SecDispatch
Internet-Draft
Intended status: Informational
Expires: September 6, 2018

Y. Nir
Dell EMC
T. Fossati
Nokia
Y. Sheffer
Intuit
T. Eckert
Huawei

March 5, 2018

Considerations For Using Short Term Certificates
draft-nir-saag-star-01

Abstract

Recently there has been renewed interest in an old idea: Issue certificates with short validity periods and forego revocation processing, reasoning that expiration is a sufficient replacement for revocation as long as that expiration is not too far off.

This document covers considerations, both security and operational, for using such Short Term Auto Renewed (STAR) certificates for various scenarios where Using a revocation protocol is considered inappropriate.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Certificates ([RFC5280]) are used in multiple protocols such as the Internet Key Exchange (IKEv2-[RFC7296]) and the Transport Layer Security protocol (TLS-[RFC5246]). Certificates are used to authenticate communicating parties to each other. Certificates are

issued by Certificate Authorities (CAs) to End Entities (EE) to be used to authenticate them to Relying Parties (RPs) in security protocols. Systems that use secure communications typically include certificate authorities, end entities and relying parties, with some nodes in the network having more than one of these roles.

When deploying a system involving secure communications, one of the challenges is how to deal with compromise of an End Entity's private key. The standard way of dealing with this is adding a protocol layer for revocation such as CRLs ([RFC5280]) or OCSP ([RFC6960]).

Such revocation protocols have drawbacks. Although caching of CRLs and OCSP responses is allowed, each setup of a secure channel may require accessing the CRL distribution point (DP) or the OCSP responder. This is both time consuming and provides the system with a few more modes of failure. Assuring reliability of the revocation service increases the cost, and overcoming the latency issue requires changes to the security protocols. All other things being equal, a system that includes revocation checking is more complex and less reliable than a system that does not include it.

For these reasons it is attractive to forego revocation checking. Some deployed systems do this by either eliminating the CRL DP and OCSP extensions from the certificates, or ignoring network and timeout errors in fetching revocation information. Both practices reduce the security of the system.

An alternative solution to the revocation problem is to issue certificates with a short validity period and forego revocation checking. Normally certificates are issued with a validity period of between a few months and a few years. With a shorter validity period if the private key is compromised the potential for abuse is lower because the certificate and its private key expire within a short period of time - a few hours to a few days.

The rest of this document describes operational and security considerations with using short term certificates.

1.1. Conventions Used in This Document

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

Throughout this document we will use the term DP to denote a server for revocation information, either a CRL distribution point or an OCSP Responder. For our purposes they are the same.

We use the term longevity for the period of time between certificate issuance and the time of its expiration as indicated in the notAfter field of the certificate. Note that issuance time may be different from the notBefore field in the certificate.

The text describes end entities as renewing their certificates because the usual operational model for certificates is one of "pull": end entities create certificate requests and send them to CAs for signature. Some systems are designed around a "push" operation where either the CA or a management function generates a new certificate and installs it on the end entity. The text in the document uses pull terminology, but is equally relevant for a push design.

2. Short Term Auto Renewed Certificates

Short term certificates are like any other [RFC5280] certificates except that the period of time between their issuance and their notAfter date is relatively short. Whereas normally certificates are issued for a period of time between a few months and a few years, short term certificates usually expire after a few hours, a few days, or at a limit a couple of weeks.

The certificates discussed in this document have neither a CRL DP extension nor an OCSP authorityInformationAccess extension. In other words such certificates cannot be revoked. Instead, they are valid until they expire.

Automatic certificate renewal is getting ever more popular with enrollment protocols such as EST ([RFC7030]) or ACME ([I-D.ietf-acme-acme]). For short term certificates automatic renewal is essential as a human cannot be expected to flawlessly perform a manual renewal every few days or hours. This document does not recommend any particular automatic renewal method, but Section 4.4 recommends that some such method be used. Automatic renewal processing can roll over the keys from one certificate to its successor, or it can generate new keys with each Certificate generation. As revocation may not exist, multiple certificates for the same EE may be valid at any given time.

The solution for revocation in this scheme is to stop the automatic renewal. The existing compromised certificate will remain valid until it expires. See the considerations in Section 5.1 about revocation.

[Topalovic] describes the design of a system involving STAR certificates for the web, and analyzes its security and efficacy. It

concludes that STAR certificates can be as secure as certificates with OCSP revocation.

2.1. Alternative Design: OCSP Stapling

Relying parties can also avoid the need for contacting the DP at connection setup by having the End Entities implement OCSP stapling. This feature has the EEs rather than the RPs retrieve the OCSP response and send it as part of the protocol. OCSP stapling is described for TLS in [RFC6961] and [RFC6066], and for IKE in [RFC4806].

STAR has several advantages over OCSP stapling:

o A CA that only signs certificates is simpler than a CA that both signs certificates and issues OCSP responses. In fact, a CA for STAR does not need to keep any record of issued certificates.

o A system that does not use CRLs or OCSP need not have an always-available DP for delivering those CRLs or OCSP responses. This reduces both complexity and attack surface.

o OCSP stapling in TLS versions prior to 1.3 works only for the server as end entity. There was no provision for sending the OCSP response for a client certificate in the protocol.

2.2. The Case For Foregoing Revocation

When explaining PKI to people, it is hard to justify why the CA or a delegate needs to both sign blob-1 (the certificate) and also sign blob-2 (the CRL or OCSP response) to tell relying parties that blob-1 is still valid. Surely one signed blob should be enough.

The explanation that we come up with is that traditionally issuing a certificate required human intervention, while the revocation checking object could be issued automatically and at great frequency. So blob-1 would have to be valid for long enough to not over-burden the human charged with maintaining them, while blob-2 could be re-issued frequently.

This explanation no longer holds up. While the initial certificate enrollment may need to be initiated by a human, protocols exist today that make certificate renewal just as automated as CRL issuance. Certificates can be just as frequently issued as CRLs were in the past. The added complexity is no longer needed.

In real systems such as the web relying parties or end entities cache revocation objects as long as it's allowed. If a CRL has a

nextUpdate field that is 4 days in the future, a typical system will not attempt to fetch a new one before those 4 days have elapsed. For this reason, moving to STAR certificates provides a similar level of security to what is generally practiced on the web.

3. Use Cases

This section lists some use cases where STAR certificates seem to be more appropriate than long-lived certificates with revocation checking. The purpose of this section is only motivational. None of the following sections are intended to be a definition of the use case or the standard by which future documents or implementations will be measured for sufficiency.

3.1. Data Center Network Hosts

TBA

3.2. Distributed System Installed in One Or More Data Centers

This is a system installed in multiple hosts in one or more data centers that fulfills some task and requires mutual authentication of its components. An example of such a system is a Storage Area Network (SAN).

3.2.1. Distributed Network Security Functions

This example of a distributed system is multiple network security functions (NSF) [RFC8192] where the SDN controller needs to authenticate the NSFs with which it communicates, and some NSFs need to communicate with each other.

3.3. Certificate Delegation for Content Delivery Networks

TBA

3.4. Autonomic Networking Infrastructure

The Autonomic Network Infrastructure (see [I-D.ietf-anima-reference-model]) is an IETF ANIMA Working Group developed network system architecture to provide the foundation for both future "autonomic networks" (AN), as well as the infrastructure to enable zero-touch secure bootstrapping of domain-wide PKI certificates for network equipment (BRSKI, see [I-D.ietf-anima-bootstrapping-keyinfra]) as well as the set-up of a zero-touch, secure communications fabric for management of existing networks (ACP, see [I-D.ietf-anima-autonomic-control-plane], especially in the context of evolving SDN control & management (see

[I-D.ietf-anima-stable-connectivity]) . These domain certificates are furthermore meant to be reuseable across all network services between network equipment in that domain, therefore allowing to eliminate the need for per-service crypto management (IGP, multicast, BGP, netconf/COPS/radius connections,...).

Overall, the PKI related functions of ANI intend to increase proliferation of PKI security through simplification, achieved through automation and making solutions more resilient by minimizing managed component requirements. CRL or OCSP introduce another set of servers/services that needs to be managed/automated/distributed. The connectivity requirement to such servers and/or the grace periods during which connectivity to them is not required introduce more complex system/security design parameters.

With ANI/ACP/BRSKI, renewal of certificates is fully automated and therefore shorter lifetimes of certificates can easily be used to avoid the additional need for CRL/OCSP. The limitation on reducing certificate lifetimes is only the desired maximum length of time during which connectivity to a CA for renewal may not exist - and the maximum renewal rate of certificates that can be supported by those CA.

Because of the ACP, connectivity to the CA is also more resilient against network/provisioning/configuration problems than network without an ACP. Lastly, the whole ANI is built and maintained autonomously without the need of any configurations except for one or more seed-nodes that perform an expanded version of a PKI Registration Authority.

4. Operational Considerations

The motivations for using short-term certificates are operational. We don't want the latency introduced by fetching the CRL from the DP; we don't want the cost of making the DP 99.999% reliable, and we don't want the cost of making the network paths from all RPs to the DP always available.

Deploying short term certificates comes with its own set of operational considerations, and some of these are enumerated in the following sub-sections.

4.1. Certificate Lifetime and Renewal Schedule

Since we do not assume the CA to be close to 100% available it makes sense for End Entities to renew their certificates well in advance. While the security considerations in Section 5.2 set an upper limit on the longevity of a STAR certificate, operational necessity sets

the frequency of renewal. It is necessary to strike a balance between renewing too often which leads to increased load on the CA and renewing too seldom which increases the risk of having the certificate expire while either the CA or the End Entity are down.

Individual system properties play a significant role here. Systems where both the CA and the EEs are expected to be up all of the time absent a fault may choose to renew a day or even an hour before expiration, while systems with nodes that are only up infrequently and for short periods of time may choose to renew the certificates whenever the EEs happen to be up.

As a general rule of thumb for systems where the CA is mostly available it makes sense for the EE to make the first attempt to renew its certificate about half-way through its lifetime. If that attempt fails because the CA is not available an EE SHOULD retry at regular intervals until it succeeds. Shortly before expiration, the EE SHOULD increase the frequency of retires.

For example, suppose a STAR certificate is issued for 8 days. The EE will first attempt to renew the certificate 4 days before expiration. If that fails it will retry every three hours until only six hours are left before expiration. At that point it will increase the frequency and retry every five minutes. If this is part of the system design, at this point it should also alert the user that something is wrong.

4.2. Availability of the Certificate Authority

While the STAR design does not require 99.999% availability, the CA does need to be available for renewing certificates. Downtimes of more than a quarter of the certificate longevity SHOULD NOT happen. For most modern hardware this is entirely possible even without exotic clustering solutions, but when configuring the system administrators should consider that the longevity of the certificates constrains the required availability of the CA.

When setting the longevity for certificates administrators SHOULD consider how long it takes to recover from a failure of the CA. That length of time can be seconds with a good clustering solution, but can span hours or days without one, especially if the fault happens at a bad time. A failure of a CA should be considered a conceivable occurrence, and longevity should be set so that such a failure does not lead to expiration and outage.

Conversely, if short longevity is required by security targets, the CA should be made more reliable with clustering solutions.

4.3. Clock Skew and the notBefore Field

Despite NTP ([RFC5905]) being over thirty years old and implemented in every major operating system clock skew is a fact of life and many deployed systems don't have the right time. It is also not possible to just mandate the use of NTP because the systems that use STAR certificates are often installed on hosts and networks where NTP is either not configured or blocked. We cannot assume that these systems can enable NTP at will.

Skewed clocks have always been a problem for certificates. Because STAR certificates are always just a few days or hours from expiration they are more sensitive to clock skew. A sufficiently skewed clock can cause three different disfunctions and for STAR certificate such disfunction happens with considerably less skew than with long term certificates:

- A valid certificate may be rejected as not yet valid if the current system time is earlier than its notBefore time. Fortunately this issue can be safely mitigated by setting the notBefore field to a time earlier than the time of issuance.

- A valid certificate may be rejected as expired if the current system time is later than its notAfter time. As long as the clock skew is not too great this is solved by a sensible renewal policy. If as in the example in Section 4.1 the certificate is renewed 4 days before expiration or within a few hours after that, a clock skew of up to 3 days will not be a problem.

- An expired certificate may be accepted if the current system time is earlier than its notAfter time. This is a security issue that is discussed in Section 5.3.

There are several common modes of clock skew:

- The system that doesn't have its clock set at all. These systems might be set to January 1st, 1970 or to some date that was interesting for the hardware vendor. Such systems are incompatible with certificates and MUST NOT be used for STAR certificates.

- The system has its timezone set wrong, and the system time was set so that local time looks good. This limits the clock skew to 24 hours and is generally workable.

- A system that has the time set right but the date set wrong. These are also not usable with certificates.

- A system that was set to the correct time once but has since drifted away. Computer hardware varies wildly between systems with quartz clocks that drift only a few seconds a month and systems that can lose or gain minutes a day. The former are quite usable, the latter are not.

Because of the prevalence of systems with a relatively small skew it is RECOMMENDED to set the notBefore field to a time 72 hours before the actual issuance date.

End Entities MUST NOT use expired certificates and Relying Parties SHOULD alert whenever an expired certificate is presented. This will help the users keep their host clocks set or encourage them to enable NTP.

4.4. Automatic Renewal

Automatic enrollment and renewal is recommended for any system using certificates. While it is possible to renew certificates manually on time, even organizations with the best of IT departments occasionally miss this: [cert-expires]

With short term certificates, this becomes even more important. Renewing a certificate manually every few days or hours is extremely labor intensive, especially when the system contains hundreds, thousands or more end entities, and the risk of outages becomes a certainty.

This document does not mandate any particular enrollment or renewal mechanism. Any of a myriad of standard and proprietary methods can be used and systems with proprietary methods have been shipping for years. The IETF is in the process of standardizing the ACME protocol for enrollment and renewal ([I-D.ietf-acme-acme]) and an extension is proposed to make it more suitable for STAR certificates ([I-D.ietf-acme-star]). The ANI as described in Section 3.4 is a complete zero touch system design providing and relying on automatic certificate renewal.

4.5. Secure (Re-)Enrollments

When short lived certificates expire, automatic re-enrollment can further help to provide survivable, resilient PKI security. Traditionally, initial enrollments, even with otherwise automated solutions such as EST ([RFC7030]) required a manual interaction, or else the device had to perform TOFU (Trust On First Use) to be automatically enrolled. TOFU is even more problematic for re-enrollments and becomes more problematic, the shorter lived certificates and/or trust anchors are. Consider the risk where

during re-enrollment, the device may already be fully configured and could be taken over by an attacker just having to wait for a short lived certificate device certificate or trust anchor to expire. Or consider devices auto-resetting themselves to factory conditions to avoid this problem and then not having to be re-enrolled, but also be re-configured - in the absence of fully zero-touch provisioning solutions.

ANIs BRSKI protocol ([I-D.ietf-anima-bootstrapping-keyinfra], which introduces extensions to EST), and NetConf Zero Touch ([I-D.ietf-netconf-zerotouch] allow fully automated enrollment and re-enrollment of device certificate and trust anchors through the use of "vouchers" ([I-D.ietf-anima-voucher]). These are new digital artifacts that allow enrolling devices to securely trust domains to (re-)enroll them. They work by providing a signed statement by a representative of the manufacturer of the device, that the device with a specified identity (e.g: IDevID) should trust a particular domain - identified by an initial trust anchor. This allows to overcome the biggest challenge of expired short lived certificates/ trust-anchors.

Furthermore, if the certificate and/or trust anchors are required for security of network connectivity - such as routing protocol security or network layer encryption - to even reach a re-enrollment server, then there is yet another challenge with short lived certificates/ trust-anchors and their higher likelyhood of expiring.

In the case of ANI, network layer security (e.g.: IPsec) is used for protecting network connectivity including to reach the EST renewal server. When certificate/TA are expired, renewal can not be used. Instead though, automatic re-enrollment can be used, which does not rely on generic network layer security, but instead relies on its own proxy service to provide connectivity for such devices that need to re-enroll. Nevertheless, re-enrollment may be a complex operation due to the potential need to involve the above mentioned representative entity of the manufacturer to generate vouchers.

4.6. Future enhancements for renewal/re-enrollment

One easy improvement that is specifically of interest with the use of short-lived device certificates/trust-anchors is a new interpretation of the lifetime of certificates. Today, there is no clear distinction when or how to apply the lifetime, and in result, it is usually assumed to be applicable to all operations relying on those certificates.

In the case of short-lived certificates, the elements performing certificate renewal/re-enrollment can easily have a different

interpretation of the lifetime and may not rely on what the certificate itself says. This allows to turn re-enrollments into renewals and avoid possible complexities or manual steps potentially required for re-enrollment (depending on the system used).

In the case of BRSKI/EST, there is only one TLS connection used for renewal and/or re-enrollment and expiry affects the certificates used on this TLS connection. The server uses EST for renewal or the extended signaling of BRSKI for re-enrollment. When a device with expired, short-lived certificate connects to the BRSKI/EST server, this server could allow to perform only simple EST renewal instead of re-enrollment with a voucher by simply considering the lifetime of the presented (and expired) device certificate to be extended.

This type of re-interpretation requires primarily some generic work to allow this type of interpretation - and then per-solution work to leverage this interpretation. In the case of BRSKI/EST for example, devices would simply use their expired domain certificate to authenticate themselves and perform certificate renewal - instead of using their IDevID and trying to re-enroll (which is a more complex operation with potentially external dependenices against the manufacturer component).

4.7. Certificate Transparency

Certificate Transparency (CT), [RFC6962] is about keeping a log of all issued certificates.

A system that issues a certificate every few days to thousands or end entities will create more records for a CT log than a web host that gets one certificate every year.

TBA: Discussion about this.

5. Security Considerations

STAR certificates eliminate an important security feature of PKI which is the ability to revoke certificates. Revocation allows the administrator to limit the damage done by a rogue node or an adversary who has control of the private key. With STAR certificates expiration replaces revocation so there is a timeliness issue.

It should be noted that revocation also has timeliness issues, because both CRLs and OCSP responses have nextUpdate fields that tell RPs how long they should trust this revocation data. These fields are typically set to hours, days, or even weeks in the future. Any revocation that happens before the time in nextUpdate goes unnoticed by the RP.

Section 5.1 discusses the reasons why a certificate would be revoked if revocation was available and how STAR certificates do the same. Section 5.2 discusses considerations for setting the longevity of a certificate, and Section 5.3 discusses how longevity should be adjusted to deal with clock skew.

More discussion of the security of STAR certificates is available in [Topalovic].

5.1. Reasons for Revocation

There are two types of compromise that require administrators to revoke a certificate:

- A host has lost control of the private key. There are many ways that this can happen: a host can be hacked and a file containing the private key may or may not have been copied; a disk may be replaced and the old one has not been securely disposed of; a fault causes the private key to be erased. In all these cases we would like to revoke the certificate to make sure an adversary cannot use the private key for nefarious purposes. For STAR certificates the only solution is to wait for the certificate to expire and the system is vulnerable until that happens. Longevity should be set so that this risk is acceptable.

- A host may begin doing unintended things, either due to a software fault or due to a malicious takeover. Again without revocation RPs will continue to trust this node until its certificate expires.

When a node "goes rogue" or an adversary gets control of the private key it is important to block renewal or these certificates or else the attack can persist forever. No matter how short-term these short term certificates are, there is a certain window of time when the attacker can use the certificate. This can often be mitigated with application-level measures.

With most systems relying parties are configured with the names of nodes with which they are allowed to communicate. When revocation is not available changing the configuration so that the rogue node cannot connect is RECOMMENDED. This is useful even when revocation is available because timeliness issues are common to both revocation and expiration.

5.2. Longevity and Revocation

There is always a period of time between when a compromise is discovered and when RPs stop trusting the certificate. With revocation this has to do with the time it takes to process the revocation and the span of time between the thisUpdate and nextUpdate fields. With STAR certificates this is controlled by the time it takes to inhibit renewals and the longevity of the certificates.

For this reason it makes sense to set the longevity to a period of time similar to the span of time that we would set for the CRL or OCSP updates. Typically a few days is an appropriate time. For some cases this can be as low as a few hours. Setting the renewal time too short may cause operational problems as discussed in Section 4.3 and Section 4.2. In general longevity should not be set shorter than the availability of the CA allows.

Fortunately modern hardware is powerful enough and reliable enough that even a system with tens of thousands of end entities with longevity of 1-2 days should not suffer an outage because of expired certificates.

5.3. Clock Skew and Security

As discussed in Section 4.3 clock skew can lead to expired certificates being treated as valid. While even the use of NTP may leave clocks with a few seconds of inaccuracy, all installations MUST take steps to limit the clock skew on their hosts.

An upper bound for the amount of skew allowed for hosts in a particular system is one of the parameters for such a system. For systems using NTP this can be 2 seconds. For systems where the clocks are set manually, this tends to be far greater, but without an upper bound no guarantees can be made about the security of certificate use.

This upper bound is also a limit on the target certificate longevity. For example, if hosts and CAs can each have a clock skew of 24 hours then it is impossible to achieve a longevity of under 48 hours. With a reasonable skew and a reasonable target longevity we can achieve our security targets by reducing the certificate longevity by twice the upper bound for skew. So if skew is bounded by 24 hours (the bad timezone case) and target longevity is 7 days, it makes sense to set the longevity on the CA to 5 days.

5.4. CA availability

A successful Denial of Service (DoS) attack against a CA prevents it from issuing certificates. With short-term certificates this could quickly lead to outages as certificates expire.

The important period of time here is the time between when the EE first attempts to renew the certificate and the time that the certificate expires. For example, if the EE attempts to renew the certificates a mere five minutes before expiration, then a five-minute CA outage can lead to an invalid certificate and failed connections.

This issue is no different from DoS attacks against the DP for certificates with revocation. The methods of protection are also similar:

- Certificate renewal should first be attempted plenty of time in advance as recommended in Section 4.1. This will leave enough time for administrators to deal with the attack.

- As for all important infrastructure, network defenses SHOULD be deployed to mitigate DoS attacks.

6. IANA Considerations

There are no requests to IANA in this document.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[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, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

7.2. Informative References

[cert-expires]
Lennon, M., "Google Lets SMTP Certificate Expire", April 2015, <http://www.securityweek.com/google-lets-smtp-certificate-expire>.

[I-D.ietf-acme-acme]
Barnes, R., Hoffman-Andrews, J., and J. Kasten, "Automatic Certificate Management Environment (ACME)", draft-ietf-acme-acme-07 (work in progress), June 2017.

[I-D.ietf-acme-star]
Sheffer, Y., Lopez, D., Gonzalez de Dios, O., Pastor Perales, A., and T. Fossati, "Use of Short-Term, Automatically-Renewed (STAR) Certificates to Delegate Authority over Web Sites", draft-ietf-acme-acme-07 (work in progress), June 2017.

[I-D.ietf-anima-autonomic-control-plane]
Eckert, T., Behringer, M., and S. Bjarnason, "An Autonomic Control Plane (ACP)", draft-ietf-anima-autonomic-control-plane-13 (work in progress), December 2017.

[I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M., Behringer, M., Bjarnason, S., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-11 (work in progress), February 2018.

[I-D.ietf-anima-reference-model]
Behringer, M., Carpenter, B., Eckert, T., Ciavaglia, L., Pierre, P., Liu, B., Nobre, J., and J. Strassner, "A Reference Model for Autonomic Networking", draft-ietf-anima-reference-model-05 (work in progress), October 2017.

[I-D.ietf-anima-stable-connectivity]
Eckert, T. and M. Behringer, "Using Autonomic Control Plane for Stable Connectivity of Network OAM", draft-ietf-anima-stable-connectivity-10 (work in progress), February 2018.

[I-D.ietf-anima-voucher]
Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "Voucher Profile for Bootstrapping Protocols", draft-ietf-anima-voucher-07 (work in progress), January 2018.

[I-D.ietf-netconf-zerotouch]
Watsen, K., Abrahamsson, M., and I. Farrer, "Zero Touch Provisioning for Networking Devices", draft-ietf-netconf-zerotouch-20 (work in progress), February 2018.

[RFC4806] Myers, M. and H. Tschofenig, "Online Certificate Status Protocol (OCSP) Extensions to IKEv2", RFC 4806, DOI 10.17487/RFC4806, February 2007, <https://www.rfc-editor.org/info/rfc4806>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <https://www.rfc-editor.org/info/rfc6066>.

[RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, DOI 10.17487/RFC6960, June 2013, <https://www.rfc-editor.org/info/rfc6960>.

[RFC6961] Pettersen, Y., "The Transport Layer Security (TLS) Multiple Certificate Status Request Extension", RFC 6961, DOI 10.17487/RFC6961, June 2013, <https://www.rfc-editor.org/info/rfc6961>.

[RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, DOI 10.17487/RFC6962, June 2013, <https://www.rfc-editor.org/info/rfc6962>.

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <https://www.rfc-editor.org/info/rfc7030>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[RFC8192] Hares, S., Lopez, D., Zarny, M., Jacquenet, C., Kumar, R., and J. Jeong, "Interface to Network Security Functions (I2NSF): Problem Statement and Use Cases", RFC 8192, DOI 10.17487/RFC8192, July 2017, <https://www.rfc-editor.org/info/rfc8192>.

[Topalovic] Topalovic, E., Saeta, B., Huang, L., Jackson, C., and D. Boneh, "Towards Short-Lived Certificates", 2012, <http://www.w2spconf.com/2012/papers/w2sp12-final9.pdf>.

Authors' Addresses

Yoav Nir
Dell EMC
9 Andrei Sakharov St
Haifa 3190500
Israel

EMail: ynir.ietf@gmail.com

Thomas Fossati
Nokia

EMail: thomas.fossati@nokia.com

Yaron Sheffer
Intuit

EMail: yaronf.ietf@gmail.com

Toerless Eckert
Huawei USA - Futurewei Technologies Inc.
2330 Central Expy
Santa Clara 95050
USA

EMail: tte+ietf@cs.fau.de