

Internet Architecture Board  
Internet-Draft  
Intended status: Informational  
Expires: May 4, 2017

R. Housley  
Vigil Security  
K. O'Donoghue  
Internet Society  
October 31, 2016

Improving the Public Key Infrastructure (PKI) for the World Wide Web  
draft-iab-web-pki-problems-05

## Abstract

The Public Key Infrastructure (PKI) used for the World Wide Web (Web PKI) is a vital component of trust in the Internet. In recent years, there have been a number of improvements made to this infrastructure, including improved certificate status checking, automation, and transparency of governance. However, additional improvements are necessary. This document identifies continuing areas of concern and provides recommendations to the Internet community for additional improvements, moving toward a more robust and secure Web PKI.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 4, 2017.

## Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

Internet-Draft

Web PKI Problems

October 2016

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">2</a>
<a href="#">2.</a>	<a href="#">A Brief Description of the Web PKI . . . . .</a>	<a href="#">3</a>
<a href="#">3.</a>	<a href="#">Improvements to the Web PKI . . . . .</a>	<a href="#">4</a>
<a href="#">3.1.</a>	<a href="#">Strong Cryptography . . . . .</a>	<a href="#">4</a>
<a href="#">3.1.1.</a>	<a href="#">Preparing for Quantum Computers . . . . .</a>	<a href="#">4</a>
<a href="#">3.1.2.</a>	<a href="#">Avoiding Weak Cryptography . . . . .</a>	<a href="#">5</a>
<a href="#">3.2.</a>	<a href="#">Support for Enterprise PKIs . . . . .</a>	<a href="#">6</a>
<a href="#">3.3.</a>	<a href="#">Web PKI in the Home . . . . .</a>	<a href="#">8</a>
<a href="#">3.4.</a>	<a href="#">Governance Improvements to the Web PKI . . . . .</a>	<a href="#">10</a>
<a href="#">4.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">12</a>
<a href="#">5.</a>	<a href="#">Privacy Considerations . . . . .</a>	<a href="#">12</a>
<a href="#">6.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">12</a>
<a href="#">7.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">13</a>
<a href="#">Appendix A.</a>	<a href="#">Acknowledgements . . . . .</a>	<a href="#">15</a>
<a href="#">Appendix B.</a>	<a href="#">IAB Members at the Time of Approval . . . . .</a>	<a href="#">16</a>
	<a href="#">Authors' Addresses . . . . .</a>	<a href="#">16</a>

## [1.](#) Introduction

The Public Key Infrastructure (PKI) for the World Wide Web (Web PKI) has evolved into a key component of the global Internet; it enables trusted business and individual transactions. This global infrastructure has been growing and evolving for many years. The success of Web PKI has contributed to significant Internet growth. The Web PKI impacts all aspects of our lives, and no one can imagine the web without the protections that the Web PKI enables.

As with any maturing technology, there are several problems with the current Web PKI. The Web PKI makes use of certificates as described in [RFC 5280](#) [[RFC5280](#)]. These certificates are primarily used with Transport Layer Security (TLS) as described in [RFC 5246](#) [[RFC5246](#)].

The economics of the Web PKI value chain are discussed in [[VFBH](#)], [[AV](#)], and [[AVAV](#)]. This document does not investigate the economic issues further, but these economic issues provide motivation for correcting the other problems that are discussed in this document.

One note of caution is that the references above assume the cost of acquiring a certificate is high. These costs have been decreasing in recent years due to a number of factors including the Let's Encrypt initiative discussed later in this document.

Over the years, many technical improvements have been made to the Web PKI, but several challenges remain. This document offers a general set of recommendations to the Internet community designed to be helpful in addressing these remaining challenges.

## 2. A Brief Description of the Web PKI

This section provides a very brief introduction to some of the key concepts of the Web PKI. It is not intended to be a full description of Web PKI but rather to provide some basic concepts to help frame the remaining discussion.

Web PKI is an infrastructure comprised of a number of PKIs that enables the establishment of trust relationships between communicating web entities. This trust may be chained through multiple intermediate parties. The root of that trust is referred to as a trust anchor. A relying party is an entity that depends upon the trust provided by the infrastructure to make informed decisions. A complex set of technical, policy, and legal requirements can make up the qualifications for a trust anchor in a specific situation.

Certificates are digitally signed structures that contain the information required to communicate the trust. Certificates are specified in [RFC 5280](#) [RFC5280]. Certificates contain, among other things, a subject name, a public key, a limited validity lifetime, and the digital signature of the Certification Authority (CA). Certificate users require confidence that the private key associated with the certified public key is owned by the named subject.

The architectural model used in the Web PKI includes:

EE: End Entity -- the subject of a certificate -- certificates are issued to end entities including Web servers and clients that need mutual authentication.

CA: Certification Authority -- the issuer of a certificate --

issues certificates for end entities including Web servers and clients.

RA: Registration Authority -- an optional system to which a CA delegates some management functions such as identity validation or physical credential distribution.

While in its simplest form, the Web PKI is fairly straightforward, there are a number of concepts that can complicate the relationships and the behavior. As mentioned already, there can be intermediate certificates that represent delegation within the certification path. There can be cross-signing of certificates that creates

multidimensional relationships. Browsers install numerous trust anchors associated with many different CAs in the Web PKI. All of this results in a complex ecosystem of trust relationships that reflect different operational practices and underlying certificate policies.

Certificates naturally expire since they contain a validity lifetime. In some situations, a certificate needs to be revoked before it expires. Revocation usually happens because the private key is lost or compromised, but an intermediate CA certificate can be revoked for bad behavior. All CAs are responsible for providing revocation status of the certificates that they issue throughout their lifetime of the certificate. Revocation status information may be provided by certificate revocation lists (CRLs) [[RFC5280](#)], the Online Certificate Status Protocol (OCSP) [[RFC6960](#)], or some other mechanism.

The enrollment process used by a CA makes sure that the subject name in the certificate is appropriate and that the subject actually holds the private key. The enrollment process should require the subject to use the private key; this can be accomplished with PKCS#10 [[RFC2986](#)] or some other proof-of-possession mechanism such as [[RFC6955](#)].

### 3. Improvements to the Web PKI

Over the years, many technical improvements have been made to the Web PKI. Despite this progress, several challenges remain. This section discusses several unresolved problems, and it suggests general directions for tackling them.

### [3.1.](#) Strong Cryptography

Quantum computers [[WIKI-QC](#)] exist today, but they are not yet able to solve real world problems faster than digital computers. No one knows whether a large-scale quantum computer will be invented in the next decade or two that is able to break all of the public key algorithms that are used in the Web PKI, but it seems prudent to prepare for such a catastrophic event.

In the mean time, the Web PKI needs to employ cryptographic algorithms that are secure against known cryptanalytic techniques and advanced digital computers.

#### [3.1.1.](#) Preparing for Quantum Computers

Hash-based signature algorithms [[HASH-S1](#)][HASH-S2] are quantum resistant, meaning that they are secure even if an attacker is able to build a large-scale quantum computer. Hash-based signature

algorithms have small public and private keys, provide fast signing and verification operation, but they have very large signature values and one private key can produce a fixed number of signatures. The number of signatures is set at the time the key pair is generated.

As a result of these properties, hash-based signature algorithms are not ideal for signing certificates. However, they are well suited for other uses, including signatures for software updates. The use of a quantum resistant signature algorithm for software updates ensures that new software can be securely deployed even if a large-scale quantum computer is invented during the lifetime of the system.

Several signature and key establishment algorithms [[WIKI-PQC](#)] are being investigated that might prove to be quantum resistant and offer properties that are suitable for use in the Web PKI. So far, none of these algorithms has achieved wide acceptance. Further research is needed.

While this research is underway, some security protocols allow a pre-shared key (PSK) to be mixed with a symmetric key that is established with a public key algorithm. If the PSK is distributed without the use of a public key mechanism, the overall key establishment

mechanism will be quantum resistant. Consider the use of a PSK for information that requires decades of confidentiality protection, such as health care information.

The Web PKI can prepare for the for quantum computing by:

1. Deploy hash-based signatures for software updates.
2. For information that requires decades of confidentiality protection, mix a pre-shared key (PSK) as part of the key establishment.
3. Continue research on quantum resistant public key cryptography.

### [3.1.2.](#) Avoiding Weak Cryptography

Several digital signature algorithms, one-way hash functions, and public key sizes that were once considered strong are no longer considered adequate. This is not a surprise. Cryptographic algorithms age; they become weaker over time. As new cryptanalysis techniques are developed and computing capabilities increase, the amount of time needed to break a particular cryptographic algorithm will decrease. For this reason, the algorithms and key sizes used in the Web PKI need to migrate over time.

CAs and Browser vendors have been managing algorithm and key size transitions, but it is a significant challenge to maintain a very high degree of interoperability across the world wide web while phasing out aged cryptographic algorithms or too small key sizes. When these appear in a long-lived trust anchor or intermediate CA certificate, refusal to accept them can impact a very large tree of certificates. In addition, if a certificate for a web site with a huge amount of traffic is in that tree, rejecting that certificate may impact too many users.

Despite this situation, the MD5 and SHA-1 one-way hash functions have been almost completely eliminated from the Web PKI, and 1024-bit RSA public keys are essentially gone [[MB2015](#)] [[MB2016](#)]. It took a very long time to make this happen, and trust anchors and certificates that used these cryptographic algorithms were considered valid long

after they were widely known to be too weak.

Obviously, additional algorithm transitions will be needed in the future. The algorithms and key sizes that are acceptable today will become weaker with time. [RFC 7696](#) [[RFC7696](#)] offers some guidelines regarding cryptographic algorithm agility.

The Web PKI can prepare for the next transition by:

1. Having experts periodically evaluate the current choices of algorithm and key size. While it is not possible to predict when a new cryptanalysis technique will be discovered, the end of the useful lifetime of most algorithms and key sizes is known many years in advance.
2. Planning for a smooth and orderly transition from a weak algorithm or key size. Experience has shown that many years are needed produce to specifications, develop implementations, and then deploy replacements.
3. Reducing the lifetime of end-entity certificates to create frequent opportunities to change an algorithm or key size.

### [3.2](#). Support for Enterprise PKIs

Many enterprises operate their own PKI. These enterprises do not want to be part of the traditional Web PKI, but they face many challenges in order to achieve a similar user experience and level of security.

Enterprise PKI users must install one or more enterprise trust anchors in their operating system or browser. There is readily-available software that can install trust anchors for use by the

operating system and browser, but the enterprise PKI will not be trusted until the system administrator or end user does this step.

Enterprise PKI users often experience greater latency than traditional Web PKI users. Standards-based and proprietary revocation status checking approaches might offer relief.

The Status Request extension to TLS [[RFC6066](#)] allows the web server

to provide status information about its certificate. By including this extension in the TLS handshake, the browser asks the web server to provide OCSP responses in addition to the server certificate. This approach greatly reduces the latency since the browser does not need to generate an OCSP request or wait for an OCSP response to check the validity of the server certificate. The inclusion of a time-stamped OCSP response in the TLS handshake is referred to as "OCSP stapling". In addition, the MUST\_STAPLE feature [[TLSFEATURE](#)] can be used to insist that OCSP stapling be used.

While not widely implemented, the Multiple Certificate Status Request extension [[RFC6961](#)] allows the web server to provide status information about its own certificate and also the status of intermediate certificates in the certification path, further reducing latency.

When OCSP stapling is used by an enterprise, the OCSP responder will not receive an enormous volume of OCSP requests because the web servers make a few requests and the responses are passed to the browsers in the TLS handshake. In addition, OCSP stapling can improve user privacy, since the web server, not the browser, contacts the OCSP responder. In this way, the OCSP responder is not able to determine which browsers are checking the validity of certificate for particular websites.

Some browser vendors provide a proprietary revocation checking mechanism that obtains revocation status for the entire Web PKI in a very compact form. This mechanism eliminates latency since no network traffic is generated at the time that a certificate is being validated. However, these mechanisms cover only the trust anchor store for that browser vendor, excluding all enterprise PKIs. In addition, measurements in 2015 [[IMC2015](#)] show that these mechanisms do not currently provide adequate coverage of the Web PKI.

Several enterprises issue certificates to all of their employees, and among other uses, these certificates are used in TLS client authentication. There is not a common way to import the private key and the client certificate into browsers. In fact, the private key can be stored in many different formats depending on the software used to generate the public/private key pair. PKCS#12 [[RFC7292](#)]

seems to be the most popular format at the moment. A standard way to



import the needed keying material and a standard format will make this task much easier, and the web might enjoy an increase in mutual authentication. However, please note the privacy considerations in [Section 5](#).

Enterprise PKIs can be better supported if:

1. Each enterprise PKI offers an OCSP Responder, and enterprise websites make use of OCSP Stapling.
2. Operating system and browser vendors support a standard way to install private keys and certificates for use in client authentication.
3. In the event that browser vendors continue to offer latency-free proprietary revocation status checking mechanisms, then these mechanisms need to expand the coverage to all of the Web PKI and offer a means to include enterprise PKIs in the coverage.

### [3.3](#). Web PKI in the Home

More and more, web protocols are being used to manage devices in the home. For example, homeowners can use a web browser to connect to a web site that is embedded in their home router to adjust various settings. The router allows the browser to access web pages to adjust these setting as long as the connection originates from the home network and the proper password is provided. However, there is no way for the browser to authenticate to the embedded web site. Authentication of the web site is normally performed during the TLS handshake, but the Web PKI is not equipped to issue certificates to home routers or the many other home devices that employ embedded web sites for homeowner management.

A solution in this environment cannot depend on the homeowner to perform duties that are normally associated with a web site administrator. However, some straightforward tasks could be done at the time the device is installed in the home. These tasks cannot be more complex than the initial setup of a new printer in the home, otherwise they will be skipped or done incorrectly.

There are three very different approaches to certificates for home devices that have been discussed over the years. In the first approach, a private key and certificate are installed in the device at the factory. The certificate has an unlimited lifetime. Since it never expires, no homeowner action is needed to renew it. Also, since the certificate never changes, the algorithms are selected by the factory for the lifetime of the device. The subject name in the

certificate is quite generic, as it must be comprised of information that is known in the factory. The subject name is often based on some combination of the manufacturer, model, serial number, and MAC address. While these do uniquely identify the device, they have little meaning to the homeowner. A secure device identifier, as defined in [[IEEE802.1AR](#)], is one example of a specification where locally significant identities can be securely associated with a manufacturer-provisioned device identifier.

In the second approach, like the first one, a private key and a certificate that are installed in the device at the factory, but the homeowner is unaware of them. This factory-installed certificate is used only to authenticate to a CA operated by the manufacturer. At the time the device is installed, the homeowner can provide a portion of the subject name for the device, and the manufacturer CA can issue a certificate that includes a subject name that the homeowner will recognize. The certificate can be renewed without any action by the homeowner at appropriate intervals. Also, following a software update, the algorithms used in the TLS handshake and the certificate can be updated.

In the third approach, which is sometimes used today in Internet of Things devices, the device generates a key pair at the time the device is configured for the home network, and then a controller on the local network issues a certificate for the device that contains the freshly generated public key and a name selected by the user. If the device is passed on to another user, then a new key pair will be generated and a new name can be assigned when the device is configured for that user's network.

[Section 3.1.2](#) of this document calls for the ability to transition from weak cryptographic algorithms over time. For this reason, and the ability to use a subject name that the homeowner will recognize, the second or third approaches are preferred.

One potential problem with the second approach is continuity of operations of the manufacturer CA. After the device is deployed, the manufacturer might go out of business or stop offering CA services, and then come time for renewal of the certificate, there will not be a CA to issue the new certificate. Some people see this as a way to end-of-life old equipment, but the users want to choose the end date, not have one imposed upon them. One possible solution might be modeled on the domain name business, where other parties will continue to provide needed services if the original provider stops doing so.

The Web PKI can prepare for the vast number of home devices that need certificates by:

1. Building upon the work being done in the IETF ACME Working Group [[ACMEWG](#)] to facilitate the automatic renewal of certificates for home devices without any actions by the homeowner beyond the initial device setup.
2. Establish conventions for the names that appear in certificates that accomodate the approaches discussed above and also ensure uniqueness without putting a burden on the homeowner.
3. Working with device manufacturers to establish scalable CAs that will continue to issue certificates for the deployed devices even if the manufacturer goes out of business.
4. Working with device manufacturers to establish OCSP Responders so that the web sites that are embedded in the devices can provide robust authentication and OCSP stapling in a manner that is compatible with traditional web sites.

#### [3.4.](#) Governance Improvements to the Web PKI

As with many other technologies, Web PKI technical issues are tangled up with policy and process issues. Policy and process issues have evolved over time, sometimes eroding confidence and trust in the Web PKI. Governance structures are needed that increase transparency and trust.

Web PKI users are by definition asked to trust CAs. This includes what CAs are trusted to do properly, and what they are trusted not to do. The system for determining which CAs are added to or removed from the trust anchor store in browsers is opaque and confusing to most Web PKI users. The CA/Browser Forum has developed baseline requirements for the management and issuance of certificates [[CAB2014](#)] for individual CAs. However, the process by which an individual CA gets added to the trust anchor store by each of the browser vendors is somewhat mysterious. The individual browser vendors determine what should and should not be trusted by including the CA certificate in their trust anchor store. They do this by reviewing the CA CPS and reports of audits conducted using the CPA Canada WebTrust for Certification Authorities criteria [[WEBTRUST](#)] or

the ETSI EN 319 411 requirements [ESTI]. The WebTrust for CAs program also provides a trust mark for CAs meet all the criteria. Failure to pass an audit can result in the CA being removed from the trust store.

Once the browser has shipped, regular updates may add or delete CAs. This is generally not something that a user would monitor. For an informed user, information about which CAs have been added to or deleted from the browser trust anchor store can be found in the

browser release notes. Users can also examine the policies, practices, and audit reports of the various CAs that have been developed and posted for the WebTrust Program. How does an individual, organization, or enterprise really determine if a particular CA is trustworthy? Do the default choices inherited from the browser vendors truly represent the organization's trust model? What constitutes sufficiently bad behavior by a CA to cause removal from the trust anchor store?

In addition, it can be hazardous for users to remove CAs from the browser trust anchor store. If a user removes a CA from the browser trust anchor store, some web sites may become completely inaccessible or require the user to take explicit action to accept warnings or bypass browser protections related to untrusted certificates.

CAs can be removed from a trust anchor store as part of the maintenance of acceptable CAs. There may be a few very large CAs that are critical to significant portions of the Web PKI. Removing one of these CAs can have a significant impact on a huge number of websites. As discussed in briefly in [Section 4](#), users are already struggling to understand the implications of untrusted certificates, so they often ignore warnings presented by the browser.

There are a number of organizations that play significant roles in the operation of the Web PKI, including the CA/Browser Forum, the WebTrust Task Force, ETSI, and the browser and operating system vendors. These organizations act on behalf of the entire Internet community; therefore, transparency in these operations is fundamental to confidence and trust in the Web PKI. In particular, transparency in both the CA/Browser Forum and the browser vendor processes would be helpful. Recently the CA/Browser Forum made some changes to their operational procedures to make it easier for people to participate

and to improve visibility into their process [[CAB1.2](#)]. This is a significant improvement, but these processes need to continue to evolve in an open, inclusive, and transparent manner. Currently, as the name implies, the CA/Browser Forum members primarily represent CAs and browser vendors. It would be better if relying parties also have a voice in this forum. Additionally, some browser vendors are more transparent in their decision processes than others, and it is felt that all should be more transparent.

Since the Web PKI is widespread, applications beyond the World Wide Web are making use of the Web PKI. For example, the Web PKI is used to secure connections between SMTP servers. In these environments, the browser-centric capabilities are unavailable. The current governance structure does not provide a way for the relying parties in these applications to participate.

The Web PKI governance structures can be made more open and transparent by:

1. Browser vendors providing additional visibility and tools to support the management of the trust anchor store.
2. Governance organizations providing a way for all relying parties, including ones associated with non-browser applications, to participate.

#### [4.](#) Security Considerations

This document considers some areas for improvement of the Web PKI. Some of the risks associated with doing nothing or continuing down the current path are articulated. The Web PKI is a vital component of a trusted Internet, and as such needs to be improved to sustain continued growth of the Internet.

Many users find browser error messages related to certificates confusing. Good man-machine interfaces are always difficult, but in this situation users are unable to fully understand the risks that they are accepting, and as a result they do not make informed decisions about when to proceed and when to stop. This aspect of browser usability has improved over the years, and there is an enormous amount of ongoing work on this complex topic. It is hoped

that further improvements will allow users to make better security choices.

## 5. Privacy Considerations

Client certificates can be used for mutual authentication. While mutual authentication is usually considered better than unilateral authentication, there is a privacy concern in this situation. When mutual authentication is used, the browser sends the client certificate in plaintext to the webserver in the TLS handshake. This allows the browser user's identity to be tracked across many different sites by anyone that can observe the traffic.

## 6. IANA Considerations

None.

{{{ RFC Editor: Please remove this section prior to publication. }}}}

## 7. Informative References

- [ACMEWG] IETF, "Charter for Automated Certificate Management Environment (acme) Working Group", June 2015, <<https://datatracker.ietf.org/doc/charter-ietf-acme/>>.
- [AV] Arnbak, A. and N. van Eijk, "Certificate Authority Collapse: Regulating Systemic Vulnerabilities in the HTTPS Value Chain", 2012 TRPC , August 2012, <<http://dx.doi.org/10.2139/ssrn.2031409>>.
- [AVAV] Asghari, H., van Eeten, M., Arnbak, A., and N. van Eijk, "Security Economics in the HTTPS Value Chain", Workshop on Economics of Information Security (WEIS) 2013 , 2013, <<http://www.econinfosec.org/archive/weis2013/papers/AsghariWEIS2013.pdf>>.
- [CAB1.2] CA/Browser Forum, "Bylaws of the CA/Browser Forum",

October 2014, <<https://cabforum.org/wp-content/uploads/CA-Browser-Forum-Bylaws-v.1.2.pdf>>.

[CAB2014] CA/Browser Forum, "CA/Browser Forum Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates, v.1.2.2", October 2014, <<https://cabforum.org/wp-content/uploads/BRv1.2.2.pdf>>.

[IEEE802.1AR] IEEE Standards Association, "IEEE Standard for Local and Metropolitan Area Networks -- Secure Device Identity", 2009.

[HASH-S1] McGrew, D. and M. Curcio, "Hash-Based Signatures", [draft-mcgrew-hash-sigs-04](#) (work in progress), March 2016.

[HASH-S2] Huelsing, A., Butin, D., Gazdag, S., and A. Mohaisen, "Hash-Based Signatures", [draft-irtf-cfrg-xmss-hash-based-signatures-06](#) (work in progress), July 2016.

[IMC2015] Liu, Y., Tome, W., Zhang, L., Choffnes, D., Levin, D., Maggs, B., Mislove, A., Schulman, A., and C. Wilson, "An End-to-End Measurement of Certificate Revocation in the Web's PKI", October 2015, <<http://conferences2.sigcomm.org/imc/2015/papers/p183.pdf>>.

[MB2015] Wilson, K., "Phase 2: Phasing out Certificates with 1024-bit RSA Keys", January 2015, <<https://blog.mozilla.org/security/2015/01/28/phase-2-phasing-out-certificates-with-1024-bit-rsa-keys/>>.

[MB2016] Barnes, R., "Payment Processors Still Using Weak Crypto", February 2016, <<https://blog.mozilla.org/security/2016/02/24/payment-processors-still-using-weak-crypto/>>.

[RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification

Request Syntax Specification Version 1.7", [RFC 2986](#),  
DOI 10.17487/RFC2986, November 2000,  
<<http://www.rfc-editor.org/info/rfc2986>>.

- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#),  
DOI 10.17487/RFC5246, August 2008,  
<<http://www.rfc-editor.org/info/rfc5246>>.
- [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,  
<<http://www.rfc-editor.org/info/rfc5280>>.
- [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,  
<<http://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,  
<<http://www.rfc-editor.org/info/rfc6961>>.
- [RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS)  
Extensions: Extension Definitions", [RFC 6066](#),  
DOI 10.17487/RFC6066, January 2011,  
<<http://www.rfc-editor.org/info/rfc6066>>.
- [RFC6955] Schaad, J. and H. Prafullchandra, "Diffie-Hellman Proof-  
of-Possession Algorithms", [RFC 6955](#), DOI 10.17487/RFC6955,  
May 2013, <<http://www.rfc-editor.org/info/rfc6955>>.

- [RFC7292] Moriarty, K., Ed., Nystrom, M., Parkinson, S., Rusch, A.,  
and M. Scott, "PKCS #12: Personal Information Exchange  
Syntax v1.1", [RFC 7292](#), DOI 10.17487/RFC7292, July 2014,  
<<http://www.rfc-editor.org/info/rfc7292>>.



- [RFC7696] Housley, R., "Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms", [BCP 201](#), [RFC 7696](#), DOI 10.17487/RFC7696, November 2015, <<http://www.rfc-editor.org/info/rfc7696>>.
- [TLSFEATURE] Hallam-Baker, P., "X.509v3 TLS Feature Extension", [draft-hallambaker-tlsfeature-10](#) (work in progress), July 2015.
- [VFBH] Vratonjic, N., Freudiger, J., Bindschaedler, V., and J. Hubaux, "The Inconvenient Truth About Web Certificates", Workshop on Economics of Information Security (WEIS) 2011 , 2011, <<http://www.econinfosec.org/archive/weis2011/papers/The%20Inconvenient%20Truth%20about%20Web%20Certificates.pdf>>.
- [WEBTRUST] CPA Canada, "WebTrust Program for Certification Authorities", August 2015, <<http://www.webtrust.org/homepage-documents/item27839.aspx>>.
- [WIKI-PQC] Wikipedia, "Post-quantum cryptography", October 2016, <[https://en.wikipedia.org/wiki/Post-quantum\\_cryptography](https://en.wikipedia.org/wiki/Post-quantum_cryptography)>.
- [WIKI-QC] Wikipedia, "Quantum computing", October 2016, <[https://en.wikipedia.org/wiki/Quantum\\_computing](https://en.wikipedia.org/wiki/Quantum_computing)>.

## [Appendix A](#). Acknowledgements

This document has been developed within the IAB Privacy and Security Program. The authors greatly appreciate the review and suggestions provided by Rick Andrews, Mary Barnes, Richard Barnes, Marc Blanchet, Peter Bowen, Alissa Cooper, Nick Doty, Stephen Farrell, Joe Hall, Ted Hardie, Paul Hoffman, Ralph Holz, Lee Howard, Christian Huitema, Eliot Lear, Xing Li, Lucy Lynch, Gervase Markham, Eric Rescorla, Andrei Robachevsky, Thomas Roessler, Jeremy Rowley, Christine Runnegar, Jakob Schlyter, Wendy Seltzer, Dave Thaler, Brian Trammell, and Juan Carlos Zuniga.

[Appendix B](#). IAB Members at the Time of Approval

{{{ RFC Editor: Please add the names to the IAB members at the time that this document is put into the RFC Editor queue. }}}}

Authors' Addresses

Russ Housley  
Vigil Security  
918 Spring Knoll Drive  
Herndon, VA 20170  
USA

Email: housley@vigilsec.com

Karen O'Donoghue  
Internet Society  
1775 Wiehle Ave #201  
Reston, VA 20190  
USA

Email: odonoghue@isoc.org

