PQCRYPTO
ICT-645622

Horizon 2020

PQCRYPTO

Post-Quantum Cryptography for Long-Term Security

Project number: Horizon 2020 ICT-645622

Small Devices: Reference implementations

Due date of deliverable: 30. September 2016
Actual submission date: 15. November 2016

Start date of project: 1. March 2015 Duration: 3 years

Coordinator:
Technische Universiteit Eindhoven
Email: `coordinator@pqcrypto.eu.org`
`www.pqcrypto.eu.org`

Revision 1

Project co-funded by the European Commission within Horizon 2020		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission services)	
RE	Restricted to a group specified by the consortium (including the Commission services)	
CO	Confidential, only for members of the consortium (including the Commission services)	

Small Devices: Reference implementations

Tung Chou, Tim Güneysu, Tobias Oder, Peter Schwabe

15. November 2016
Revision 1

The work described in this report has in part been supported by the Commission of the European Communities through the Horizon 2020 program under project number 645622 PQCRYPTO. The information in this document is provided as is, and no warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Abstract

This document provides the PQCRYPTO project's intermediate overview of reference implementations of post-quantum cryptographic primitives and schemes targeting (or suitable for) small embedded devices.

Keywords: Post-quantum cryptography, small devices, microcontrollers, public-key encryption, public-key signatures, secret-key encryption, secret-key authentication

1 Introduction

This document lists the reference software implementations of post-quantum schemes that are suitable for use in "small devices", in particular embedded microcontrollers. The main reference platform we are targeting with the implementations is the ARM Cortex-M4, a relatively high-end 32-bit microcontroller. There are essentially three reasons that we decided to choose this microcontroller as reference platform:

- Microcontrollers of the ARM Cortex-M family are becoming more and more widely deployed; we expect that they will dominate the market for small devices over the next years.
- Choosing a rather high-end microcontroller allows us to also consider schemes that may not be feasible on smaller devices.
- By the time that post-quantum cryptography is deployed in products, even lower-end microcontrollers will probably have grown to the size of high-end microcontrollers today. Choosing a high-end microcontroller now includes this prediction of increase in computing power (and size) of future microcontrollers.

Picking the M4 as a reference platform does not mean that we exclude smaller devices from the Cortex-M family or even smaller 8-bit microcontrollers. For some space-efficient schemes we also include software implementations for those lower-end platforms.

2 Implementations

The goal of this deliverable is to provide software *reference implementations* of post-quantum cryptographic schemes for embedded microcontrollers. The following two subsections list such software implementations developed as part of research within the PQCRYPTO project. In Subsection 2.1 we first list implementations written in C that would typically be considered "reference" implementations. However, it turns out that in many cases it is necessary to adapt those implementations to make them fit into the restrictions of embedded microcontrollers. The C reference implementations typically serve as a good starting point and reference to provide test vectors for more specialized implementations that we list in Subsection 2.2. Those implementations are also written in C, but use hand-optimized assembly for certain subroutines. Most of those specialized microcontroller implementations target the ARM Cortex-M family of 32-bit microcontrollers, but some extend to the lower-end 8-bit AVR microcontrollers.

For some schemes (in particular symmetric schemes like AES or ChaCha20) there already exist many portable implementations in C that can serve as a reference (and starting point) for microcontroller implementations. We do not list those implementations here; however, many of them are included in the SUPERCOP benchmarking framework [4].

2.1 Portable software in C

1. The C reference implementation of the SPHINCS stateless **hash-based** signature scheme described in [3] is included in the SUPERCOP benchmarking framework [4] in subdirectory `crypto_hash/sphincs256/ref`.

2. C reference implementation of the XMSS-MT **hash-based** signature scheme described in [6]:
   `https://joostrijneveld.nl/papers/multitarget_xmss/`
3. C reference implementation of the NewHope (**lattice-based**) key-exchange scheme described in [2]:
   `https://cryptojedi.org/crypto/#newhope`
4. C reference implementation of the TESLA (**lattice-based**) signature scheme described in [1]:
   `https://cryptojedi.org/crypto/#tesla`
5. C reference implementation of the MQDSS **multivariate** signature scheme described in [5]:
   `https://joostrijneveld.nl/papers/mqdss/`
6. C reference implementation of McEliece/Niederreiter **code-based** public-key encryption:
   `http://www.win.tue.nl/~tchou/mcbits/`

2.2 Software for specific embedded microcontrollers

2.2.1 Software targeting ARM Cortex-M

1. Optimized AES for Cortex M3 and M4:
   `https://github.com/Ko-/aes-armcortexm`
2. Implementation of the ChaCha20 stream cipher for Cortex-M:
   `https://gitlab.science.ru.nl/mneikes/arm-chacha20`.
3. Implementation of the SPHINCS **hash-based** signature scheme for ARM Cortex-M3:
   `https://joostrijneveld.nl/papers/armedsphincs/`.
4. Implementation of the NewHope **lattice-based** key-exchange scheme on ARM Cortex-M0, M3, and M4:
   `https://github.com/erdemalkim/newhopearm`
5. Implementation of the BLISS **lattice-based** signature scheme for ARM Cortex-M4F
   `https://www.emsec.rub.de/media/crypto/veroeffentlichungen/2016/08/26/bliss_arm.zip`
6. Implementation of the TESLA **lattice-based** signature scheme for ARM Cortex-M4F
   `https://github.com/OtoriTakeo/Flying_TESLA`
7. Implementation of **lattice-based** (binary Ring-LWE) encryption for ARM Cortex-M0:
   `https://www.emsec.rub.de/media/crypto/veroeffentlichungen/2016/08/26/bin_lwe_arm.zip`
8. Implementation of the QcBits **code-based** (QC-MDPC) encryption scheme for ARM Cortex M4:
   `http://www.win.tue.nl/~tchou/qcbits/`

2.2.2 Software targeting AVR

1. Implementation of **lattice-based** (Ring-LWE-based) encryption and BLISS signatures for AVR ATMega:
   https://www.sha.rub.de/media/crypto/veroeffentlichungen/2016/06/08/High-Performance-Lattice-Crypto-Code.zip
2. Implementation of **code-based** (QC-MDPC) encryption for AVR ATMega:
   http://www.sha.rub.de/media/attachments/files/2013/08/MDPC_Atmel.rar

References

[1] Erdem Alkim, Nina Bindel, Johannes Buchmann, and Özgür Dagdelen. Tesla: Tightly-secure efficient signatures from standard lattices, 2016. http://cryptojedi.org/papers/#tesla.

[2] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange – a new hope. In *Proceedings of the 25th USENIX Security Symposium*. USENIX Association, 2016. http://cryptojedi.org/papers/#newhope.

[3] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: practical stateless hash-based signatures. In Marc Fischlin and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Computer Science*, pages 368–397. Springer-Verlag Berlin Heidelberg, 2015. http://cryptojedi.org/papers/#sphincs.

[4] Daniel J. Bernstein and Tanja Lange. eBACS: ECRYPT benchmarking of cryptographic systems. http://bench.cr.yp.to (accessed 2016-09-30).

[5] Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. From 5-pass mq-based identification to mq-based signatures. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – Asiacrypt 2016*, Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg, 2016 (to appear). http://cryptojedi.org/papers/#mqdss.

[6] Andreas Hlsing, Joost Rijneveld, and Fang Song. Mitigating multi-target attacks in hash-based signatures. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *Public-Key Cryptography – PKC 2016*, volume 9614 of *Lecture Notes in Computer Science*, pages 387–416. Springer-Verlag Berlin Heidelberg, 2016. https://eprint.iacr.org/2015/1256.