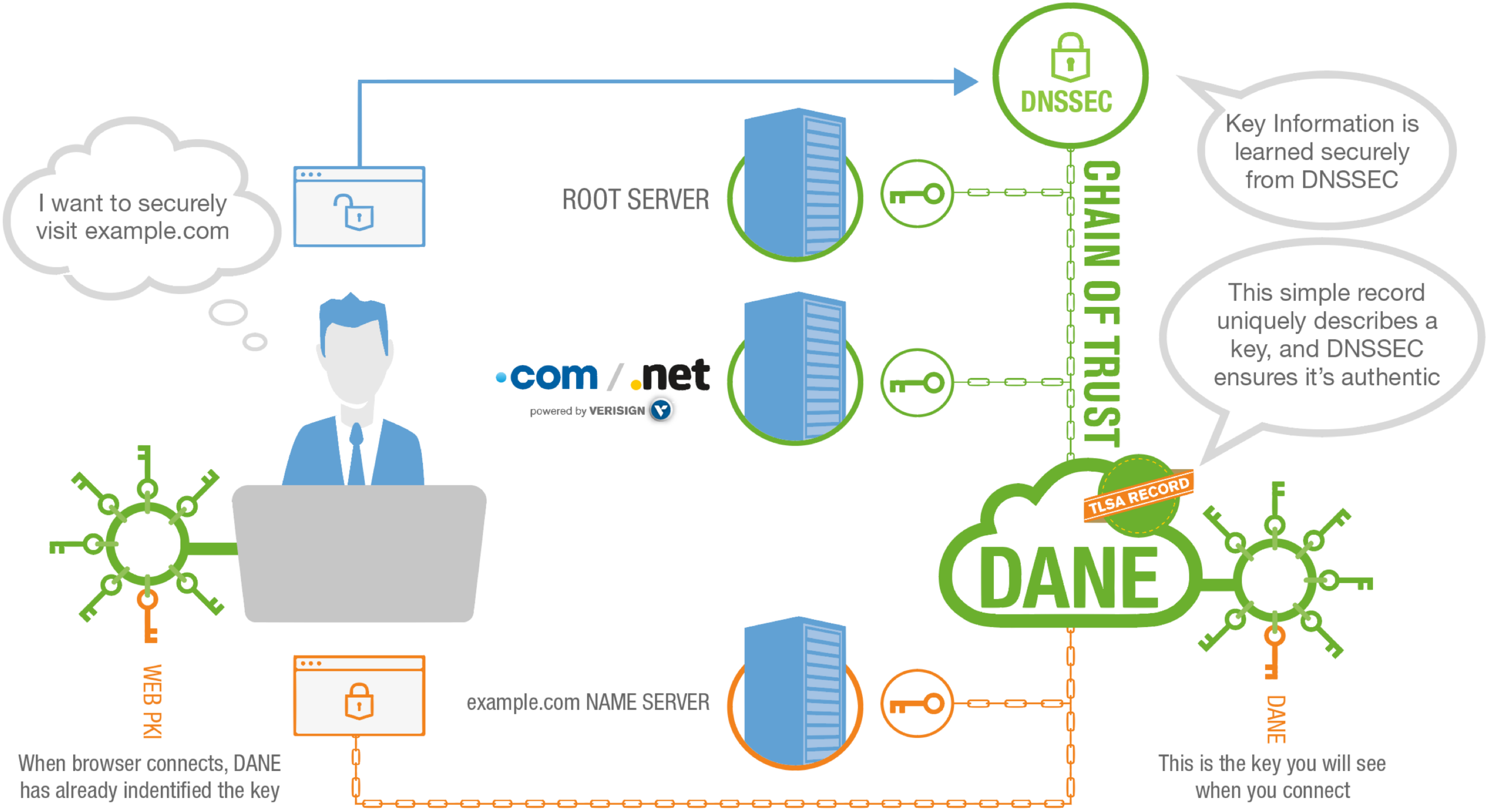


Key Information is learned securely from DNSSEC

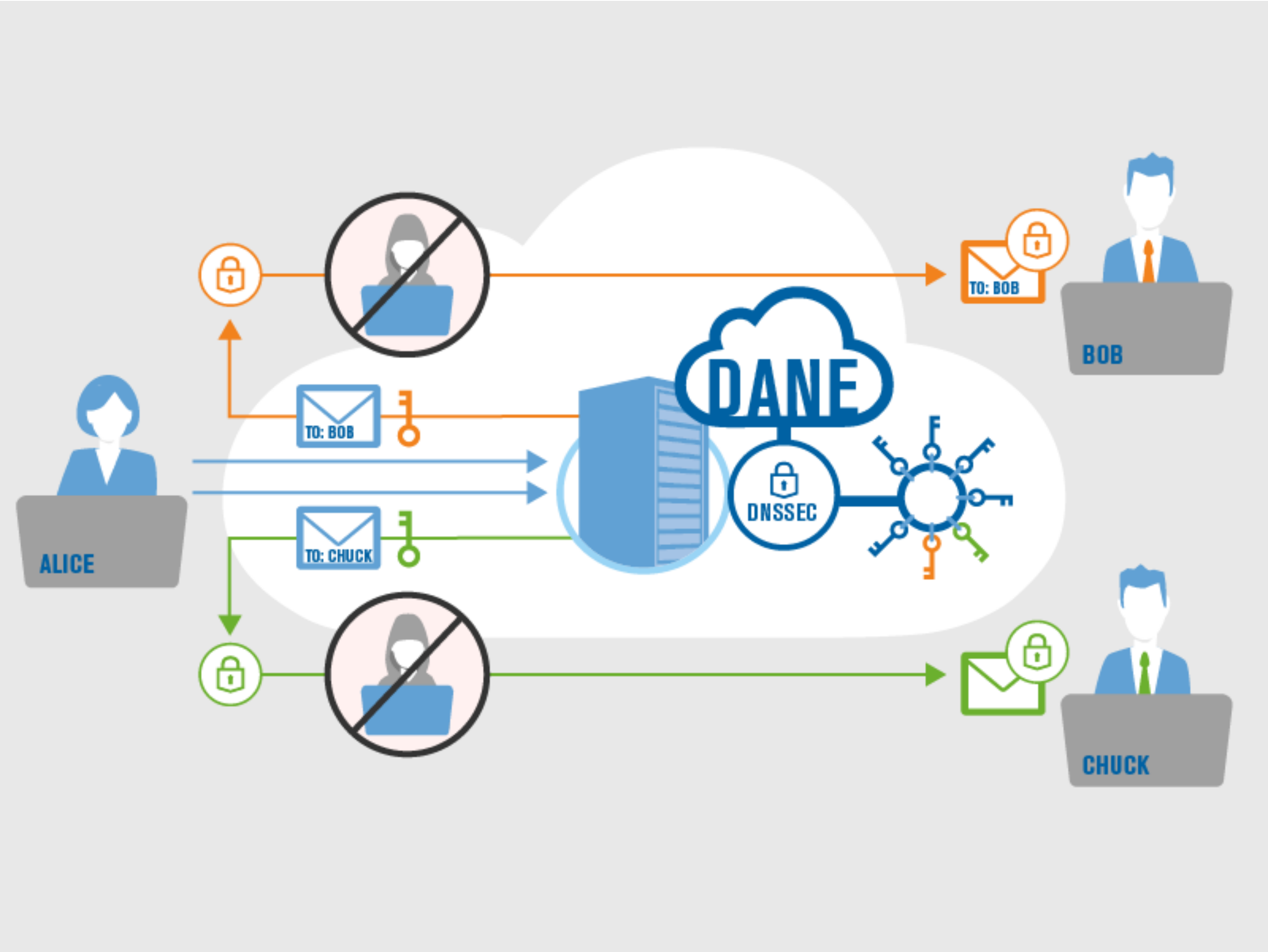
This simple record uniquely describes a key, and DNSSEC ensures it's authentic

This is the key you will see when you connect

Security for TLS, Using CAs and DANE


I want to securely visit example.com
ROOT SERVER
.com / .net
powered by VERISIGN
example.com NAME SERVER
DNSSEC
CHAIN OF TRUST
Key Information is learned securely from DNSSEC
This simple record uniquely describes a key, and DNSSEC ensures it's authentic
TLSA RECORD
DANE
WEB PKI
When browser connects, DANE has already indentified the key
DANE
This is the key you will see when you connect

Security for email, using DANE



Without DANE, we have used the WebPKI

- Applications have needed to trust a large number of global Certification Authorities (CA)
- No namespace constraints! Any CA can issue certificates for any entity on the Internet
  - “An attack on one defeats all” [2]
  - Least common denominator security: our collective security is equal to the weakest one!
- Furthermore, many of them issue subordinate CA certificates to their customers, again with no naming constraints

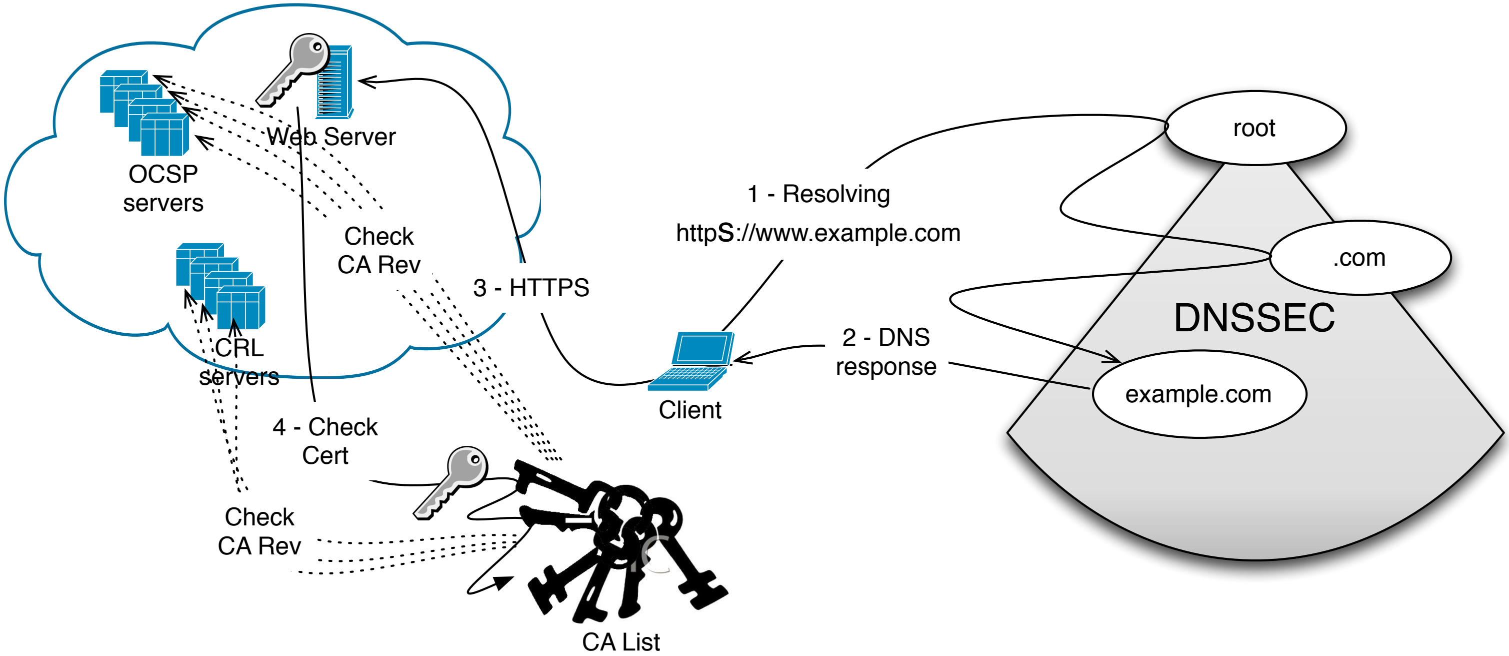
[2] Osterweil, Eric, Burt Kaliski, Matt Larson, and Danny McPherson. "Reducing the X. 509 Attack Surface with DNSSEC’s DANE." SATIN: Securing and Trusting Internet Names (March 2012) (2012).

WebPKI model issues

- “Analysis of the HTTPS Certificate Ecosystem”, UMich, October 2013, Internet Measurement Conference
  - http://conferences.sigcomm.org/imc/2013/papers/imc257-durumericAemb.pdf
  - Over 1,800 separate CAs are capable of issuing certificates for anyone! (Root CAs and intermediate CAs issued by them)

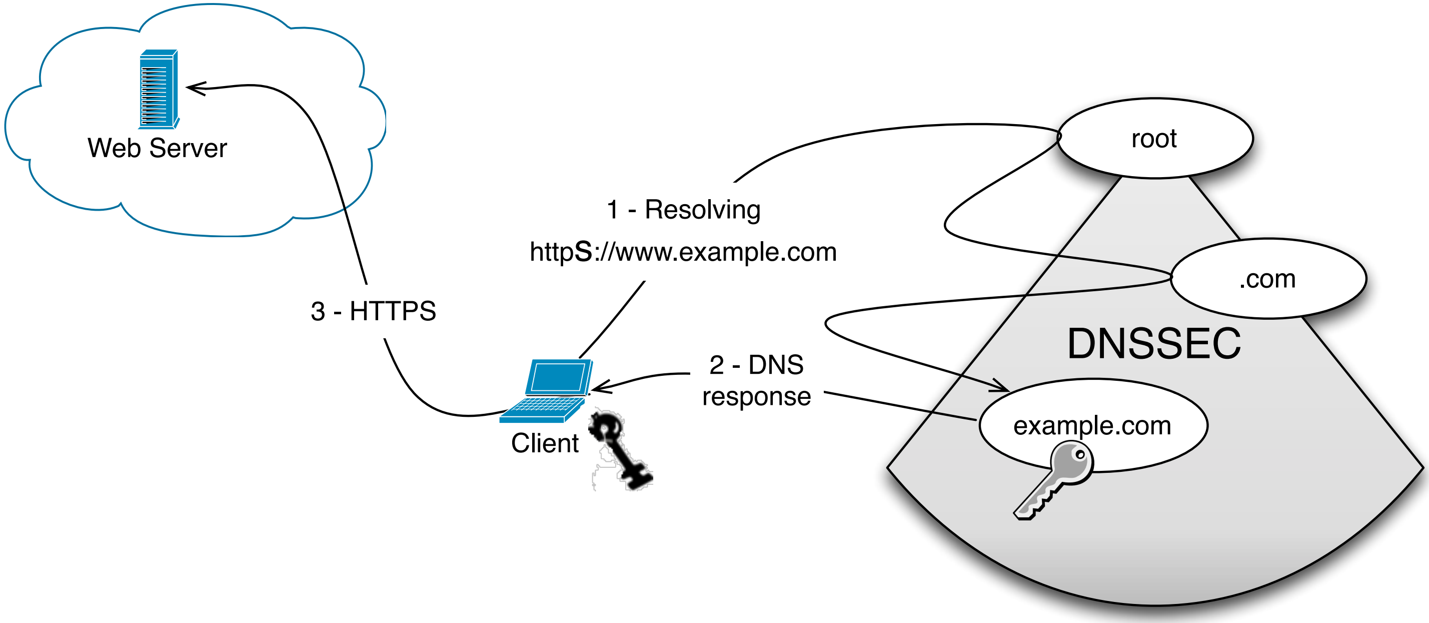
- “The Shape & Size of Threats: Defining a Networked System’s Attack Surface”
  - Eric Osterweil (Verisign), Danny McPherson (Verisign), Lixia Zhang (UCLA), NPsec 2014 best paper

WebPKI Verification



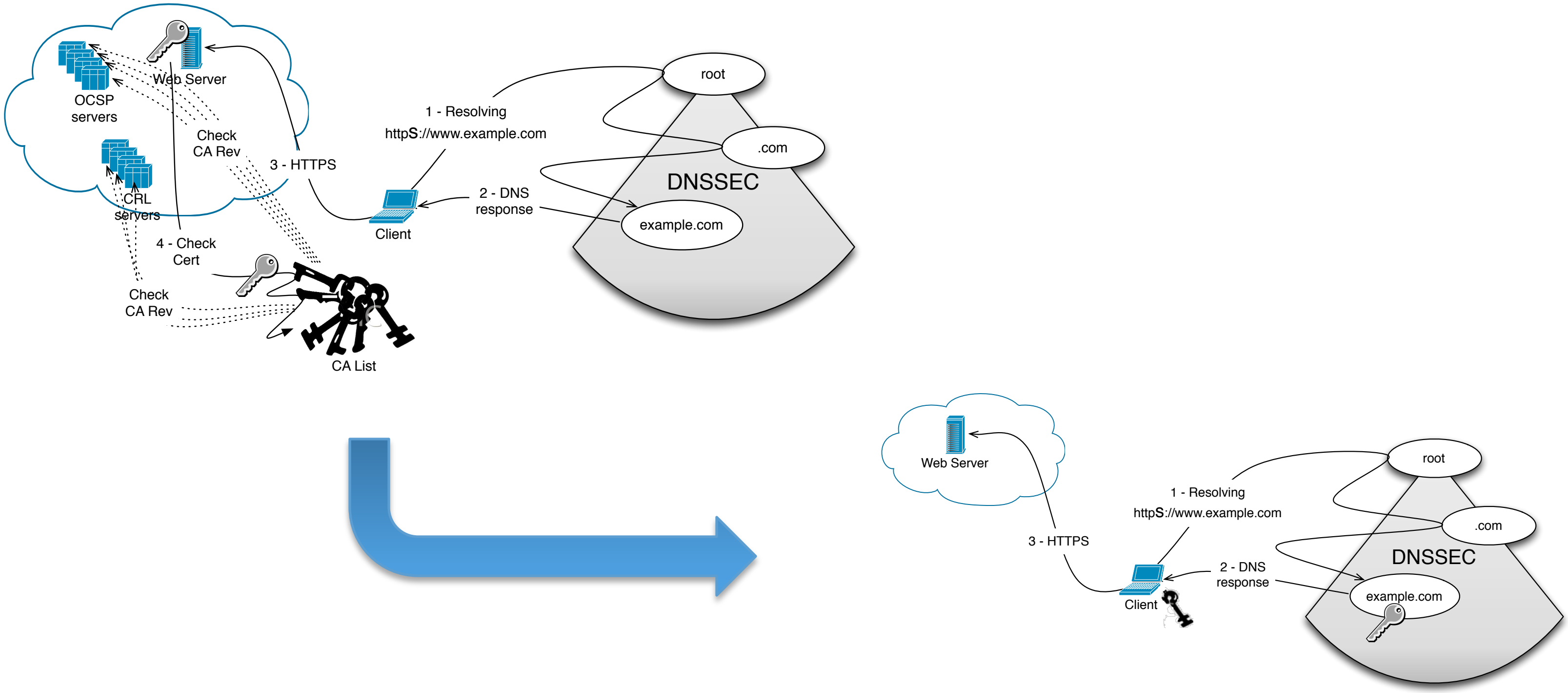
- Transport Layer Security (TLS) needs to be bootstrapped by cryptographic keys
- CA verification uses a set of globally trusted authorities who can *each* vouch for *any* certificate's authenticity
    - Certificates represent previous verification: contain signatures from CAs, and point to revocation points for status checks

DANE verification process



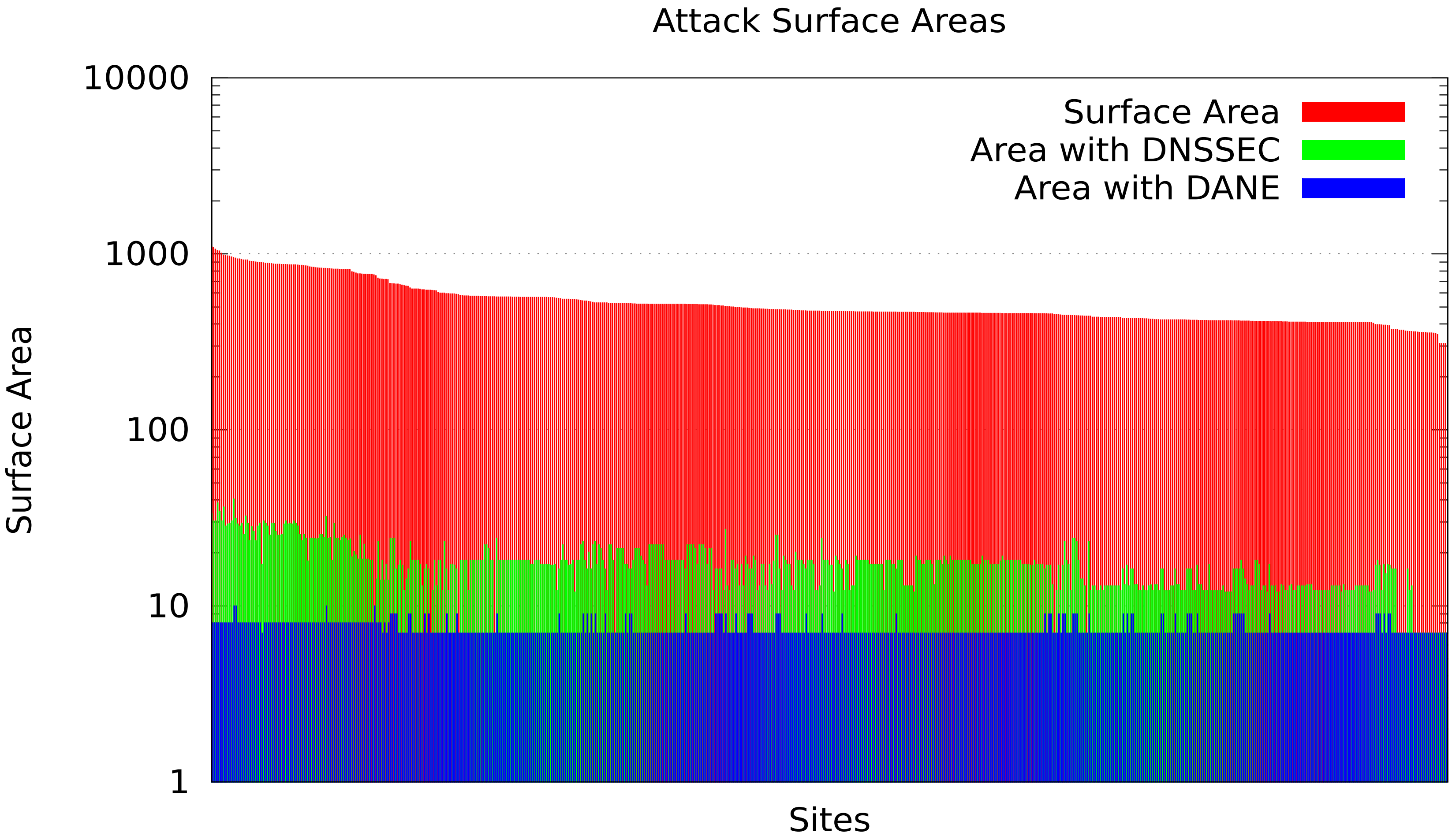
- DNS-Based Authentication of Named Entities (DANE)
  - IETF working group, and standards track RFC for TLS
- DNS zones have TLSA record(s) that uniquely authorize cert used by web servers

Look at what we just cut out…



- Qualitatively, a picture is worth 1,000 words: we can *see* that the attack surface is reduced
- By cutting out our WebPKI check and revocation checks, we removed a lot of moving parts

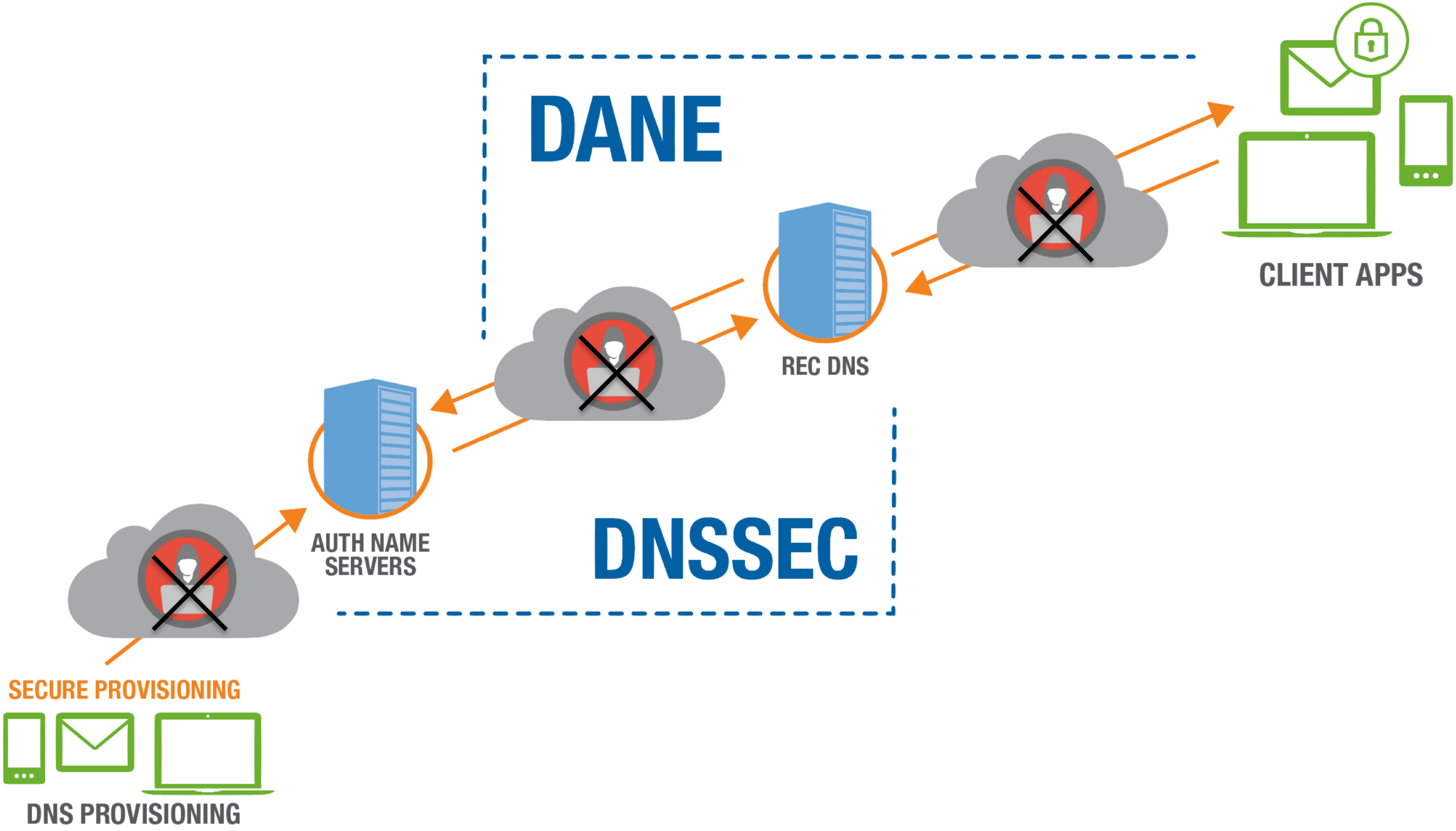
Count what we just cut out (Alexa 1,000) [3]…



[3] *Osterweil, Eric, Danny McPherson, and Lixia Zhang. "The Shape and Size of Threats: Defining a Networked System's Attack Surface." In Network Protocols (ICNP), 2014 IEEE 22nd International Conference on, pp. 636-641. IEEE, 2014.*

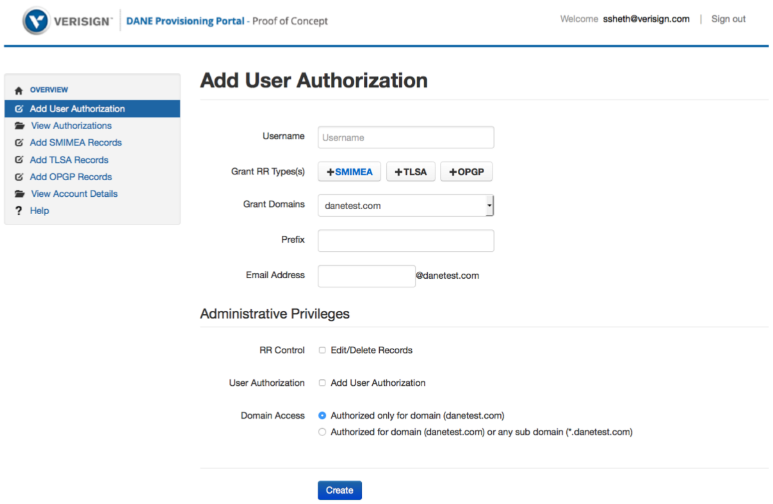
DANE Protocols and Tools

What it takes for DANE to work



Secure Provisioning: A Proof of Concept Portal

- Free provisioning web UI and REST API
- Limited RR types (DANE focused)
- Users can change their keys without affecting parent zone



Add User Authorization

Username: Username

Grant RR Types(s): +SMIMEA +TLSA +OPGP

Grant Domains: danetest.com

Prefix:

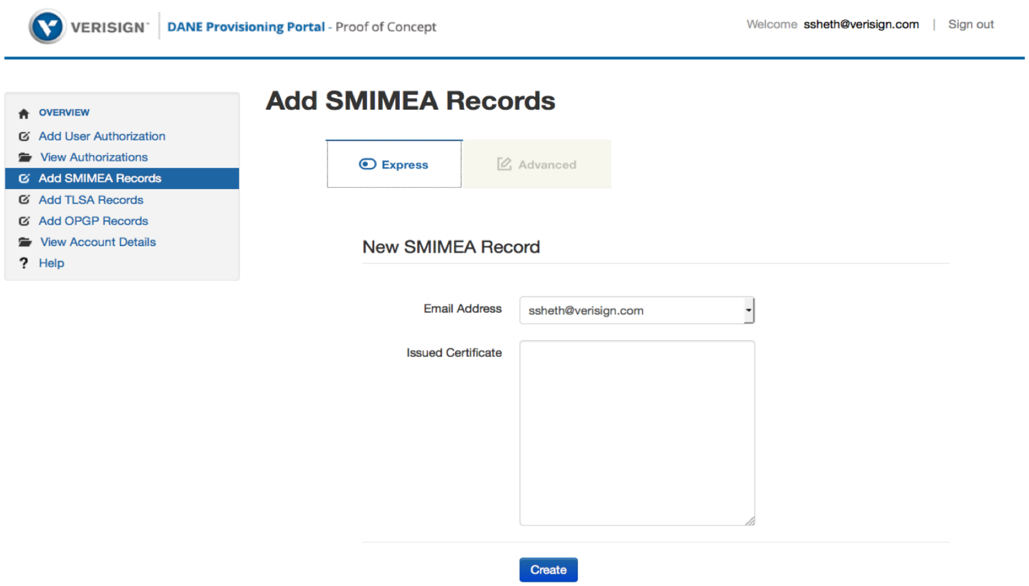
Email Address: @danetest.com

Administrative Privileges

RR Control: □ Edit/Delete Records

User Authorization: □ Add User Authorization

Domain Access: ● Authorized only for domain (danetest.com)
○ Authorized for domain (danetest.com) or any sub domain (*.danetest.com)



Add SMIMEA Records

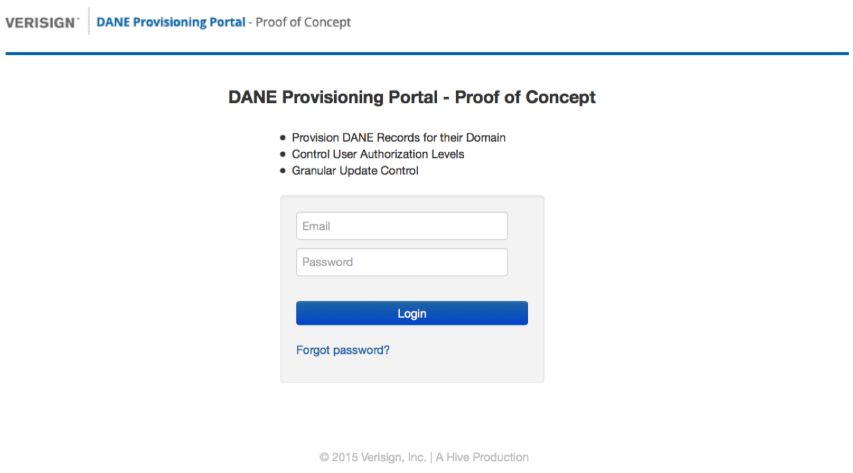
New SMIMEA Record

Email Address: ssheth@verisign.com

Issued Certificate:

Experimental Service to encourage adoption

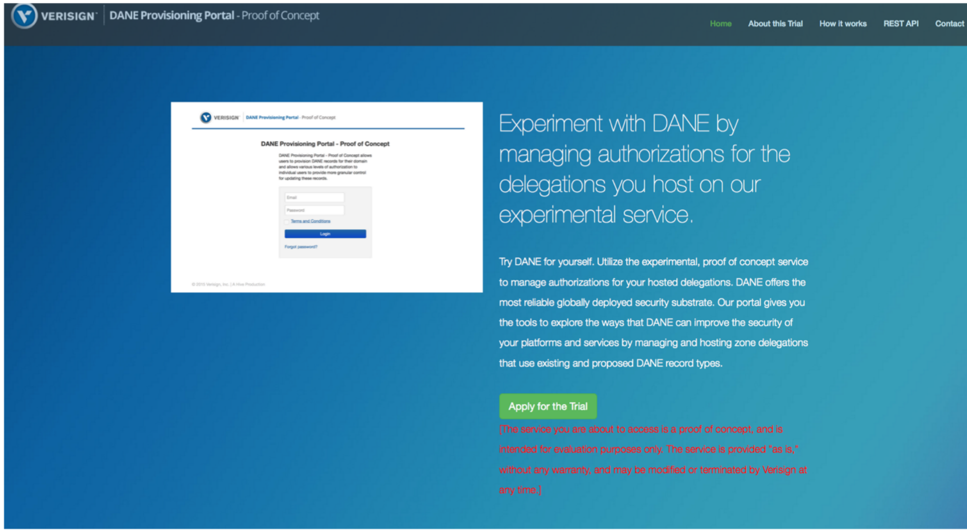
[Provisioning Portal](#) [Provisioning Portal Documentation](#)



DANE Provisioning Portal - Proof of Concept

- Provision DANE Records for their Domain
- Control User Authorization Levels
- Granular Update Control

© 2015 Verisign, Inc. | A Hive Production



DANE Provisioning Portal - Proof of Concept

DANE Provisioning Portal - Proof of Concept allows users to provision DANE records for their domain and allows various levels of authorization to individual users to provide more granular control for updating these records.

© 2015 Verisign, Inc. | A Hive Production

Experiment with DANE by managing authorizations for the delegations you host on our experimental service.

Try DANE for yourself. Utilize the experimental, proof of concept service to manage authorizations for your hosted delegations. DANE offers the most reliable globally deployed security substrate. Our portal gives you the tools to explore the ways that DANE can improve the security of your platforms and services by managing and hosting zone delegations that use existing and proposed DANE record types.

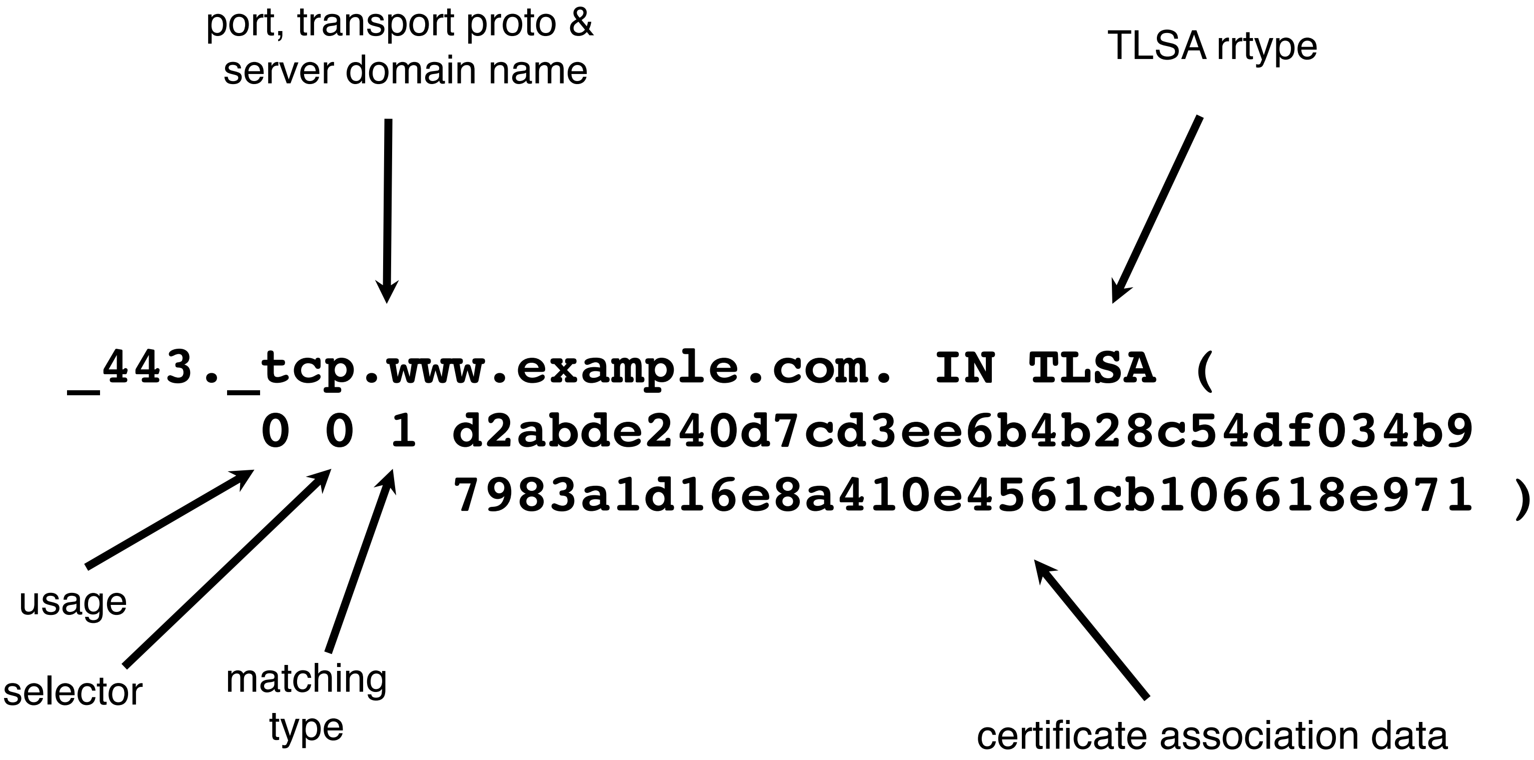
[The service you are about to access is a proof of concept, and is intended for evaluation purposes only. The service is provided "as is," without any warranty, and may be modified or terminated by Verisign at any time.]

https://www.dane-provisioning.verisignlabs.com

DANE for TLS

- RFC 6698: The **DNS-based Authentication of Named Entities (DANE)** Protocol for Transport Layer Security
- http://tools.ietf.org/html/rfc6698
- Defines a new DNS record type "**TLSA**", that can be used for better & more secure ways to authenticate SSL/TLS certificates
  - By specifying constraints on which CA can vouch for a certificate, or which specific PKIX end-entity certificate is valid
  - By specifying that a service certificate or a CA can be directly authenticated in the DNS itself.

TLSA record example



port, transport proto & server domain name

TLSA rrtype

```
_443._tcp.www.example.com. IN TLSA (
      0 0 1 d2abde240d7cd3ee6b4b28c54df034b9
            7983a1d16e8a410e4561cb106618e971 )
```

usage

selector

matching type

certificate association data

TLSA configuration parameters

Usage field:

0	PKIX-TA: CA Constraint
1	PKIX-EE: Service Certificate Constraint
2	DANE-TA: Trust Anchor Assertion
3	DANE-EE: Domain Issued Certificate

Selector field:

0	Match full certificate
1	Match only SubjectPublicKeyInfo

Matching type field:

0	Exact match on selected content
1	SHA-256 hash of selected content
2	SHA-512 hash of selected content

Certificate Association Data: raw cert data in hex

TLSA configuration parameters

Usage field:

```
0    PKIX-TA: CA Constraint
1    PKIX-EE: Service Certificate Constraint
2    DANE-TA: Trust Anchor Assertion
3    DANE-EE: Domain Issued Certificate
```

Co-exists with and Strengthens Public CA system

Operation without Public CAs

Selector field:

```
0    Match full certificate
1    Match only SubjectPublicKeyInfo
```

Matching type field:

```
0    Exact match on selected content
1    SHA-256 hash of selected content
2    SHA-512 hash of selected content
```

`Certificate Association Data: raw cert data in hex`

Usage types

0 **PKIX-TA: CA Constraint**
Specify which CA should be trusted to authenticate the certificate for the service. Full PKIX certificate chain validation needs to be performed.

1 **PKIX-EE: Service Certificate Constraint**
Define which specific service certificate (“EE cert”) should be trusted for the service. Full PKIX cert validation needs to be performed.

2 **DANE-TA: Trust Anchor Assertion**
Specify a domain operated CA which should be trusted independently to vouch for the service certificate.

3 **DANE-EE: Domain Issued Certificate**
Define a specific service certificate for the service at this domain name.

Example TLSA record (for WWW)

- Uses Public CA
- Covers full cert
- Encodes a hash

```
_443._tcp.fedoraproject.org. 263 IN TLSA 0 0 1 (
            19400BE5B7A31FB733917700789D2F0A2471C0C9D506
            C0E504C06C16D7CB17C0 )

_443._tcp.fedoraproject.org. 263 IN RRSIG TLSA 5 4 300 (
            20141114150617 20141015150617 7725
fedoraproject.org.
            hrk0si7I/BWTz0wEtMcFZNUCj/0o5796k5FVuZx6eXrc
            YOe/ChHA/Shu/WHr3iM1yNGi86+8t4wMq9GA+JZthWZC
            ZmENxf9OTNe/t/LBAc2EDW/fMBJq0JO2b4ZkJHXCEyX0
            CDsIYz8shZ20nPGlrsYqwLdQiCeravWcwcJiPuc= )
```

Usage 0 (“CA Constraint”) – this record says:
- For service at fedoraproject.org tcp port 443
- only the CA with the specified SHA-256 certificate fingerprint (19400BE5B…) should be trusted

DANE/TLSA tools and software

- TLSA Record Generation
  - Command line tools: “tlsagen” (in libsmaug), “swede”, “hash-slinger”, “ldns-dane”
  - Web based tool: https://www.huque.com/bin/gen_tlsa

- TLSA validators for web
  - Some 3rd party validator plugins are available (Firefox, Chrome, Opera, Safari):
  - https://www.dnssec-validator.cz/
  - http://blog.huque.com/2014/02/dnssec-dane-tlsa-browser-addons.html
  - Bloodhound Mozilla fork:
  - https://www.dnssec-tools.org/wiki/index.php/Bloodhound

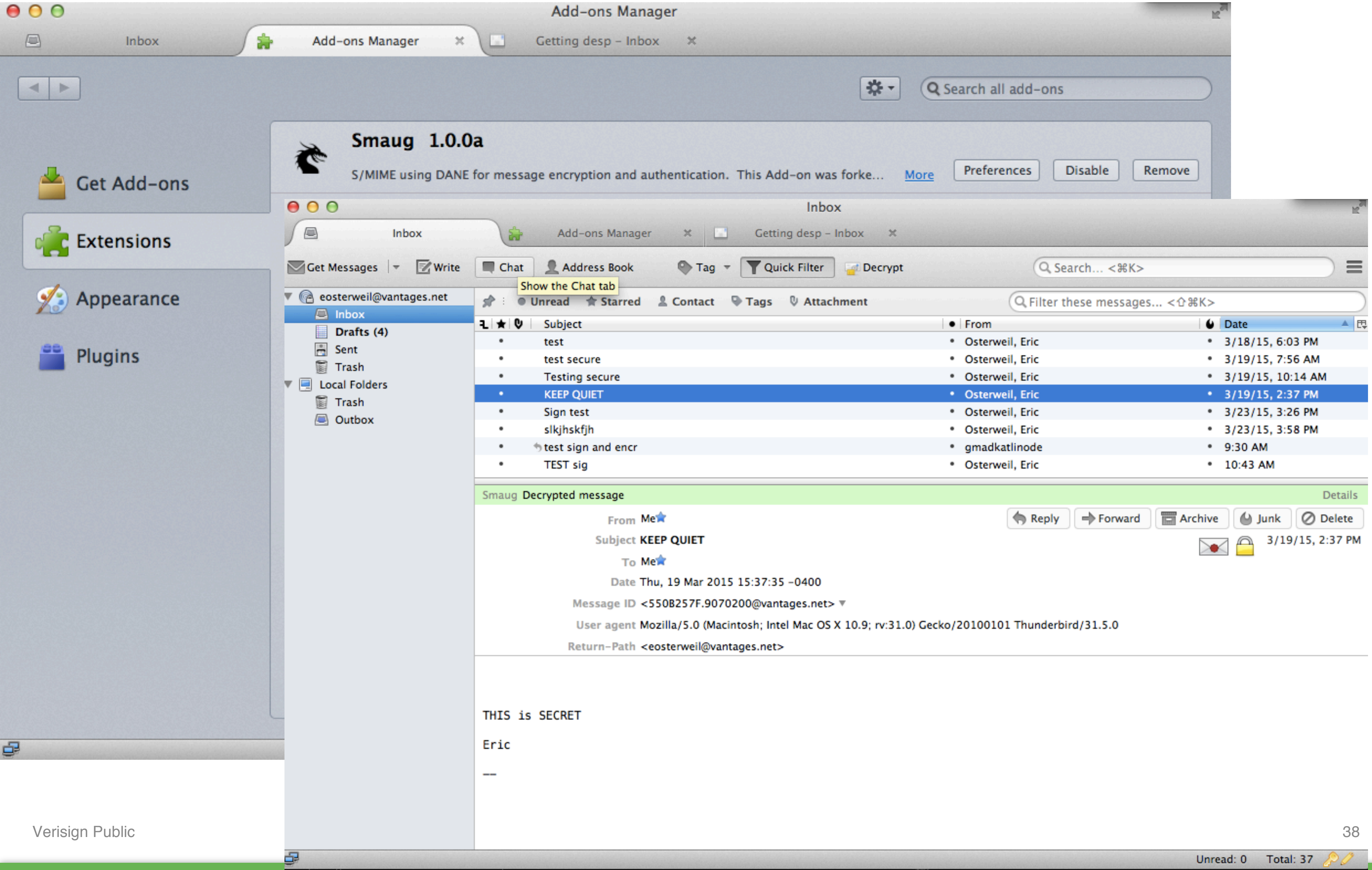
SMIMEA

- Using DNSSEC to associate certificates with domain names for S/MIME
  - https://tools.ietf.org/html/draft-ietf-dane-smime
- S/MIME is a method of encrypted and signing MIME data used in email messages
- The SMIMEA DNS record proposes to associate S/MIME certificates with DNS domain names
- Verisign DANE/SMIMEA early Mail User Agent Prototype
  - https://tools.ietf.org/agenda/92/slides/slides-92-dane-2.pdf
  - https://buenosaires53.icann.org/en/schedule/wed-dnssec/presentation-dnssec-dane-tools-24jun15-en.pdf

Object Security via S/MIME (libsmaug)

- libsmaug leverages DANE for object security in applications
  - Email is just *one* use of S/MIME
- libsmaug optionally uses full featured resolver
- Implementation
  - Open source
  - C/C++
  - https://github.com/verisign/smaug and https://github.com/verisign/smaug-tbird-plugin

Thunderbird Add-on



Smaug 1.0.0a

S/MIME using DANE for message encryption and authentication. This Add-on was forke…

Subject	From	Date
test	Osterweil, Eric	3/18/15, 6:03 PM
test secure	Osterweil, Eric	3/19/15, 7:56 AM
Testing secure	Osterweil, Eric	3/19/15, 10:14 AM
KEEP QUIET	Osterweil, Eric	3/19/15, 2:37 PM
Sign test	Osterweil, Eric	3/23/15, 3:26 PM
slkjhskfjh	Osterweil, Eric	3/23/15, 3:58 PM
test sign and encr	gmadkatlinode	9:30 AM
TEST sig	Osterweil, Eric	10:43 AM

Smaug Decrypted message

From Me

Subject KEEP QUIET

To Me

Date Thu, 19 Mar 2015 15:37:35 -0400

Message ID <550B257F.9070200@vantages.net>

User agent Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:31.0) Gecko/20100101 Thunderbird/31.5.0

Return-Path <eosterweil@vantages.net>

3/19/15, 2:37 PM

```
THIS is SECRET

Eric

--
```

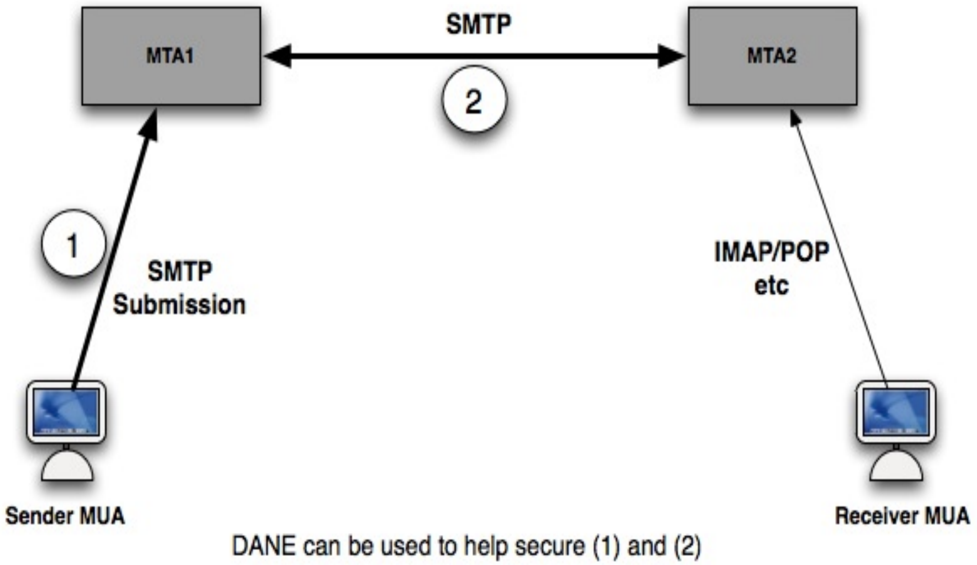
DANE for SMTP

- Connections between SMTP servers today can use TLS encryption opportunistically
- Even when encryption is used, it is vulnerable to attack:
  - Attackers can strip away the TLS capability
  - TLS certificates are often unauthenticated (self signed certificates)

- DANE can address both these vulnerabilities
  - Authenticate the certificate using a DNSSEC signed TLSA record
  - Use the presence of the TLSA record as an indicator that encryption must be performed (prevent downgrade)
  - http://tools.ietf.org/html/draft-ietf-dane-smtp-with-dane

DANE for SMTP

- SMTP over TLS, or SMTP + STARTTLS



DANE can be used to help secure (1) and (2)

- DANE can authenticate TLS for the SMTP connection between the mail server and the user’s mail client (MUA)

- DANE can authenticate TLS connections between SMTP servers (“MTA”s or Mail Transfer Agents)

Example TLSA record (for SMTP)

```
_25._tcp.mx1.freebsd.org. 2389 IN TLSA 3 0 1 (
             5EC0508C3F337D18509F41BFF9D8AB07FED588A132FA
             12FA1E223BA6B9403ACB )

_25._tcp.mx1.freebsd.org. 2389 IN RRSIG  TLSA 8 5 3600 (
             20141023072418 20141009105807 39939
freebsd.org.
             ll6DEQ7oP2lbEcOeJyPk+I8tYiGz4CzuDiqiMbr4Mzp3
             90UWdej3kdAz4t+1BT0dO3/o0nz0pp3HFsDu+gkwT6YH
             Jg4C6mi3STPciCP1tjbFuW/dv4lPkCUaN7kJt/qwPrR6
             0kQmyvcuUoYgUDPbNYbJNJXai+mFai5WqLS2MEP15ydU
             nt8KympnjHS5mVLVGXW0e7tLY1afQz1VrIeYsGW8YztM
             DYUpCXjWiq+YpCFv7rZ7ICejQR6ot1M35CDsfjk68eu0
             EAjx+HlqaTdGyilcMB+GduFwqkULDPIgiFu/3xb+srJR
             zuR89YpHga9OCnz6nXJgQ6cxvSImZWbKuw== )
```

- Uses End Entity cert
- Covers full cert
- Encodes a hash

This is a domain-issued certificate (usage 3), which can be authenticated without a trusted CA.

Large adopters of SMTP + DANE

- Roughly 400 .com domains
  - us-core.com
  - omc-mail.com
  - five-ten-sg.com
- Quite a few are large email systems in Germany.
  - posteo.de
  - mailbox.org
  - umbkw.de
  - bund.de
  - denic.de
  - freebsd.org
  - debian.org, debian.net
  - ietf.org
  - nlnetlabs.nl
  - nic.cz
  - nic.ch
  - torproject.org

SMTP servers that support DANE

- Postfix MTA (works today, version 2.11 onwards)
- Exim (currently under development)

```
Quick start for Postfix:

  postconf -e "smtpd_use_tls = yes”
  postconf -e "smtp_dns_support_level = dnssec”
  postconf -e ”smtp_tls_security_level = dane”
```

Jabber / IM servers

- XMPP (Jabber) has seen some uptake of DANE.
- To authenticate the c2s and/or s2s portion of the XMPP protocol
- List of XMPP servers with DANE TLSA records:
  - https://xmpp.net/reports.php#dnssecdane

Example:

```
_xmpp-server._tcp.mail.de. 3600    IN  SRV  10 20 5269 jabber.mail.de.

_5269._tcp.jabber.mail.de. 600 IN  TLSA    3 1 1 (
                         A0315F0CF61CAC787140833C2C608550476
                         246DDA54122D66BB339D5 0FBB10E3 )
```

- Uses End Entity cert
- Covers just the SPKI
- Encodes a hash

OpenPGPKEY

- OPENPGPKEY record
- Used to publish an OpenPGP public keys in the DNS
- DNSSEC signature provides authentication
- Spec under development, but RR code already assigned
  - https://tools.ietf.org/html/draft-ietf-dane-openpgpkey

Example OPENPGPKEY record

sha256(username)[0:28]._openpgpkey.<domain>

e.g. for shuque@huque.com

1st label: sha256-hash(“shuque”) truncated to 28 octets =
adcd5698c7fc6c44e65e893ab7e84a638db4910d04e8e53314e8a101

2nd label: “_openpgpkey”

Remaining labels: domain portion of email address:
huque.com

Resulting record looks like this:

adcd5698c7fc6c44e65e893ab7e84a638db4910d04e8e53314e8a101.
_openpgpkey.huque.com. IN OPENPGPKEY <base64 encoding of
the openpgp key>

The promise of DANE

- Providing security to data in motion and data at rest
- Secure resting data objects
  - SMIMEA / OPENPGPKEY can secure email, documents, etc.

- and -

- Securing that secure data while in flight
  - TLSA secures TLS sessions: HTTPS, inter-SMTP, etc.

So, where are we today

- DANE has one proposed standard
  - TLSA, RFC 6698
  - There is a growing toolset
  - Mainly operational in inter-SMTP mail security

- Draft standards for email encryption and signing
  - SMIMEA
  - There is an open source library (libsmaug), a pilot MUA support in Thunderbird, and a DNS zone management portal

  - OPENPGPKEY
  - There is an open source toolkits (libsmaug and hashslinger)

Coordinated efforts include…

- National Cybersecurity Center of Excellence (NCCoE) just announced a building block: DNS-Based Secure Email
  - NCCoE is a public-private collaborative FFRDC focused on the implementation and practice of Internet security
  - Vendors work together within NCCoE to “build modular end-to-end reference designs”
  - Call for comments through August 14
  - Call for interest to become a vendor partner (US and non-US)
  - See: https://nccoe.nist.gov/dnssecuredemail

- The Internet Society (ISOC) has a deployment program called Deploy 360
  - http://www.internetsociety.org/deploy360/resources/dane/

A glimpse into the future

- Imagine a future where you can send anyone encrypted email, and they can verify it came from you

- Imagine a future where connected to web servers can be encrypted, *and* we don't have WebPKI vulnerabilities

- In the future, DANE will give us (the users) true end-to-end security powered by

VERISIGN™

© 2015 VeriSign, Inc. All rights reserved. VERISIGN and other trademarks, service marks, and designs are registered or unregistered trademarks of VeriSign, Inc. and its subsidiaries in the United States and in foreign countries. All other trademarks are property of their respective owners.