

Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: January 9, 2017

R. Salz
Akamai Technologies
July 08, 2016

No MTI Crypto without Public Review
draft-rsalz-drbg-speck-wap-wep-01

Abstract

Cryptography is becoming more important to the IETF and its protocols, and more IETF protocols are using, or looking at, cryptography to increase privacy on the Internet [[RFC7258](#)].

This document specifies a proposed best practice for any mechanism (or data format) that uses cryptography; namely, that RFCs cannot specify an algorithm as mandatory-to-implement (MTI) unless that algorithm has had reasonable public review. This document also "sketches out" a rough definition around what such a review would look like.

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 January 9, 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

Internet-Draft

crypto-pubreview

July 2016

carefully, as they describe your rights and restrictions with respect 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

1.	Terminology	2
2.	Introduction	2
3.	Why is Cryptography Hard?	3
4.	Things to avoid	4
5.	Why limit to MTI?	4
6.	How to Do it Right	5
6.1.	Public Review	6
7.	Acknowledgements	6
8.	References	6
8.1.	Normative References	6
8.2.	Informative References	6
	Author's Address	7

[1.](#) Terminology

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 [RFC 2119](#) [[RFC2119](#)].

The term mandatory to implement (MTI) is used in this document to describe a cryptographic algorithm that is listed as a MUST in an RFC.

The term "snake oil" is used as a pejorative for something which appears to do its job acceptably, but actually does not; see https://en.wikipedia.org/wiki/Snake_oil_%28cryptography%29 . It is a goal of the IETF that we never be misled into being, or mistakenly taken as, snake oil salesman.

[2.](#) Introduction

Cryptography is becoming more important to the IETF and its protocols, and more IETF protocols are using, or looking at, cryptography to increase privacy on the Internet [[RFC7258](#)].

This document specifies a proposed best practice for any protocol (or data format) that uses cryptography. Namely, that such RFCs cannot specify an algorithm as mandatory-to-implement (MTI) unless that algorithm has had reasonable public review. This document also

"sketches out" a rough definition around what such a review would look like.

3. Why is Cryptography Hard?

Cryptography is hard because it is not like traditional IETF protocol deployments. In this classic situation, if one party implements a protocol incorrectly, it usually becomes obvious as interoperability suffers or completely fails. But with cryptography, one party can have implementation defects, or known exploitable weaknesses, that expose the entire communication stream to an attacker. Open source and code reviews are not a panacea here, but using only widely-accepted cryptographic mechanisms (e.g., avoiding facilities like https://en.wikipedia.org/wiki/Dual_EC_DRBG) will reduce the attack surface.

Cryptography is hard because subtle design characteristics can have disastrous consequences. For example, the US Digital Signature Algorithm requires the random nonce to be protected and never re-used. If those requirements are not met, the private key can be leaked.

Cryptography is hard because adversaries design new attacks and refine existing ones. Attacks get better over time; they never get worse. For example, it is now de rigueur to protect against CPU timing attacks, even when the device is only viewable over a network. A recent paper [[acoustic](#)] (XXX reference) can identify a private key if your smartphone is just laid next to an innocuous charging device. We understand power differential attacks, timing attacks, and perhaps cache line attacks; we now have to think about RFI emissions from our phone.

Cryptography is hard because the order of operations can matter. It is not intuitively obvious to most developers, which should come first among signing, compression and encryption. This issues was first raised in Spring of 2001 [[davis](#)] but was only addressed in TLS

by [RFC7366] more than a dozen years later.

Getting the cryptography right is important because the Internet, and therefore the work of the IETF, has become a tempting target for all types of attackers, from individual "script kiddies," through criminal commercial botnet and phishing ventures, up to national-scale adversaries operating on behalf of their nation-state.

Salz

Expires January 9, 2017

[Page 3]

Internet-Draft

crypto-pubreview

July 2016

4. Things to avoid

"Sunlight is said to be the best of disinfectants; electric light the most efficient policeman." - Louis Brandeis, *Other People's Money and How Bankers Use it*, first published as a set of articles in *Harper's Weekly* in 1914.

Cryptography that is developed in private, such as among an industry consortium is a bad idea. Notable examples of this include:

- o A5/1 and A5/2 for GSM-based mobile phones.
- o WEP and WPA for WiFi access.
- o SSLv2, while published, was developed by a private group at an Internet startup. It had security flaws that had global effects decades later, see <https://drownattack.com/>.

It is hard to get good public review of patented cryptography, unless there is a strongly compelling need. For example, decades ago RSA was the only practical public-key mechanism available and it was therefore studied pretty extensively.

Part of the concern about patented cryptography is that the patent-holder has every incentive to provide that their system is good, while the rest of the world generally has little interest in proving that their commercial venture is bad. Examples of this include:

- o Algebraic Eraser, prior to its presentation at IETF-xx, received

little public interest.

- o There is not a great deal of study about NTRU.

Both of these items are "lattice cryptography" and that might also be a reason for lack of review; the field might not have much interest yet.

- o XXX STILL MORE NEEDED

5. Why limit to MTI?

There is an argument that any new RFC not classified as "historical" should not specify or recommend insufficiently-reviewed cryptography, whether it MTI or not. This document limits itself to MTI for a couple of reasons.

Salz

Expires January 9, 2017

[Page 4]

Internet-Draft

crypto-pubreview

July 2016

- o Informational RFCs often document how to interoperate with other systems, and this is useful. As examples of this, see the Internet-Drafts on scrypt and [[RFC7693](#)].
- o Putting insufficiently-reviewed algorithms into an RFC can be one way to spur interest in getting more reviews. This MUST NOT be the primary motivation for inclusion, but it can be a useful side-effect, and might lead to future "promotion" to MTI. Note that waiting through draft and last-call state, then claiming "nobody broke it" MUST NOT be used as the rationale; this is using the IETF to host a "proof by contest."
- o Drawing a strict boundary just around MTI is a tractable problem. Drawing a similar boundary around all potential IETF uses of cryptography is bound to have mistakes and errors, any one of which can have the potential to make the IETF look bad, if not incompetent.
- o Requiring MTI to have public review also pressures everyone to conform and raise the bar. Imagine a hypothetical national security body that has a new cryptographic algorithm, Military Top-secret Encryption, or MITE. If MITE is not MTI, then that

government might be hard-pressed to get it accepted into off-the-shell offerings. If it is MTI without sufficient review, then they have good reason to keep flaws in existing cryptography private. To avoid both situations, the that government should work to get MITE as an MTI, and would now have the burden to make sure it receives sufficient analysis.

6. How to Do it Right

Cryptographic agility, [[RFC7696](#)], is probably a MUST. While it has its detractors, there are no known (to the author) practical considerations to evolving a deployed based to stronger crypto, while still maintaining interoperability with existing entities. This requires being able to make informed choices about when to use old weak crypto, and when to use the "latest and greatest," and while not much software, and essentially no end-users, are capable of making that choice, it seems sadly the best we can do.

NIST is an important reference for crypto algorithms. Yes, they have made mistakes (DUAL_EC_DRBG), but so has the IETF (opaque-prf) in the same area. But they have run respected international contests and their output receives heavy scrutiny.

The second consideration is to avoid temptation and premature optimization. Do not adopt an algorithm just because it seems "small and fast" or comes from "someone I respect."

Salz

Expires January 9, 2017

[Page 5]

Internet-Draft

crypto-pubreview

July 2016

6.1. Public Review

What constitutes sufficient public review? It is hard to say. This section attempts to provide some guidelines.

An open competition, such as those that led to AES (XXX ref) and SHA-3 (XXX ref) seem to be good, even when they come from sources that are under widespread suspicion, like the US Government. These efforts, like the Password Hashing Competition <https://password-hashing.net/> , had wide international participation and analysis by many noted experts.

Papers presented in the various Crypto conferences (XXX need list) are good. Same for various Usenix workshops.

Proof by contest - "Nobody's Claimed my \$200 reward" - are generally useless, for a number of reasons. They tend to be promoted by amateur cryptographers as a way to get attention, and if someone actually looks at them they are always cracked. Numerical analysis is a better approach, albeit much harder work. Contests designed to show the amount of "brute-force" work needed, such as the old RSA factoring challenges, can be useful. But they do not show, for example, if the cryptography under test is fundamentally flawed or not.

Public review is also a natural fit for the IETF, which takes "rough consensus and running code" as an axiom. Theory reduced to practice is much easier, and much less of a limited academic exercise, to review.

7. Acknowledgements

Thanks to Stephen Farrell for instigating this.

8. References

8.1. 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, <<http://www.rfc-editor.org/info/rfc2119>>.

8.2. Informative References

[acoustic]

Technion and Tel Aviv University, Weizmann Institute of Science, and Tel Aviv University, "RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis", December 2013, <<http://www.tau.ac.il/~tromer/papers/acoustic-20131218.pdf>>.

[davis]

"Defective Sign & Encrypt in S/MIME, PKCS#7, MOSS, PEM,

PGP, and XML.", Usenix Proc. Usenix Tech. Conf., June 2001, <http://world.std.com/~dtd/sign_encrypt/sign_encrypt7.PDF>.

- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", [BCP 188](#), [RFC 7258](#), DOI 10.17487/RFC7258, May 2014, <<http://www.rfc-editor.org/info/rfc7258>>.
- [RFC7366] Gutmann, P., "Encrypt-then-MAC for Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", [RFC 7366](#), DOI 10.17487/RFC7366, September 2014, <<http://www.rfc-editor.org/info/rfc7366>>.
- [RFC7693] Saarinen, M-J., Ed. and J-P. Aumasson, "The BLAKE2 Cryptographic Hash and Message Authentication Code (MAC)", [RFC 7693](#), DOI 10.17487/RFC7693, November 2015, <<http://www.rfc-editor.org/info/rfc7693>>.
- [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>>.

Author's Address

Rich Salz
Akamai Technologies

Email: rsalz@akamai.com