Vincent Hwang

Email | Github | Personal Website | Google Scholar | DBLP

1 Education

PhD, Cryptographic Engineering — Germany/the Netherlands | Jan. 2023 – March/April 2026
Max Planck Institute for Security and Privacy (employed)
Radboud Univeristy (external)
Supervisors: Peter Schwabe (supervisor) and Bo-Yin Yang (co-supervisor)
Thesis submission: July 2025.

MSc., Department of Computer Science and Information Engineering — Taiwan | Sept. 2021 – Jun. 2022
National Taiwan University
Supervisors: Yen-Huan Li and Bo-Yin Yang

BSc., Department of Computer Science and Information Engineering — Taiwan | Sept. 2016 – Jun. 2021
National Taiwan University

2 Research Interests

I'm doing research in cryptographic engineering. Cryptographic engineering is a field exploring how the mathematical constructs in cryptography could/should be best implemented in real-world computing devices. I position myself as a computer scientist turning the high-level ideas into optimized computer programs. Frequently, this amounts to several iterations of refinements for the high-level ideas and experiments through assembly programming. I programmed the computationally-intensive polynomial arithmetic in lattice-based cryptosystems for embedded devices (Armv7-M onward) and high performance processors (Armv8-A onward for `aarch64`, AVX2 extension onward for `x86-64`). I coauthored implementation papers targeting lattice-based candidates NTRU, NTRU Prime, and Saber, and standards `ml-kem` and `ml-dsa` (previously known as Kyber and Dilithium) of the Post-Quantum Cryptography Standardization by the National Institute of Standards and Technology (NIST). Optimizing polynomial arithmetic for cryptography usually involves some basics of algebra and various assembly languages. I'm also working on other topics like formal verification, and exploring interesting computing devices from time to time. Recently, I'm mainly programming for the elliptic-curve discrete logarithm on the H100 GPU, and also optimizing the mathematically-heavy NIST standard `fn-dsa` (previously known as Falcon) in assembly.

Key words: Cryptographic engineering, assembly programming, post-quantum cryptography, and practical integer and polynomial multiplications.

3 Academic Service

- 2026: Reviewer of TCHES 2026 (×6), Eurocrypt 2026 (×1), Crypto 2026 (×2), Artifact Evaluation Committee Member of TCHES 2026 (×6), Artifact Review Committee of Eurocrypt 2026 (×4).
- 2025: Reviewer of TCHES 2025 (×8), ArcticCrypt 2025 (×1), CT-RSA 2025 (×2), JCEN (×1), Artifact Evaluation Committee Member of TCHES 2025 (×5).
- 2024: Reviewer of Crypto 2024 (×1), TCHES 2024 (×3).
- 2023: Artifact Committee Member of TCHES 2023 (×2).

TCHES = Transactions on Cryptographic Hardware and Embedded Systems;
CT-RSA = The Cryptographers' Track at RSA Conference;
JCEN = Journal of Cryptographic Engineering;
$(\cdot)$ = The number of reviews submitted.

4 Programming Skills

- Assembly (very familiar): Armv7-M, Armv7E-M, Armv8-A, AVX2. Low-level optimizations.
- Assembly (somewhat familiar): Armv9-A, AVX-512. Low-level optimizations. Ongoing research.
- C (very familiar): Primary for interfacing between the assembly and the high-level api. Sometimes I use function pointers for unit tests.
- C++ (somewhat familiar): Primary about templates for scalability. I rarely use the standard libraries as they are not built for cryptographic uses where various secure programming practice must be employed.
- CUDA (somewhat familiar): Ongoing research.
- Haskell (some experience): Scripts for generating some programs and constants used in low-level optimizations.

5 Preferred Work Locations

Citizenship: USA, Taiwan.

6 Publications (Reversed Chronological Order)

Author names in alphabetical order. * = Contributions included in the PhD thesis. ** = Contributions included in the Master's thesis.

PhD program ongoing.

16. Phillip Gajland, Vincent Hwang, Jonas Janneck. Shadowfax: Hybrid Security and Deniability for AKEMs. To appear at USENIX 2026. IACR ePrint. Artifact.

Summary: We proposed Shadowfax demonstrating how deniability could be preserved in post-quantum and hybrid settings, and provided several portable implementations of Shadowfax. When instantiated with standardised components (ML-KEM and Falcon), Shadowfax yielded ciphertexts of 1728 bytes and public keys of 2036 bytes, with encapsulation and decapsulation costs of 1.8M and 0.7M cycles on an Apple M1 Pro running at 3 GHz. I was responsible for the portable implementations.

15*. Gilles Barthe, Gustavo Xavier Delerue Marinho Alves, Hugo Pacheco, José Bacelar Almeida, Luís Esquível, Manuel Barbosa, Peter Schwabe, Pierre-Yves Strub, Tiago Oliveira, and Vincent Hwang. Faster Verification of Faster Implementations: Combining Deductive and Circuit-Based Reasoning in EasyCrypt. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 3526–3544. IEEE Computer Society, 2025. Paper. IACR ePrint.

Summary: We combined deductive reasoning and circuit-based reasoning in EasyCrypt, verified the AVX2-optimized rejection sampling in Kyber for the first time and the equivalences of the optimized implementations of the compression functions, and simplified the verification of the AVX2-optimized Keccak permutation. I proposed the optimizations for the compression functions and wrote the corresponding part in the paper.

14*. Vincent Hwang, YoungBeom Kim, and Seog Chung Seo. Multiplying Polynomials without Powerful Multiplication Instructions. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(1):160–202, 2024. Paper. Artifact. Slides. IACR ePrint.

Summary: We optimized Dilithium and Saber on Cortex-M3 and 8-bit AVR. For the Dilithium coefficient ring modulo a prime, we generalized Barrett multiplication and proposed several versions suitable for multi-limb arithmetic on platforms without powerful multiplication instructions. For the power-of-two-coefficient ring case, we showed that Nussbaumer performs the best if there are no precision issues. I proposed the ideas, implemented the Cortex-M3 optimizations, and wrote the corresponding parts of the paper.

13*. Vincent Hwang. Formal Verification of Emulated Floating-Point Arithmetic in Falcon. In *International Workshop on Security*, pages 125-141. Springer, 2024. Paper. Artifact. Slides. IACR ePrint.

Summary: This paper verified the functional equivalences of the software-emulated floating-point additions/-multiplications and the range of the intermediate floating-point values of the FFT computations in the signature generation of Falcon with the domain-specific language CryptoLine. I proposed the ideas, carried out the verification, and wrote the paper.

12*. Vincent Hwang. A Survey of Polynomial Multiplications for Lattice-Based Cryptosystems. *IACR Communications in Cryptology*, 1(2), 2024. Paper. IACR ePrint.

Summary: This paper reviewed several techniques for multiplying polynomials in polynomial rings of the form $\mathbb{Z}_q[x]/\langle x^n - \alpha x - \beta\rangle$ in practice. There are three emphases: modular arithmetic, homomorphisms, and vectorization. I wrote the paper.

11*. Vincent Hwang. Pushing the Limit of Vectorized Polynomial Multiplication for NTRU Prime. In *Australasian Conference on Information Security and Privacy*, pages 84–102. Springer, 2024. Paper. Artifact. Slides. IACR ePrint.

Summary: We furthered the NTRU Prime optimizations and optimized on Cortex-A72 with Armv8-A Neon and on Haswell with AVX2. The paper systematically reviewed the notion of vectorization on various platforms, and replaced Rader's FFT with truncated Rader's FFT. I proposed the ideas, implemented the optimizations, and wrote the paper.

10*. Vincent Hwang, Chi-Ting Liu, and Bo-Yin Yang. Algorithmic Views of Vectorized Polynomial Multipliers – NTRU Prime. In *International Conference on Applied Cryptography and Network Security*, pages 24–46. Springer, 2024. Paper. Artifact. Slides. IACR ePrint.

Summary: We optimized NTRU Prime on Cortex-A72 with Armv8-A Neon. We explored Schönhage and Bruun's FFTs, integrated them with Good–Thomas and Rader's FFTs, and proposed two approaches for multiplying polynomials in NTRU Prime. Our approaches significantly reduced the number of small-dimensional polynomial multiplications while vectorizing for NTRU Prime. I proposed the optimizations, implemented the Bruun's FFT and the fastest approach, and wrote the paper.

9*. Han-Ting Chen, Yi-Hua Chung, Vincent Hwang, and Bo-Yin Yang. Algorithmic Views of Vectorized Polynomial Multipliers – NTRU. In Anupam Chattopadhyay, Shivam Bhasin, Stjepan Picek, and Chester Rebeiro, editors, *Progress in Cryptology – INDOCRYPT 2023*, pages 177–196. Springer, 2024. Paper. Artifact. Slides. IACR ePrint.

Summary: We optimized NTRU on Cortex-A72 with Armv8-A Neon. We improved the design choice of Toom–Cook and implemented the Toeplitz matrix-vector product approach for the polynomial multiplications in NTRU. I proposed the improvement of Toom–Cook, integrated the polynomial multipliers by coauthors, implemented the then-not-yet-deployed obvious optimizations, wrote the non-implementation part of the paper, and refined the implementation part of the paper.

Master's degree conferral.

8*. Vincent Hwang, Jiaxiang Liu, Gregor Seiler, Xiaomu Shi, Ming-Hsien Tsai, Bow-Yaw Wang, and Bo-Yin Yang. Verified NTT Multiplications for NISTPQC KEM Lattice Finalists: Kyber, SABER, and NTRU. 2022. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(4):718–750, 2022. Paper. Artifact.

Summary: We verified the assembly-optimized NTT-based polynomial multiplications in Kyber, Saber, and NTRU on Cortex-M4 and on Skylake with AVX2 with the domain-specific language CryptoLine. I rewrote the assembly implementation of the polynomial multiplication for NTRU on Cortex-M4, explained the program structure of the assembly implementations on Cortex-M4, and drew some figures in the paper.

7**. Erdem Alkim, Vincent Hwang, and Bo-Yin Yang. Multi-Parameter Support with NTTs for NTRU and NTRU Prime on Cortex-M4. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(4):349-371, 2022. Paper. Artifact. Talk. IACR ePrint.

Summary: We revisited NTRU and NTRU Prime on Cortex-M4. We improved the initial butterflies of the Good–Thomas FFT, and the odd-radix butterflies, and proposed incomplete Good–Thomas FFT for code-size optimization. I proposed the optimizations, implemented the optimizations, and wrote the paper.

6*. Hanno Becker, Vincent Hwang, Matthias J. Kannwischer, Lorenz Panny, and Bo-Yin Yang. Efficient Multiplication of Somewhat Small Integers using Number–Theoretic Transforms. In *International Workshop on Security*, pages 3-23. Springer, 2022. Paper. Artifact. IACR ePrint.

Summary: We explored the FFT-based integer multiplications on Cortex-M3 and Cortex-M55, and found that the crossover point of the bit-size/performance of the FFT-based and the $\Theta(n^2)$ non-FFT-based integer multiplications is around 2048 bits. I was involved in the Cortex-M3 implementation.

5*. Amin Abdulrahman, Vincent Hwang, Matthias J. Kannwischer, and Amber Sprenkels. Faster Kyber and Dilithium on the Cortex-M4. In *International Conference on Applied Cryptography and Network Security*, pages 853–871. Springer. 2022. Paper. Artifact. IACR ePrint.

Summary: We revisited the optimization of Dilithium and Kyber on Cortex-M4. We incorporated assembly optimizations from prior Cortex-M4 works and Cortex-A72 works, and optimized the challenge polynomial multiplications in Dilithium. I proposed the Fermat number transform for the challenge polynomial multiplication in Dilithium, an optimization for the base multiplication in Kyber, and wrote the corresponding part of the paper with other coauthors.

4**. Hanno Becker, Vincent Hwang, Matthias J. Kannwischer, Bo-Yin Yang, and Shang-Yi Yang. Neon NTT: Faster Dilithium, Kyber, and Saber on Cortex-A72 and Apple M1. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(1):221-244, 2021. Paper. Artifact. IACR ePrint.

Summary: We rediscovered the multiplicative form of Barrett reduction, discovered a correspondence between Montgomery and Barrett multiplications, implemented the optimizations on Cortex-A72 and Apple M1 with Armv8-A Neon, and integrated the optimizations to Dilithium, Kyber, and Saber. I implemented the optimizations and wrote the corresponding parts of the paper.

3**. Amin Abdulrahman, Jiun-Peng Chen, Yu-Jia Chen, Vincent Hwang, Matthias J. Kannwischer, and Bo-Yin Yang. Multi-moduli NTTs for Saber on Cortex-M3 and Cortex-M4. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(1):127-151, 2021. Paper. Artifact. Talk. Slides. IACR ePrint.

Summary: Continuing the Saber optimizations on Cortex-M4, we further optimized Saber on Cortex-M3, the memory footprint on Cortex-M3 and Cortex-M4, and the masked implementation on Cortex-M4, and concluded that NTT is the fastest approach for Saber. I proposed the ideas, implemented the Cortex-M4 optimizations and the fastest approach on Cortex-M3, and wrote the corresponding parts of the paper.

Bachelor's degree conferral.

2*. Chi-Ming Marvin Chung, Vincent Hwang, Matthias J. Kannwischer, Gregor Seiler, Cheng-Jhih Shih, and Bo-Yin Yang. NTT Multiplication for NTT-unfriendly Rings: New Speed Records for Saber and NTRU on Cortex-M4 and AVX2. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(2):159–188, 2021. Paper. Artifact. Talk. Slides. IACR ePrint.

Summary: We migrated and implemented the Good–Thomas approach to all the parameter sets of NTRU on Cortex-M4. Additionally, we also optimized Saber on Cortex-M4 and NTRU and Saber on Skylake with AVX2. I was deeply involved in the Cortex-M4 implementations.

1*. Erdem Alkim, Dean Yun-Li Cheng, Chi-Ming Marvin Chung, Hülya Evkan, Leo Wei-Lun Huang, Vincent Hwang, Ching-Lin Trista Li, Ruben Niederhagen, Cheng-Jhih Shih, Julian Wälde, and Bo-Yin Yang. Polynomial Multiplication in NTRU Prime: Comparison of Optimization Strategies on Cortex-M4. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(1):217–238, 2020. Paper. Artifact. Talk. Slides. IACR ePrint.

Summary: We proposed three assembly-optimized NTT-based implementations for the polynomial multiplication in NTRU Prime on Cortex-M4. I was deeply involved in the Good–Thomas implementation.