Securing Edge Devices in the Post-Quantum Internet of Things Using Lattice-Based Cryptography

Zhe Liu, Kim-Kwang Raymond Choo, and Johann Großschädl

The authors provide a brief introduction to cryptographic methods for securing the post-quantum IoT, including hash-based digital signature schemes, code-based cryptography, multivariate public key cryptography, and lattice-based cryptography. In particular, they focus on the implementation aspects of lattice-based cryptography for resource-constrained IoT devices, and practical suggestions to choose appropriate implementation techniques.

Abstract

In order to increase the security of edge computing, all data transmitted to and from edge devices, as well as all data stored on edge devices, must be encrypted. Especially when the transmitted or stored data contains sensitive personal information, long-term protection over periods of ten or more years may be required, which can only be achieved with post-quantum cryptography. This article first gives a brief overview of post-quantum public-key cryptosystems based on hard mathematical problems related to hash functions, error-correcting codes, multivariate quadratic systems, and lattices. Then the suitability of lattice-based cryptosystems for resource-constrained devices is discussed and efficient implementations for 8 and 32-bit microcontrollers are outlined.

Introduction

The Internet of Things (IoT) can be defined as a global network of physical objects ("things") that are equipped with computation and communication capabilities, which enables them to be identified, monitored, and controlled over the Internet. It is estimated that by the end of 2020, somewhere between 20 and 50 billion smart devices — including various kinds of sensors, actuators, and other microsystems — will be connected to the Internet, outnumbering the world's population by a factor of between 2.5 and 6.5. These billions of devices will collect unprecedented amounts of data about the physical world in their environment, which needs to be transmitted to a central resource (e.g., a server in the cloud) for analysis and storage. However, such a traditional IoT model, in which the end devices are primarily used for data collection, while the information extraction, post processing, and decision making is primarily performed in the cloud, has raised many concerns about bandwidth requirements, latency problems, as well as security and privacy issues. *Edge computing*, also known as fog computing, aims to mitigate said concerns by performing some processing and analysis in the gateway that connects the IoT devices with the Internet or on a dedicated device located at the edge of the network, near the source of the data. This approach reduces the amount of data to be transmitted to the cloud and eliminates the round-trip delay associated with the transmission of data from the gateway to the cloud and the transmission of results in the other direction. In addition, edge computing has the potential to alleviate certain privacy issues by ensuring that all sensitive data is either kept on the edge device or only sent to the cloud after anonymization.

As edge devices may process and store sensitive personal information about their owners or users, they require effective protection against many kinds of attacks. An edge device can be attacked from two different directions, namely through the Internet on one hand, and through connected devices on the other hand, assuming the attacker is able to inject one or more manipulated devices into the network. These threats call for a sophisticated security architecture, which has to take the very specific constraints and requirements of the IoT into account. One of the main challenges toward a secure IoT is the fact that many IoT devices are highly constrained in terms of computational resources and network bandwidth. For example, a typical wireless sensor node, like Memsic's MICAz mote, is equipped with an 8-bit AVR microcontroller and has a few kilobytes of RAM and around 100 kB of flash memory to store a primitive operating system and application programs. More advanced IoT devices often come with an ARM Cortex-M microcontroller and possess between 32 and 64 kB RAM, as well as a few hundred kilobytes of flash memory. Edge devices are, in general, much more powerful than ordinary IoT devices since they have to perform local data processing. However, when choosing cryptographic algorithms and protocols to be run on an edge device, the resource restrictions of the IoT devices with which it needs to communicate securely must be taken into account. Regarding public key techniques, elliptic curve cryptosystems, such as Elliptic Curve Digital Signature Algorithm (ECDSA) and Elliptic Curve Diffie-Hellman (ECDH) have clear advantages over their traditional counterparts such as Rivest, Shamir, & Adleman (RSA) and DH due to the much shorter key lengths and associated savings in execution time, RAM requirements, and transmission bandwidth.

Digital Object Identifier:
10.1109/MCOM.2018.1700330

*Zhe Liu is with Nanjing University of Aeronautics and Astronautics and the University of Luxembourg;
Kim-Kwang Raymond Choo is with The University of Texas at San Antonio; Johann Großschädl is with the University of Luxembourg.*

0163-6804/18/$25.00 © 2018 IEEE

The security of modern public key cryptosystems relies on the hardness of well studied mathematical problems such as the integer factoring problem (IFP), the discrete logarithm problem (DLP), or its elliptic curve variant, the ECDLP. It is widely acknowledged, however, that all these problems could be effectively solved with a quantum computer, which puts essentially every key exchange and digital signature scheme in use today at risk of being broken [1]. Estimates as to when the first powerful quantum computer will be available vary significantly, but according to some predictions it could happen before the end of the next decade (i.e., in less than 15 years). Fortunately, there are a few mathematical problems that are intractable not only for classical computers, but also when using a sophisticated quantum computer. The sub-area of cryptography that deals with the design, cryptanalysis, and implementation of cryptographic algorithms supposed to be able to withstand attacks by quantum computers is known as Post-Quantum Cryptography (PQC) and has recently gained a lot of interest. Regarding public key cryptosystems, the focus of research activities is currently directed toward four main categories: lattice-based cryptography, multivariate cryptography, hash-based cryptography, and code-based cryptography. These four categories differ greatly in terms of the underlying hard mathematical problem and also with respect to performance, key lengths, and the length of ciphertexts as well as digital signatures. Lattice-based cryptography is often regarded as one of the best options for PQC in the IoT since it combines high efficiency with reasonably short keys.

PQC has attracted the interest of standardization bodies all over the world. In 2016, the National Institute of Standards and Technology (NIST) announced their intention to standardize post-quantum cryptosystems and also released a call for proposals. Interested organizations and individuals were invited to submit quantum-resistant public key encryption algorithms, key agreement mechanisms, and digital signature schemes that could replace the currently used cryptosystems such as RSA. The deadline for the submission of proposals was November 2017, and it is estimated that a draft standard will be available after five to seven years (i.e., between 2023 and 2025), which includes a public analysis phase of three to five years. As part of this effort, it will be necessary to evaluate how well the proposals for post-quantum cryptosystems can satisfy the requirements of the IoT and its billions of devices with limited computing and communication capabilities.

Post-Quantum Cryptography

As stated before, there are four major directions for the realization of public key PQC, namely hash-based cryptography, code-based cryptography, multivariate cryptography, and lattice-based cryptography. We briefly summarize their main properties and present a few examples of each category.

Hash-based digital signature schemes, originally introduced by Ralph Merkle in 1979, require fewer security assumptions than number-theoretic signature schemes and are expected to be resistant against quantum computers. The classical Merkle Signature Scheme (MSS) is based on a one-time signature (OTS) system and uses a binary hash tree (nowadays called Merkle tree) to obtain a "many-time" signature scheme. The 2^n leaf nodes are hash values of the public keys of 2^n OTS key pairs, each of which can only be used to sign one message. Each inner node contains the hash of the concatenation of its two children, and the hash value at the root node is the public key of the scheme. The private key of the MSS is the entire set of OTS key pairs, which can be generated using a pseudo-random generator with a short seed to reduce storage requirements. An MSS with 2^n OTS key pairs can be used to sign 2^n messages, whereby the signature of the ith message m_i consists of the OTS signature on m_i using the ith OTS secret key (at the ith leaf), the ith OTS public key, and the so-called authentication path of the latter, which includes all $n + 1$ intermediate nodes on the path from the ith leaf to the root. MSS only requires a secure hash function to guarantee the overall security of the scheme, but has the disadvantage of large signature sizes. Two recent developments in this area of research are an improved variant of MSS called XMSS and the stateless signature scheme SPHINCS.

Code-based cryptosystems use an error correcting code to construct a one-way function; their security is based on the hardness of decoding a message that contains random errors and recovering the code structure. A well-known example is the McEliece public key encryption scheme, which employs binary Goppa codes due to their high error correction capability. The private key is a binary irreducible degree-t Goppa code chosen secretly by the receiver of the message, and the corresponding public key is a generator matrix G describing a scrambled and randomly permuted version of this code. To encrypt a message, the sender first encodes it using G and then adds t random errors. Only the legitimate receiver, who knows the hidden algebraic structure of the Goppa code, can correct the errors and recover the message. Even though the McEliece scheme has been analyzed for 40 years, no serious weaknesses are known, and this is expected to remain so in the quantum-computing era. The Niederreiter cryptosystem is a McEliece variant that can serve as the basis of both encryption and signature schemes. Unfortunately, all McEliece variants have relatively large public keys (up to 10^6 bits).

Lattice-based cryptosystems are promising PQC candidates because some of them combine strong security guarantees in the form of a worst-to-average case reduction with high efficiency and small key and ciphertext/signature sizes. Examples of lattice-based cryptosystems include the NTRU encryption scheme (whose security is related to the shortest vector and closest vector problem in a special kind of lattices) as well as encryption, key exchange, and signature schemes built on the hardness of the Learning With Errors (LWE) problem and its ring variant, the RLWE problem [6]. The latter cryptosystems operate in a polynomial ring $\mathcal{R}_q = \mathbb{Z}_q[x] \langle f \rangle$ where f is an irreducible polynomial. RLWE-based key exchange protocols specify a set of public system parameters that define besides $\mathcal{R}_q$ also a fixed polyno-

The focus of research activities is currently directed toward four main categories: lattice-based cryptography, multivariate cryptography, hash-based cryptography, and code-based cryptography. These four categories differ greatly in terms of the underlying hard mathematical problem and also with respect to performance, key lengths, and the length of ciphertexts as well as digital signatures.

Operation	Type	Algorithms and implementations
Symmetric encryption	Block/stream ciphers	AES-256, Salsa20
Public key encryption	Lattice-based	[2, 3, 5]
	MPKC	SimpleMatrix, ZHFE, PMI+, IPHFE+
	Code-based	McEliece, Niederreiter
Public key signature	Lattice-based	BLISS ([4, 12, 13]), GPV, GLP [11], NTRUSign, ring-TESLA [14], Tesla] [15]
	MPKC	UOV, Rainbow, TTS, HFEv-, GUI
	Hash-based	XMSS, SPHINCS-256, Lamport, Merkle
	Code-based	Niederreiter
Key exchange	Lattice-based	[6, 8, 9, 10]

Table 1. Major post-quantum cryptography primitive constructions: a comparative summary.

mial $\mathbf{a} \in \mathcal{R}_q$ and the parameters for a discrete Gaussian distribution χ used to sample polynomials with "small" coefficients from $\mathcal{R}_q$. Each of the two involved entities samples a secret polynomial **s** and an error polynomial **e** from χ, computes a public key $\mathbf{b} = \mathbf{as} + \mathbf{e} \in \mathcal{R}_q$, and sends it to the other entity. Then each entity multiplies the received public key **b** by its secret polynomial **s**, similar as in Diffie-Hellman key exchange, to arrive at an approximate or "noisy" agreement on a shared secret (i.e., the coefficients of the polynomials obtained on both sides differ slightly due to **e**. Finally, a reconciliation mechanism is applied to ensure both entities reach an exact (i.e., identical) shared secret.

Table 1 gives an overview of the four outlined approaches for PQC and provides references to implementations.

IoT Processors

At the time of this research, many widely used low-end IoT devices use an 8-bit AVR microcontroller (e.g. Arduino UNO), and the latter has an 8-bit RISC instruction set and a modified Harvard architecture that features 32 8-bit general-purpose registers denoted by `r0–r31`. From this pool of registers, the last three pairs, `X` (`r27:r26`), `Y` (`r29:r28`), and `Z` (`r31:r30`) are used as 16-bit address pointers to load and store data from memory. The AVR instruction set supports a total of 133 instructions, and each instruction has a fixed latency. For example, arithmetic/logical instructions, such as addition (`ADD`) and addition with carry (`ADC`), are executed in a single clock cycle. Unsigned multiplication (`MUL`) and load/store instructions take two clock cycles. The Cortex-M4 is part of the increasingly popular Cortex-M family, which includes a wide range of 32-bit RISC ARM microcontrollers. Cortex-M4 supports the ARMv7E-M instruction set, comprising Thumb-2 instructions and additional saturating/SIMD instructions, namely the "DSP extension." The Cortex-M4 architecture has a three-stage pipeline with branch speculation, includes 16–32-bit registers (`r0~r15`), and supports a mix of 16- and 32-bit operations corresponding to the Thumb-2 instruction set. Examples of the instructions are the 32-bit arithmetic/logical instructions such as addition (`ADD`), addition with carry (`ADC`), as well as memory instructions that perform a single-data loading/storing (`LDR/STR`) or multiple-data loading/storing (`LDM/STM`). It also supports the powerful single-cycle multiply and multiply-and-accumulate instructions from the DSP extension, namely `UMUL`, `UMLAL`, and `UMAAL`. These instructions execute a 32 × 32-bit computation resulting in a 64-bit value, plus a 64-bit accumulation with a single 64-bit value (`UMLAL`) or a 64-bit accumulation with two 32-bit values (`UMAAL`).

Implementation of Lattice-Based Cryptography on IoT Devices

Ring-LWE Encryption Schemes

The first practical software implementation of a public key cryptosystem based on the learning with errors (LWE) problem is reported by Göttert *et al.* [2]. They assessed the practicality of ring-LWE encryption, and presented both hardware and software implementations. In particular, for software implementation, they gave a comparison between a matrix and a polynomial-based variant of the LWE scheme. The authors employed the fast Fourier transform (FFT) to speed up multiplication in polynomial rings, which is the most critical operation in lattice-based cryptography.

De Clercq *et al.* [3] provided an improved implementation of the ring-LWE encryption scheme on a 32-bit ARM processor. They introduced two optimization techniques. First, they used the Knuth-Yao sampling algorithm for a fast discrete Gaussian sampler. To generate the random numbers, the target ARM processor's built-in true random number generator (TRNG) is exploited. For high-speed sampling, Gaussian distribution is obtained from pre-computed look-up tables. Second, they used the negative-wrapped NTT and stored the multiple coefficients in each processor word for efficient polynomial multiplication. Finally, the implementation requires 121k clock cycles per encryption and 43.3k cycles per decryption at a medium-term security level, as well as 261k cycles per encryption and roughly 96.5k cycles per decryption for long-term security. Their results represent the current state of the art in efficient implementation of ring LWE encryption on a 32-bit processor.

Liu *et al.* [5] focused on efficient arithmetic techniques for the ring variant of the LWE encryption scheme on 8-bit AVR. Their contributions include several optimizations for improving the execution time of the number theoretic transform (NTT) based on polynomial multiplication and the memory requirements of the coefficient. For Gaussian sampling, the byte-wise scanning for the Knuth-Yao Gaussian distribution sampler is proposed to improve performance. In particular, for the 8-bit AVR processor, they proposed the MOV-and-ADD technique for coefficient multiplication and the shifting-addition-multiplication-subtraction-subtraction (SAMS2) technique for modular reduction, which optimizes the instruction sets. Later, they extended the work on the 32-bit ARM-NEON processor, and the authors proposed a parallel NTT to reduce the execution time for

coefficient multiplication, which introduces four-way NTT computations over the SIMD architecture. For fast modular reduction, a 32-bit-wise SAMS2 method is efficiently implemented. The random number is efficiently generated through a block-cipher-based pseudo random number generator (PRNG).

Lattice-Based (Authenticated) Key Exchange Protocols

Ding *et al.* [6] proposed the first LWE and RLWE-based provably secure key exchange protocol. This work is the foundation of current LWE and RLWE key exchange protocols. The authors leverage the property of commutativity and the notion of approximate equivalence to construct key exchange protocols over LWE and RLWE. They then design an error reconciliation mechanism and send signal from one side to the other to reconcile error between two close values. In order to reconcile errors with an overwhelming probability, the norm of difference is strictly bounded by choosing modulus q carefully. All reconciliation-based protocols follow the same idea to construct variants of this work. Boorghany *et al.* [7] implemented lattice-based authenticated key exchange (AKE) protocols on both 8-bit AVR and 32-bit ARM processors, where they used FFT instead of NTT to optimize the number of transformations.

Zhang *et al.* [8] introduced the first practical and provably secure two-pass AKE protocol from ideal lattices. It is an RLWE variant of classical HMQV protocol. The security is demonstrated under the Bellare–Rogaway model with weak perfect forward secrecy. The authors also provided a one-pass variant of their two-pass protocol for specific applications. Parameter choices between 80- and 360-bit security are provided. The proof-of-concept implementation demonstrated the utility of the post-quantum RLWE-based AKE protocol.

Bos *et al.* [9] instantiated Peikert's RLWE key exchange protocol with a 128-bit secure parameter choice. Peikert's key exchange is almost the same as the scheme reported in [6]. Bos *et al.* provided proof-of-concept implementation of Peikert's protocol and their parameter choice. They further integrated the protocol into OpenSSL and combined quantum-insecure digital signatures (ECDSA or RSA) as post-quantum TLS ciphersuites `RLWE-ECDSA (RSA)-AES128-GCM-SHA256`. They then proved the security of their ciphersuites in the authenticated and confidential channel establishment (ACCE) model. The authors also demonstrated that their implementation and post-quantum TLS ciphersuite are efficient, which suggested the costs of transitioning from quantum-insecure cryptographic primitives to quantum-safe primitives is not too high, and real-world post-quantum application for our digitized society is practical.

Alkim *et al.* [10] improved the scheme presented in [9], in the sense of using more compact parameter choices, adopting a different error distribution that is less expensive to sample, presenting a more comprehensive and strict security analysis on their protocol, and proposing a new and more efficient error reconciliation mechanism and a technique to defend against possible backdoors in public parameters. A follow-up work implementation on an ARM Cortex-M0 (ARMv6-M) processor was presented in [10].

Lattice-based Signature Scheme

Güneysu *et al.* [11] presented a signature scheme (GLP), whose security relies on the hardness of lattice problems. Oder *et al.* [12] described an efficient implementation of BLISS [13] on a 32-bit ARM Cortex-M4F microcontroller. They investigated three different samplings, namely Bernoulli, Knuth-Yao, and Ziggurat. For polynomial arithmetic, NTT and sparse multiplication methods were studied. They achieved execution times of 35.3 ms and 6 ms for signature generation and verification, respectively, at a medium-term security level. BLISS-B, an improvement of BLISS, speeds up key generation by a factor of 5–10 and signing by a factor 2–3, while maintaining the same security level as BLISS. Pöppelmann *et al.* [4] optimized the BLISS signature scheme on an 8-bit AVR architecture, where they merged certain multiplication operations and removed the expensive bit-reversal step. The compact implementation only requires 329 ms and 88 ms for signature generation and verification, respectively.

In 2016, Akleylek *et al.* [14] proposed ring-TESLA, the first provably secure lattice-based signature scheme with good performance. They provided a tight security reduction for the new scheme from the ring-LWE problem, which allows for a provably secure but efficient instantiation. The experimental results from a software implementation demonstrated that ring-TESLA performs comparably to both GLP and BLISS schemes. Also in the same year, Barreto *et al.* [15] presented an improved scheme of [14] — Tesla#. The latter is a digital signature scheme based on the RLWE assumption, which achieves much faster key pair generation, signing, and verification. It also outperforms most (conventional and lattice-based) signature schemes on modern processors. We will now summarize the performance of these three state-of-the-art implementations for lattice-based encryption [5], signature [12], and key exchange ([10] on ARM) schemes on IoT devices in Table 2. All numbers reported in Table 2 are clock cycles on the respective processors. It is clear that implementations of lattice-based constructions on IoT devices can be efficient, in the sense that it is possible to implement post-quantum cryptographic schemes on resource-constrained devices. Also, according to Table 4 of [5] and Table 3 of [4], the implementation of the respective scheme is even faster than 1024-bit RSA on an Atmel ATmega128 processor at 8 MHz.

Given the potential of lattice-based cryptography in post-quantum systems, it is likely that we will see more advanced implementation of lattice-based cryptography primitives and high-level applications in future IoT deployments.

Conclusions

With the increasing prevalence of IoT devices in countless applications, ranging from home automation over health care to traffic control, the ability to securely collect and analyze data becomes more and more important. Edge computing has the potential to alleviate some security and privacy concerns associated with the IoT by distributing data processing and decision making toward the

With the increasing prevalence of IoT devices in consumer, business, government and military applications, the ability to reliably and securely transmit, store, and analyze sensitive data in and between IoT devices and architecture is important not only to businesses and users, but also to our national security.

With continuing advances in quantum computing, lattice-based cryptography will play an increasingly important role in (real-world) post-quantum applications due to its versatility, high efficiency, and relatively small key size and communication cost.

Usage	Implementation	Platform	KeyGen	Encryption	Decryption	Security
Encryption	[3]	ARM Cortex-M4F	116,772	121,166	43,324	128
	[5]	ATxmega128A1	589,900	671,628	275,646	128
			KeyGen	**Signing**	**Verifying**	**Security**
Signature	[12]	ARM Cortex-M4F	367,859,092	5,984,686	1,002,299	128
	[4]	ATxmega128A1		10,537,981	2,814,118	128
			Client	**Server**		**Security**
Key exchange	[10]	ARM Cortex-M0	1,760,837	1,467,769		256

Table 2. Implementations of lattice-based cryptography on IoT devices: a comparative summary.

edge of the network rather than allocating these tasks exclusively on a centralized cloud platform. However, it is essential to encrypt the communication between the IoT devices and the edge gateways, especially in human-driven edge computing when personal devices such as smartphones form part of the network or when the collected data contains sensitive personal information. In certain cases (e.g., healthcare applications), long-term protection of the data for ten or more years is required, which is only possible with post-quantum cryptosystems. Using classical algorithms such as RSA, DH, and ECDH for key establishment bears the risk that an attacker with the capability to eavesdrop on and store the communication between the devices will be able to break the encryption in the not-so-distant future when large quantum computers become available.

We give a succinct overview of four important approaches for the design of post-quantum cryptosystems and make the point that lattice-based cryptography provides a combination of desirable properties. On one hand, some lattice-based cryptosystems, including several primitives based on the LWE problem and its variants, come with strong security guarantees backed by a worst-case to average-case security reduction. On the other hand, a number of RLWE-based cryptosystems are very efficient, as our survey of recent implementation papers shows, and provide (relatively) short keys as well as small ciphertext and signature sizes. Therefore, it can be expected that RLWE-based cryptosystems will play an essential role in post-quantum edge computing and the post-quantum IoT.

References

[1] C. Cheng *et al.*, "Securing the Internet of Things in a Quantum World," *IEEE Commun. Mag.*, vol. 55, no. 2, Feb. 2017, pp. 116–20.

[2] N. Göttert *et al.*, "On the Design of Hardware Building Blocks for Modern Lattice-Based Encryption Schemes," *Int'l. Wksp. Cryptographic Hardware and Embedded Systems*, Springer, 2012, pp. 512–29.

[3] R. De Clercq *et al.*, "Efficient Software Implementation of Ring-LWE Encryption," *Proc. 2015 Design, Automation & Test in Europe Conference & Exhibition*, 2015, pp. 339–44.

[4] T. Pöppelmann *et al.*, "High-Performance Ideal Lattice-Based Cryptography on 8-Bit ATxmega Microcontrollers," *Latincrypt '15*, Springer, 2015.

[5] Z. Liu *et al.*, "Efficient Implementation of Ring-LWE Encryption on 8-bit AVR Processors," *Int'l. Wksp. Cryptographic Hardware and Embedded Systems*, Springer, 2015, pp. 663–82.

[6] J. Ding *et al.*, "A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem"; https://eprint.iacr.org/2012/688, accessed 31 Aug., 2017.

[7] A. Boorghany *et al.*, "On Constrained Implementation of Lattice-Based Cryptographic Primitives and Schemes on Smart Cards," ACM Trans. Embedded Computing Sys., vol. 14, no. 3, 2015, Acticle 42.

[8] J. Zhang *et al.*, "Authenticated Key Exchange from Ideal Lattices," *Annual Int'l. Conf. Theory and Applications of Cryptographic Techniques*, Springer, pp. 719-751.

[9] J. Bos *et al.*, "Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem," *2015 IEEE Symp. Security and Privacy*, 2015, pp. 553–70.

[10] E. Alkim *et al.*, "Post-Quantum Key Exchange — A New Hope," *USENIX Security Symp.*, 2016, pp. 327–43.

[11] T. Güneysu *et al.* "Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems," *Int'l. Wksp. Cryptographic Hardware and Embedded Systems*, Springer, 2012, pp. 530–47.

[12] T. Oder *et al.*, "Beyond ECDSA and RSA: Lattice-Based Digital Signatures on Constrained Devices," *Proc. 51st ACM Annual Design Automation Conf.*, 2014, pp. 1–6.

[13] L. Ducas *et al.* "Lattice Signatures and Bimodal Gaussians," *Advances in Cryptology*, Springer, 2013, pp. 40–56.

[14] S. Akleylek *et al.*, "An Efficient Lattice-Based Signature Scheme with Provably Secure Instantiation," *Progress in Cryptology – AFRICACRYPT 2016*, Springer, 2016, pp. 44-60.

[15] P. S. Barreto *et al.*, "Sharper Ring-LWE Signatures"; https://eprint.iacr.org/2016/1026/20161101:020659, accessed 31 Aug., 2017.

Biographies

Zhe Liu (sduliuzhe@gmail.com) is a full professor in the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. He is also a research fellow in SnT, University of Luxembourg. He received his Ph.D degree from the Applied Cryptography Group, University of Luxembourg in 2015 and the prestigious FNR Outstanding Ph.D Thesis Award in 2016. His research areas include computer arithmetic and cryptographic engineering for pre-quantum and post-quantum cryptography.

Kim-Kwang Raymond Choo [SM'15] holds the Cloud Technology Endowed Professorship at the University of Texas at San Antonio. He is the recipient of the ESORICS 2015 Best Paper Award, was a member of the 2015 Winning Team of Germany's University of Erlangen-Nuremberg Digital Forensics Research Challenge, and received the 2014 Australia New Zealand Policing Advisory Agency's Highly Commended Award, the 2010 Australian Capital Territory Pearcey Award, a Fulbright Scholarship, the 2008 Australia Day Achievement Medallion, and the British Computer Society's Wilkes Award.

Johann Großschädl is a member of research staff at LACS, University of Luxembourg. Before joining the University of Luxembourg, he was a research scientist in the Computer Science Department of the University of Bristol, United Kingdom. He has published more than 70 papers in international, peer-reviewed journals and conference proceedings, such as ACSAC and CHES, which are the flagship events in the field of applied cryptography.