CURDLE
Internet-Draft
Intended status: Informational
Expires: May 3, 2018

R. Moskowitz
L. Xia
Huawei

October 30, 2017

Small Crypto for Small IOT
draft-moskowitz-small-crypto-00

Abstract

This memo proposes to leverage the Keccak algorithm at a function "width" of b=400 to provide a set of "small" cryptographic functions well suited to the IOT constrained environment. As such, only 128 bit security level is provided here. The full set of NIST approved Keccak derived functions that can work within the b=400 constraint, plus the 3rd round candidate in the CAESAR competition, Ketje, are defined.

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 3, 2018.

Copyright Notice

Table of Contents

1. Introduction

The Keccak [Keccak] algorithm at a width of b=1600 was selected by NIST for SHA-3 as defined in FIPS 202 [NIST.FIPS.202]. It is further used to define additional hashing functions in NIST SP 800-185 [NIST.SP.800_185], all with b=1600. This selection is well suited for 64 bit processors and large messages. It can take advantage of multiple core CPUs and works well even on 32 bit processors. It is

not well suited for small messages and 8 bit processors that are common in IOT.

A full set of function widths of 25, 50, 100, 200, 400, and 800 are also defined in Keccak. Selection of values of other than 1600 is a risk/design trade off. A width of 400 is used here as the smallest that can provide 128 bits security strength.

Keccak provides more than a secure hash function. FIPS 202 defines the SHAKE function to provide a hash output of arbitrary length, rather than current practice of truncating a longer hash to the needed length. SP 800-185 defines a keyed hash, KMAC, that does not require the HMAC complexity and computational cost. This also can provide a PRF.

Finally, Keccak also provides an AEAD cipher, Ketje, to thus deliver in a single base function, the full suite of symmetric cryptographic functions.

1.1. Document status

Significant portions of this document are stubs. That is not because the information is not available. Rather the information for those sections need to be formatted for inclusion in this document. All the algorithms exist for Keccak and need only be adjusted for B=400. There is considerable performance data and security comparison data from the Keccak team. It needs extraction from other publications and formatted into this document.

2. Terms and Definitions

2.1. Requirements 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].

2.2. Notations

This section will contain notations

2.3. Definitions

TBD

3. Keccak Basics

Keccak is a extendable-output function (XOF) sponge construction hash. It breaks from the 'standard' ARX (addition, rotation and exclusive-or (XOR)) hash approach. Instead it using only bit-level transpositions, bit-level additions and multiplications (in GF(2)). This contributes to its superior performance.

This leads to one important nomenclature difference in Keccak. It does not have a block size. Rather in Keccak there is bit-rate which is one of the variable parameters.

3.1. Keccak Parameters

The Keccak primitive is:

Keccak-f[b], where b is 25, 50, 100, 200, 400, 800 or 1600 bits

b=1600 is used in FIPS 202 and SP 800-185. Here, b=400 is used.

Instances are denoted

Keccak[r, c]

capacity c determines the proven security strength against generic attacks
for a security level of n bits, the capacity must be c = 2n
and bitrate r = b - c

Thus for b=400 and a strength of 128 bits, r=144 and c=256

4. Keccak[144,256]

The instance Keccak[144,256] can be used for SHAKE128 [FIPS 202], cSHAKE128, KMAC128, KMACXOF128, TupleHash128, TupleHashXOF128, ParallelHash128, ParallelHashXOF128 [SP 800-185]. The difference is a bitrate of 144 rather than 1344.

Some of the above variants, such as the parallel hashes are not of value in small devices with small r which is not amendable to tree hashing.

To distinguish the form of the above hashes between the standard r=1344 and r=144, a designation of 'i' is appended to each name so that here SHAKE128i and KMAC128i are used.

5. The Ketje authenticated encryption scheme

Ketje Sr [Ketje] is already defined to use b=400.

6. Using SHAKE128i in Ed25519

Ed25519, RFC 8032 [RFC8032], specifies the parameter H(x) as SHA-512. A new form, Ed25519i, will use SHAKE128i for H(x). This one change will bring Ed25519 more into the 'reach' of the constrained environment.

The use of SHAKE128i is the only variant between Ed25519 and Ed25519i.

7. IANA Considerations

TBD. OID assignments

8. Security Considerations

8.1. Defense against future attacks

The safety margin in Keccak can be increased or de-creased simply by changing the number of rounds in Keccak-f. This is explained in "Note on Keccak parameters and usage" [NotesPandU], Section 6.

8.2. Security Comparisons

This memo proposes a radical change in the basic cryto-primatives used in protocols. As such, care is called for.

Keccak is not new. It has been well reviewed as explained in "Why Keccak is not ARX" [notARX]. Still, it is important to compare each Keccak function proposed here against the current Best Practice.

8.2.1. SHAKE128i to SHA-256

TBD.

8.2.2. KMAC128i to HMAC(SHA-256)

TBD.

8.2.3. Ketje Sr to AES-CCM

TBD.

8.2.4. Ed25519i to Ed25519

TBD.

9. Acknowledgments

Sections here draw heavily on information available on the Keccak [Keccak] site. I thank the Keccak Team in making this information openly available.

10. References

10.1. Normative References

[NIST.FIPS.202]
Dworkin, M., "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions", FIPS PUB 202, DOI 10.6028/nist.fips.202, August 2015, <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf>.

[NIST.SP.800_185]
Kelsey, J., Change, S., and R. Perlner, "SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash and ParallelHash", Special Publication SP 800-185, DOI 10.6028/nist.sp.800-185, December 2016, <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-185.pdf>.

[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>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

10.2. Informative References

[IEEE.802.15.6_2012]
IEEE, "IEEE Standard for Local and metropolitan area networks - Part 15.6: Wireless Body Area Networks", IEEE 802.15.6-2012, DOI 10.1109/ieeestd.2012.6161600, February 2012, <http://ieeexplore.ieee.org/servlet/opac?punumber=6161598>.

[Keccak] Bertoni, G., Daemen, J., Peeters, M., Van Assche, G., and R. Van Keer, "Team Keccak Home Page", <https://keccak.team/index.html>.

[Ketje] Bertoni, G., Daemen, J., Peeters, M., Van Assche, G., and R. Van Keer, "The Ketje authenticated encryption scheme", <https://keccak.team/ketje.html>.

[notARX] "Why Keccak is not ARX", <https://keccak.team/2017/not_arx.html>.

[NotesPandU] Bertoni, G., Daemen, J., Peeters, M., and G. Van Assche, "Note on Keccak parameters and usage", February 2010, <https://keccak.team/files/NoteOnKeccakParametersAndUsage.pdf>.

[RFC8163] Lynn, K., Ed., Martocci, J., Neilson, C., and S. Donaldson, "Transmission of IPv6 over Master-Slave/Token-Passing (MS/TP) Networks", RFC 8163, DOI 10.17487/RFC8163, May 2017, <https://www.rfc-editor.org/info/rfc8163>.

Appendix A. Use Cases

TBD.

A.1. Pacemaker connected via Body Area Network

In-body sensors, like pacemakers, connected via the Body Area Network (BAN [IEEE.802.15.6_2012]) will be very power conscious. Battery replacement can often require surgery. This will lead to loose interpretation of the HIPAA protection requirements to the detriment of the patient. In-body sensors are an important class of devices that would benefit from the most power and code efficient security components that still meet a strong security claim.

A.2. Automotive CAN FD Sensors

CAN FD (CAN with Flexible Data-Rate) is an extension to the original CAN bus protocol specified in ISO 11898-1. Developed in 2011 and released in 2012. Its larger data payload of 64 bytes can support a encrypting and authenticating protocol, but various in-vehicle constraints can make this challenging. Sensors may be in a very high temperature environment, making even a marginal CPU temperature increase due to heavy computations catastrophic. Cost is always a automotive component constraint and security tends to come in last in the cost competition. Thus any way that the cost of security can be lowered is a major win.

A.3. Building Automation and Control Network Sensors

BACnet (ISO 16484-5) is the standard for wired, in-building control systems. Recently IPv6 support (RFC 8163 [RFC8163]) was added. Some BACnet devices are extremely constrained; 4-bit processors are still common. Though AES on such 4-bit processors is available, it is extremely slow. A smaller security suite would be a real benefit for this environment.

Appendix B. Performance Comparisons

Most new crypto algorithm changes are to reduce risk. This proposal is to reduce cost and thus enable security in a lower class of devices as currently supported. This will only come about if there is a real performance improvement to justify potential non-interoperability.

Performance improvements can come for lower CPU/power demands, smaller code size, and/or smaller storage/memory requirements.

B.1. SHAKE128i to SHA-256

TBD. Magnitude faster?

B.2. KMAC128i to HMAC(SHA-256)

TBD.

B.3. Ketje Sr to AES-CCM

TBD.

B.4. Ed25519i to Ed25519

TBD.

B.5. Keccak Hardware to AES Hardware

TBD.

Authors' Addresses

Robert Moskowitz
Huawei
Oak Park, MI 48237

Email: rgm@labs.htt-consult.com

Liang Xia
Huawei
No. 101, Software Avenue, Yuhuatai District
Nanjing
China

Email: Frank.xialiang@huawei.com