Fully Hybrid TLSv1.3 in WolfSSL on Cortex-M4

Mila Anastasova[1], Reza Azarderakhsh[1,2] *Member, IEEE*, and Mehran Mozaffari Kermani [3] *Senior Member, IEEE*

[1]CEECS Department and I-SENSE at Florida Atlantic University, Boca Raton, FL, USA
(manastasova2017, razarderakhsh)@fau.edu

[2]PQSecure Technologies, LLC, Boca Raton, FL, USA

[3]CES Department at University of South Florida, Tampa, FL, USA
mehran2@usf.edu

March 2024

Content

Motivation behind Hybrid Public Key Infrastructure

ECC offers :

- Efficiency in timing, energy, power consumption, and memory consumption (small key sizes).
- Deployed for key derivation and authentication - ECDH and ECDSA (EdDSA).

Curve448 (and it birationally equivalent Ed448) offers :

- Higher security level (224-bits).
- Addresses security backdoor issues of NIST curves.

Elliptic Curve Cryptography is well-studied and widely deployed. However, large scale quantum computers threatens to break ECDLP in sub-exponential time.

Motivation behind Hybrid Public Key Infrastructure

Lattice-based Post-Quantum Cryptography offers :

- Efficiency in timing, energy, power consumption. Relatively compact key sizes.
- Applicable for many crypto instances (PKE, DSA, Digest, Identification functions, etc.).

CRYSTALS-Kyber and CRYSTALS-Dilithium are :

- Finalist of the NIST PQ Standardization process.
- Addresses security backdoor issues of NIST curves.

Post-Quantum primitives are believed to be robust against quantum adversary. However, they fail to fulfill the security criteria set by the government and industry.

Previous Work

- PQ and Hybrid Signatures in TLSv1.2 and TLSv1.3 :
  - Hybrid Signatures in X.509 presented in [KPDVG18].
  - PQ-only message Signatures in TLSv1.2 are shown in [SKD20b], [SKD20a].
  - PQ-only message and X.509 Signatures in TLSv1.2 are shown in [MS22].
- PQ and Hybrid Key Exchange in TLS and other protocols :
  - Hybrid Key Exchange prototyping and deployment is shown in [CPS19], [CC19].
  - Hybrid Key Exchange in HPKE is presented in [AKM22].
- Library enhancements :
  - OQS and OpenSSL libraries integration of PQ-standalone message and X.509 PKI Signatures [SKD20b].

Our Contributions

Present a Fully Hybrid TLSv1.3 based on Curve448 and Crystals-Kyber1024 and Ed448 and Crystals-Dilithium5 based OpenSSL & wolfSSL cryptographic libraries :

- We enhance OpenSSL to generate X.509 hybrid `Ed448_Dilithium5` keys and certificates in PEM format.
- We implement `Curve448_Kyber1024` hybrid key exchange.
- We upgrade to sign and verify based on hybrid DSS `Ed448_Dilithium5`.
- We deploy processing `Ed448_Dilithium5` hybrid keys and certificates.
- We evaluate our hybrid TLSv1.3 based on `Curve448_Kyber1024` and `Ed448_Dilithium5` on the ARMv7 Cortex-M4 STM32F413 microcontroller.

Curve448 ECDH & Ed448 DSA

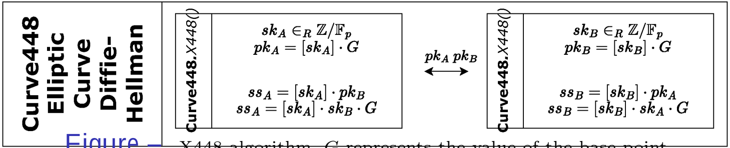

Figure – X448 algorithm. G represents the value of the base point

Figure – Ed448 algorithm [JL17]. H denotes $SHAKE256$. L represents the order of Ed448 curve. G represents the value of the base point

Curve448 ECDH & Ed448 DSA

Figure – X448 algorithm. G represents the value of the base point

Figure – Ed448 algorithm [JL17]. H denotes $SHAKE256$. L represents the order of Ed448 curve. G represents the value of the base point

Crystals-Kyber & Crystals-Dilithium

Post-Quantum Public Key Encryption
Crystals-Kyber

Kyber.PKE.KeyGen()

Output: $sk \in B^{12 \cdot k \cdot n/8}, pk \in B^{12 \cdot k \cdot n/8+32}$
$d \leftarrow B^{32}$
$(\rho, \sigma) := G(d)$
$N := 0$
for $i = 0$ **to** $k-1$ **do**
for $j = 0$ **to** $k-1$ **do**
$\hat{\mathbf{A}}[i][j] := Parse(XOF(\rho, j, i))$
end for
end for
for $i = 0$ **to** $k-1$ **do**
$\mathbf{s}[i] := CBD_{\eta_1}(PRF(\sigma, N))$
$N := N + 1$
end for
for $i = 0$ **to** $k-1$ **do**
$\mathbf{e}[i] := CBD_{\eta_1}(PRF(\sigma, N))$
$N := N + 1$
end for
$\hat{\mathbf{s}} := NTT(\mathbf{s})$
$\hat{\mathbf{e}} := NTT(\mathbf{e})$
$\hat{\mathbf{t}} := \hat{\mathbf{A}} \cdot \hat{\mathbf{s}} + \hat{\mathbf{e}} pk := (enc_{12}(\hat{\mathbf{t}} \bmod^{+} q) || \rho)$
$sk := enc_{12}(\hat{\mathbf{s}} \bmod^{+} q)$
return (pk, sk)

Kyber.PKE.Dec()

Input: $sk \in B^{12 \cdot k \cdot n/8}, c \in B^{d_u \cdot k \cdot n/8 + d_v \cdot n/8}$
Output: $m \in B^{32}$
$\mathbf{u} := D_q(dec_{d_u}(c), d_u)$
$v := D_q(dec_{d_v}(c + d_u \cdot k \cdot n/8), d_v)$
$\hat{\mathbf{s}} := dec_{12}(sk)$
$m := enc_1(C_q(v - NTT^{-1}(\hat{\mathbf{s}}^T \circ NTT(\mathbf{u})), 1))$
return m

Kyber.PKE.Enc()

Input: $pk \in B^{12 \cdot k \cdot n/8+32}, m \in B^{32}, r \in B^{32}$
Output: $c \in B^{d_u \cdot k \cdot n/8 + d_v \cdot n/8}$
$N := 0$
$\hat{\mathbf{t}} := dec_{12}(pk)$
$\rho := pk + 12 \cdot k \cdot n/8$
for $i = 0$ **to** $k-1$ **do**
for $j = 0$ **to** $k-1$ **do**
$\hat{\mathbf{A}}^T[i][j] := Parse(XOF(\rho, i, j))\$$
end for
end for
for $i = 0$ **to** $k-1$ **do**
$\mathbf{r}[i] := CBD_{\eta_1}(PRF(r, N))$
$N := N + 1$
end for
for $i = 0$ **to** $k-1$ **do**
$\mathbf{e_1}[i] := CBD_{\eta_2}(PRF(r, N))$
$N := N + 1$
end for
$e_2 := CBD_{\eta_2}(PRF(r, N))$
$\hat{\mathbf{r}} := NTT(\mathbf{r})$
$\mathbf{u} := NTT^{-1}(\hat{\mathbf{A}}^T \cdot \hat{\mathbf{r}}) + \mathbf{e_1}$
$v := NTT^{-1}(\hat{\mathbf{t}}^T \cdot \hat{\mathbf{r}}) + e_2 + D_q(d_1(m), 1)$
$c_1 := enc_u(C_q(\mathbf{u}, d_u))$
$c_2 := enc_v(C_q(v, d_v))$
return $c = (c_1 \| c_2)$

Figure – Crystals-Kyber algorithm [BDK+18]. Each variable represents (the coefficients of) a polynomial, bold text style denotes vector of polynomials, capital letter notation denotes a matrix. *enc* and *dec* represents encode/decode, *C* and *D* present Compress/Decompress, respectivel

Crystals-Kyber & Crystals-Dilithium

Figure CRYSTALS-Dilithium algorithm [DKL+18]. Each variable represents a polynomial, bold text style denotes vector of polynomials, capital letter notation denotes a matrix

Crypto Performance on ARM Cortex-M4

Cortex-M4 Features

- 16 32-bit core registers
- *32 32-bit FP registers
- 1 CC per instruction except memory accesses

Implementation strategies

- Use the entire register set.
- Operate on larger operand sets.
- Re-organize the instruction ow for e cient design.

NIST recommended Cortex-M4 WiFi-equipped STM32F413-ZH.

- Features :
  - 1.5MB of ash memory
  - 320KB of RAM
- Tools :
  - STM32CubeIDE
  - wolfSSL library

- Goal :
  - Deploy enhanced Fully Hybrid wolfSSL TLS1.3 protocol to obtain performance.

Crypto Performance on ARM Cortex-M4

Cortex-M4 Features

- 16 32-bit core registers
- *32 32-bit FP registers
- 1 CC per instruction except memory accesses

Implementation strategies

- Use the entire register set.
- Operate on larger operand sets.
- Re-organize the instruction ow for e cient design.

NIST recommended Cortex-M4 WiFi-equipped STM32F413-ZH.

- Features :
  - 1.5MB of ash memory
  - 320KB of RAM
- Tools :
  - STM32CubeIDE
  - wolfSSL library

- Goal :
  - Deploy enhanced Fully Hybrid wolfSSL TLS1.3 protocol to obtain performance.

Public Key Infrastructure

Figure The Public Key Infrastructure (PKI) built using classical (Ed448) and post-quantum (Dilithium5) Digital Signature Algorithm (DSA) techniques. The gray data refers to the information elds found in the X.509 les. Superscript indicates the owner of the data, while subscript indicates the type of information

Fully Hybrid TLSv1.3

Figure TLSv1.3 execution ow graphical representation. Gray data refers to the information elds included in X.509 les, where superscript indicates the owner of the data and subscript indicates the type of information. The compute stages are represented by solid box lines, the message ow is represented by discontinuous lines, and the certi cate le is represented by scattered box lines

wolfSSL Fully Hybrid TLSv1.3 Performance Evaluation

Work	KEX	Auth	Cert Verify	TLS1.3 handshake	TLS1.3 with AEAD
wolfSSL[wol]	X448	Ed448	Ed448	-	44,358,855
Anastasova et al. [AEKL+23]	X448	Ed448	Ed448	-	46,310,749
This work	X448 & Kyber1024	Ed448 & Dil5	-	97,624,103	106,735,300
This work	X448 & Kyber1024	Ed448 & Dil5	Ed448 & Dil5	114,017,313	123,139,034

Table Performance of the entirely hybrid TLSv1.3 handshake and the overall TLSv1.3 protocol when a short 15B message is delivered between communication parties. The values are expressed in terms of clock cycles [CC]

Conclusions & Future Work

- Conclusions
  We propose Fully Hybrid TLSv1.3 based on wolfSSL and OpenSSL cryptographic libraries :
  - We generate Ed448_Dilithium5 hybrid keys and certi cates adapting OpenSSL.
  - We deploy high-security Curve448_Kyber1024 hybrid key exchange in the TLSv1.3 handshake.
  - We deploy Ed448_Dilithium5 hybrid message signature.
  - We deploy Ed448_Dilithium5 hybrid certi cate veri cation.
  - We report 2:43 overhead compared to the classical-only TLSv1.3 based on the same classical primitives when certi cate veri cation is omitted, and 2.67 overhead when hybrid certi cate is transmitted and veri ed.
- Future Work
  - SCA protected Fully Hybrid TLSv1.3.
  - Fully Hybrid TLSv1.3 on higher end ARM platforms.

Thank you for the attention !

If you have any inquiries, please feel free to contact our team :

manastasova2017@fau.edu
razarderakhsh@fau.edu
mehran2@usf.edu

[AEKL+23] Mila Anastasova, Rabih El Khatib, Aimee Laclaustra, Reza Azarderakhsh, and Mehran Mozaffari Kermani. Highly Optimized Curve448 and Ed448 design in wolfSSL and Side-Channel Evaluation on Cortex-M4. In *2023 IEEE Conference on Dependable and Secure Computing (DSC)*, pages 1–8. IEEE, 2023.

[AKM22] Mila Anastasova, Panos Kampanakis, and Jake Massimo. PQ-HPKE : post-quantum hybrid public key encryption. *Cryptology ePrint Archive*, 2022.

[BDK+18] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Kyber : a CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367. IEEE, 2018.

[CC19] Matt Campagna and Eric Crockett. Hybrid post-quantum key encapsulation methods (PQ KEM) for transport layer security 1.2 (TLS). *Internet Engineering Task Force, Internet-Draft draft-campagna-tls-bike-sike-hybrid*, 1, 2019.

[CPS19] Eric Crockett, Christian Paquin, and Douglas Stebila. Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH. *Cryptology ePrint Archive*, 2019.

[DKL+18] Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium : A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 238–268, 2018.

[JL17] Simon Josefsson and Ilari Liusvaara. Edwards-Curve Digital Signature Algorithm (EdDSA). RFC 8032, January 2017.

[KPDVG18] Panos Kampanakis, Peter Panburana, Ellie Daw, and Daniel Van Geest. The viability of post-quantum X. 509 certificates. *Cryptology ePrint Archive*, 2018.

[MS22] Dominik Marchsreiter and Johanna Sepúlveda. Hybrid Post-Quantum Enhanced TLS 1.3 on Embedded Devices. In *2022 25th Euromicro Conference on Digital System Design (DSD)*, pages 905–912. IEEE, 2022.

[SKD20a] Dimitrios Sikeridis, Panos Kampanakis, and Michael Devetsikiotis. Assessing the overhead of post-quantum cryptography in TLS 1.3 and SSH. In *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, pages 149–156, 2020.

[SKD20b] Dimitrios Sikeridis, Panos Kampanakis, and Michael Devetsikiotis. Post-quantum authentication in TLS 1.3 : a performance study. *Cryptology ePrint Archive*, 2020.

[wol] wolfSSL. wolfSSL. Last accessed on October 1, 2023 from https ://www.wolfssl.com/.