

Toward Postquantum Security for Embedded Cores

**Rafael Misoczki, Sean Gulley, Vinodh Gopal,
Martin G. Dixon, Hrvoje Vrsalovic, and
Wajdi K. Feghali**
Intel Corporation

■ **THE USE OF** firmware agents—including their use to define the functionality of embedded cores—has proliferated on computer systems of all scales, especially servers. The agents are often not visible to the operating system as they independently perform configuration, monitoring, and certain control tasks. For example, the baseboard management controller (BMC) on server platforms runs a firmware stack (e.g., OpenBMC (<https://github.com/openbmc/openbmc>)), has a network port, some peripherals on external buses (e.g., I2 C's, SPI, etc.), and storage. The BMC is effectively another full (but scaled-down) computing system on the server of which it is a component. While the BMC can directly affect the operation of a server (e.g., by controlling its power states), it does not interact with the OS. Its behavior is defined completely by its firmware.

Improvements in silicon manufacturing processes have reduced the size of these cores to the point where they can be physically embedded within silicon dies of the system components (such as a CPU) on which they operate. In parallel, the volume and importance of their responsibilities have increased, beginning with power management and escalating to security operations that can affect functional safety. Given this, it is critical that they run only signed and authenticated code.

One of today's best practices to authenticate the firmware that runs embedded cores is public key cryptography (digital signatures), which relies on FIPS-140 digital signature algorithms such as RSA and EC-DSA. However, quantum computing will render these algorithms useless since factorizing integers and solving the discrete logarithm problem (i.e., the underlying security problems of RSA and EC-DSA) will be solvable in polynomial time.¹¹ This implies that increasing RSA/ECC key sizes will be insufficient to defeat a quantum adversary. Prof. Michele

Digital Object Identifier 10.1109/MM.2019.2920203

Date of current version 23 July 2019.

Mosca, from the Institute of Quantum Computing of the University of Waterloo, predicts that there is a 50% chance that RSA-2048 public key cryptosystem will be broken by 2031 by quantum computers. In response to this threat, the National Institute of Standards and Technology (NIST) has started a competition for standardizing post-quantum cryptography (PQC). Additionally, Grover's algorithm⁵ challenges AES-128 and symmetric cryptography algorithms of short key/digest size, in general.

In response, cryptographers have proposed algorithms that can be classified into several families. Since embedded cores cannot house large keys and are not typically high-performance to handle complex operations, one interesting question to ask is how suitable these PQC algorithms are for embedded codes. In this paper, we will discuss the proliferation of embedded cores, why security depends upon the firmware running on them, provide an overview of the existing proposals for postquantum digital signatures, discuss what approaches seem the most reasonable for embedded cores, and discuss Intel's current direction.

FIRMWARE AGENT PROLIFERATION

In the 1990s, a processor such as the Pentium Pro Processor from Intel began using a firmware binary that defined or redefined certain operations of that CPU, which came to be known as Intel Microcode (or simply *microcode*). As the complexity of CPU and chipset designs grew, Intel added a separate microcontroller core (Foxton) to the design of the Core i7 processor code-named Nehalem. This microcontroller was tasked with power management, and it ran in conjunction with the microcode. At the same time, the Platform Controller Hub began to add microcontrollers for audio, power management, and manageability. SoC's from the rest of the industry followed the same paradigm, embedding

One of today's best practices to authenticate the firmware that runs embedded cores is public key cryptography (digital signatures), which relies on FIPS-140 digital signature algorithms such as RSA and ECDSA. However, quantum computing will render these algorithms useless since factorizing integers and solving the discrete logarithm problem (i.e., the underlying security problems of RSA and ECDSA) will be solvable in polynomial time.

microcontrollers *in lieu* of hard-wired logic. Today, the same processors contain even more embedded agents.

In a typical new design from Intel, there are microcontrollers to handle: USB Type-C port switching, reconfiguration of analog lanes, the display engine, graphics memory translations, temperature compensation for analog circuits, hardware and software debugging support, image processing, asset administration and manageability (such as Intel's AMT features), security, packet processing, and more. These microcontrollers exist physically discrete in the silicon dies and execute independently of the execution of the primary

Intel CPU(s) or governance by the operating system; their functionality and behavior are controlled by firmware that is usually contained in flash memory local to the microcontroller core itself (i.e., embedded in the same silicon die).

The proliferation of microcontrollers as embedded agents has been enabled by multiple factors: the availability of controllers in a variety of sizes due to improvements in manufacturing processes, the desire for flexibility of designs, and demand for in-the-field updates. Microcontroller cores are available in sizes ranging from ten thousand gates to hundreds of thousands of gates. For example, an exemplary microcontroller with 64 KB of SRAM occupies approximately 0.1 mm² on a 10-nm process or around 0.1% of a client processor. Software tools and runtimes for popular cores (e.g., ARM-based) allow sophisticated development flows similar to regular desktop/server software; system designers can introduce new functionality to a computer system through these controllers using modern, high-level languages (e.g., Rust). This late-binding flexibility compensates for the lengthening of hardware development cycles and fabrication times, leading to more firmware agents working in conjunction in these designs. On the other

hand, such proliferation of the embedding of microcontrollers, their increased handling of operation-critical tasks for the overall system, and the easy access to sophisticated development tools have made microcontroller cores' firmware an attractive target for malfeasance. For product assurance, the firmware needs to be authenticated and protected against tampering.

FIRMWARE SIGNATURES

To assert that the code running on a given microcontroller originated from the expected source (the vendor) and has not been tampered since it was deployed, the firmware image needs to have ways by which it can be signed and attested; there must exist the ability to assure that the firmware originated from a specific vendor (e.g., Intel) and that the bits that comprise the firmware binary have not been altered by a 3rd party from the time it was installed to the time it is being loaded. This is commonly done with digital signature verification. A simplified example of performing this verification is to sign a message digest (or hash) of the firmware binary with a vendor private key and append it to the firmware. A system that has access to the firmware vendor's public key can then correctly verify the authenticity of this hash value. Any signature verification failure signals a modified firmware or the use of a foreign private key (i.e., the signature did not originate from the vendor). Thus, this mechanism prevents attackers from generating valid signatures of modified firmware. In practice, firmware signature verification is complicated by the constraints on the environment in which the verification is to take place; since it usually occurs (at least once) very early in the system boot process, limited memory, and/or computational resources are available. To provide a tangible, real-world example of firmware signatures, we discuss Intel's microcode patch.

Intel Microcode Signing

The constraints on microcode are significant. Microcode may be loaded multiple times from the power-on of a processor, and since it must be verified each time, the duration to

decrypt and verify is critical to system responsiveness. Additionally, since main memory is not available in the earliest stages of initialization, microcode verification is done within the processor's cache so it must fit into a relatively small footprint (e.g., 16 to 64 KB). Microcode as a firmware mainly differs from a traditional microcontroller firmware image by its target processor core: The Core and Atom processors are multiple orders of magnitude higher performance than a typically embedded microcontroller core. While microcode, having the benefit of a much faster execution via the main CPU, could potentially adopt much more cryptographically stronger (but more computation-resource demanding) algorithms, those same algorithms may be unsuitable for the more constrained embedded cores.

STATE OF ART ON POSTQUANTUM DIGITAL SIGNATURES

There has been a recent surge in activity in the field of PQC, with several new schemes proposed every year and a rapidly growing community of researchers that scrutinize the robustness, performance and ease of use of new and well-established postquantum cryptosystems. This increased interest may be related to the recently established standardization processes on PQC. The National Institute of Standardization and Technology started a project to analyze possible PQC candidates for standardization. The Internet Engineering Task Force (IETF) has recently published Request for Comments (RFC's) informational documents that specify stateful postquantum digital signatures such as the XMSS scheme.¹ Experts from the International Standards Organization have also been working on a standing document on PQC and, more recently, on a study-period on stateful hash-based signatures. In this section, we provide a brief analysis of the maturity, robustness, and performance of the signature schemes considered in the aforementioned PQC standardization processes.

Since most of the schemes discussed in this section have been submitted to the NIST PQC project, we will give some additional context about this process. In November 2017, NIST

received 19 submissions on postquantum stateless digital signatures for the 1st round of their PQC standardization competition. Among those, five were lattice-based, two code-based, seven multivariate-quadratic based, three symmetric-crypto-based schemes, and two others that could not be classified in any of the previous categories. In January 2019, NIST selected only nine submissions to pass on to the second round of their competition. Among those, three are lattice-based, four are multivariate-quadratic based, and two are symmetric-crypto-based schemes. According to NIST, the evaluation criteria used to select the second round candidates was security (formal security proof and resistance to side-channel attacks), cost and performance (size of public key and signature, computational efficiency, and probability of failures), and algorithm and implementation characteristics (parameters flexibility, parallelism amenability, and simplicity). NIST defined a few security levels for their competition. Level 1 should match the postquantum security of AES128, level 3 should match the postquantum security of AES192, and level 5 should match the postquantum security of AES256. Our performance analysis focuses on parameter sets that achieve security levels 1 and 5 (whenever possible), and the reference implementations written by the scheme designers.

Symmetric-Crypto-Based Schemes

From a security perspective, these are the most conservative schemes since their security relies uniquely on the security of hash or block ciphers, thus they do not introduce any additional security assumptions. This category splits into two subcategories: stateful and stateless schemes. Stateful signature schemes require the secure storage of some state (data) in between signatures generation. Stateless signature schemes do not have this additional requirement. For example, RSA and ECDSA are stateless schemes.

In case state management is a doable task, i.e., maintaining a piece of data securely in between signature generation, the stateful schemes should be considered. We remark that code signing applications (such as firmware

authentication) seem to be among the most suitable applications for stateful schemes since the manufacturer can carefully implement state management.

The **XMSS Scheme**¹ is a stateful hash-based signature that has been recently published as an informational RFC by the IETF. It can be regarded as an evolved version of the classical Merkle scheme. From a security perspective, XMSS enjoys a security proof in the standard model based on mild security assumptions from the hash function (e.g., pre-image and target collision resistance). From a performance perspective, since full collision resistance is not needed, XMSS can operate with smaller hash digests than the Merkle scheme. This leads to shorter signatures and faster processing. For 128 bits of quantum security, XMSS requires 64 bytes of a public key and 2.44 KB for the signature, and verification takes about 760,000 cycles in a modern processor. For 64 bits of quantum security, XMSS requires 32 bytes of the public key and 740 bytes of signatures, and verification takes about 390,000 cycles in the same machine.

In the stateless subcategory, we consider SPHINCS+ and PICNIC. **SPHINCS+**¹² is a stateless hash-based signature scheme that uses huge Merkle trees (e.g., height of 60). It is stateless because it selects the Merkle leaf nodes at random, and the chance of accidentally reusing the same leaf node twice (which would void its security guarantees) is negligible given the huge number of leaf nodes (e.g., 2^{60}). SPHINCS+ offers two sets of parameters per security level (thus six in total), one set optimized for speed and another optimized for small signatures. We focus our analysis on the speed optimized parameters using SHA-256. For level 1, SPHINCS+ offers signatures of 16.57 KB, public keys of 32 bytes; signing takes 340 million cycles, and verification takes 14.98 million cycles (“SPHINCS+–SHA-256-128f-robust” parameter set). For level 5, these numbers grow to 48.06 KB, 64 bytes; 1,491.94 million and 37.59 million cycles (“SPHINCS+–SHA-256-256f-robust” parameter set), respectively. **PICNIC**⁸ is a signature algorithm based on a block cipher (LowMC) and a noninteractive

proof of knowledge. LowMC has been chosen as the underlying block cipher since it gives smaller signature sizes. Still, the signature for level 1 is 33.23 KB and the public key size is 32 bytes (“picnic-L1-FS” parameter set). For level 5, signature size is 129.74 KB and the public key size is 64 bytes (“picnic-L5-FS” parameter set). For level 1, signing takes 137.91 million cycles and verification takes 90.63 million cycles, while for level 5 it takes 1,112.23 million and 736.31 million cycles, respectively.

The symmetric-crypto-based candidates have very strong security guarantees. In case state management is possible, stateful HBS schemes may be the most promising approach since they are interesting from both security and performance perspectives. It is worth mentioning that digital signatures applied to verify firmware authenticity of embedded cores seem one application where state management seems possible. The fact that the signatures are generated by manufacturers (and not end users) that can afford a robust signing facility with state management capabilities seems to facilitate the adoption of stateful HBS schemes in this scenario. Also, the other limitation of certain stateful HBS schemes that can issue a limited number of signatures does not seem a problem since manufacturers can, most of the times, predict how many firmware updates a device will receive during its lifetime. On the other hand, in case state management is not possible, stateless schemes may be advisable; however, they are less attractive from a performance perspective from both size and speed metrics.

Lattice-Based Schemes

In this category, we have Falcon,³ CRYSTALS-DILITHIUM,² and qTesla.⁹ From the PQC families that introduce additional security assumptions, lattices are one of the most popular approaches. From a side-channel perspective, the Gaussian sampling process seems to offer some challenges to be implemented in a side-channel resilient way.

Falcon is based on the Short Integer Solution problem, known in the crypto community for some time, but it is applied to (structured) NTRU lattices. For level 1, Falcon signatures have 617 bytes and the public key has 897

bytes, signing takes 542,000 cycles and verification takes 88,000 cycles. For level 5, these numbers change to 1.20 KB, 1.75 KB, 1.07 million cycles, and 186,000 cycles, respectively.

CRYSTALS-DILITHIUM uses module lattices problems, which can be viewed in between the ones used in Learning-With-Errors (LWE) and Ring-LWE problems. In other words, according to the authors, they are just as efficient as Ring-LWE schemes but closer to the (stronger, more conservative) LWE underlying security problem. For levels 1 and 3, CRYSTALS-DILITHIUM offers signatures of size 1.99 and 3.28 KB, and public keys of size 1.15 and 1.71 KB, respectively. The authors did not provide parameters for level 5. In terms of speed, for level 1, it takes 1.3 million cycles to sign and 272,000 cycles to verify. For level 3, it takes 1.82 million and 510,000 cycles, respectively.

qTesla is based on the well-known Ring-LWE problem, and it offers good performance. On April 14, 2019, researchers presented a potential attack against qTesla that may affect some of their parameter sets (qTesla’s authors have yet to respond). From a performance perspective, for level 1, qTesla offers signatures of size 1.3 KB and public keys of size 1.5 KB, and for level 5 signatures of size 5.9 KB and public keys of size 6.4 KB. Regarding speed, for level 1, signature generation takes 492,000 cycles and verification takes 82,000 cycles, while for level 5, signature generation takes 2.1 million cycles and verification takes 394,000 cycles.

The lattice-based cryptography field is a popular PQC approach. One point of attention is the secure selection of parameters that still seems to be challenging depending on the underlying security problem. From a performance perspective, all three lattice candidates offer reasonable performance and should be considered promising candidates.

Multivariate Quadratic (MQ)-Based Schemes

In this category, we have GeMSS,⁴ Rainbow,¹⁰ LUOV,⁶ and MQDSS.⁷ Several MQ schemes have been proposed in the past and subsequently broken. Security has become more stable in recent years, however MQ remains the PQC family of digital signatures whose security is the

Table 1. Comparison for security level 1 or the closest. Sizes in KB, speed in millions of cycles.

	Symmetric Crypto			Lattices			Multivariate Quadratic			
	<u>XMSS</u>	<u>SPHINCS+</u>	<u>PICNIC</u>	<u>Falcon</u>	<u>CRYSTALS-DILITHIUM</u>	<u>qTesla</u>	<u>GeMSS</u>	<u>Rainbow</u>	<u>LUOV</u>	<u>MQDSS</u>
Signature size	0.72	16.57	33.23	0.60	1.99	1.37	0.03	0.06	0.30	20.36
Public key size	0.03	0.03	0.03	0.87	1.15	1.50	417.40	149.00	12.10	0.04
Signing speed	–	340.00	137.91	0.54	1.37	0.49	690.00	0.40	5.40	26.63
Verification speed	0.39	14.98	90.63	0.08	0.27	0.08	29.10	0.15	4.30	19.84

least understood. The main benefit of MQ schemes is the compact signature sizes.

GeMSS is a scheme based on the hidden field equations underlying problem and can be seen as a variant of Quartz, a scheme proposed in 2001, which remains one of the fewest unbroken MQ schemes. GeMSS signatures are very compact: only 258 bits for level 1, and 588 bits for level 5. Public key sizes are 417 KB and 3,046.84 KB, respectively. However, GeMSS is not speed efficient: for level 1, signing takes 6,690 million cycles and verification takes 29 million cycles, while for level 5, signing takes 25,300 million cycles and verification takes 172 million cycles.

Rainbow is a signature scheme based on the well-known Unbalanced-Oil-and-Vinegar (UOV) signature scheme (which itself is based on the Oil-and-Vinegar scheme). From a practical perspective, for level 1, the signature is 512 bits long and the public key is 149.00 KB long, while signing takes 402,000 cycles and verification takes 155,000 cycles. For level 5, the signature is 159 KB and the public key is 1,227.10 KB long, while signing takes 3.6 million cycles and verification takes 2.3 million cycles.

Lifted-UOV (LUOV) is a scheme also based on the UOV scheme. The main difference from UOV consists of some optimizations to reduce the public key size (e.g., lifting the UOV public key to an extension field). For level 2, signature and public key sizes are 311 bytes and 12.1 KB, respectively. Signing takes 5.4 million cycles and verification takes 4.3 million cycles. For level 5, signature and public keys are 494 bytes

and 75.5 KB, respectively, while signing takes 24 million cycles and verification takes 18 million cycles.

MQDSS is a scheme based on the combination of the Sakumoto–Shirai–Hiwatari (SSH) identification scheme with the Fiat-Shamir transform. This is a very innovative proposal. For level 1, its signature and public key sizes are 20 KB and 46 bytes long, respectively, while signing takes 26 million cycles and verification takes 19 million cycles. For level 3, its signature and public key sizes are 42 KB and 64 bytes long, respectively, while signing takes 85 million cycles and verification takes 62 million cycles.

MQ-based schemes offer interesting performance benefits, such as tiny signatures from GeMSS or tiny public keys from MQDSS. However, the field of multivariate-quadratic schemes would still benefit from a more comprehensive security analysis. The PQC standardization process may help in this process by promoting these schemes and thus attracting an increasing number of researchers to expand the knowledge in this field and increase the confidence of potential users.

Tables 1 and 2 show the performance of the second round candidates of the NIST PQC competition plus the XMSS scheme published in IETF RFC8391. We acknowledge that these numbers (most of them obtained from the submission packages) were collected in different platforms, and therefore the speed numbers should be taken as a rough approximation of the actual performance, useful when considered from the orders-of-magnitude perspective.

Table 2. Comparison for security level 5 or the closest. Sizes in KB, speed in millions of cycles.

	Symmetric Crypto			Lattices			Multivariate Quadratic			
	<u>XMSS</u>	<u>SPHINCS+</u>	<u>PICNIC</u>	<u>Falcon</u>	<u>CRYSTALS-DILITHIUM</u>	<u>qTesla</u>	<u>GeMSS</u>	<u>Rainbow</u>	<u>LUOV</u>	<u>MQDSS</u>
Signature size	2.44	48.06	129.74	1.20	3.28	5.92	0.07	1.59	0.5	42.70
Public key size	0.06	0.06	0.06	1.75	1.71	6.43	3,046.84	1,227.10	75.50	0.06
Signing speed	–	1,491.94	1,112.23,	1.07	1.82	2.15	25,300	3.64	24.00	85.26
Verification speed	0.74	37.59	736.31	0.18	0.51	0.39	172.00	2.39	18.00	62.30

Analysis of Available PQC Solutions

The most important criterion for the selection of PQC schemes should be security. The PQC transition offers considerably greater challenges than previous (symmetric) cryptographic transitions (for example, from 3DES to AES, or SHA-1 to SHA-2). Public key cryptosystems are considerably more complex than symmetric ones, and we still do not fully understand the capabilities of a typical, future quantum adversary. In this context, conservativeness seems to be the most reasonable stance.

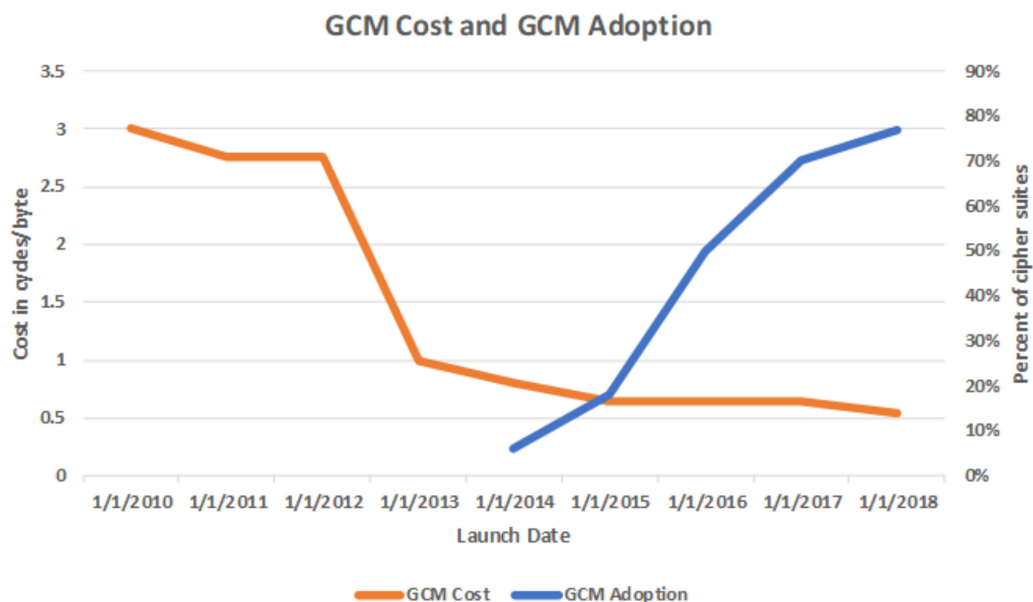
Schemes based on symmetric crypto building blocks (e.g., hash or block ciphers) seem to be an extremely promising approach, as they do not introduce any additional security assumptions. In cases where state management is possible, XMSS is a strong candidate as it offers interesting performance and security proof in the standard model. In case state management is not possible, SPHINCS+ may be preferable since it does not even need zero-knowledge proofs as seen in PICNIC. Where state management is not possible and the performance offered by stateless symmetric-crypto-based algorithms is not acceptable, the competing candidates from other PQC families should be considered. In this case, lattice-based algorithms seem to offer an interesting balance between security and performance. Some of the lattice-based underlying problems have gone through intense academic scrutiny for several years (in some cases, decades). One remark, however, is that the community may still benefit from a better understanding on how to choose secure parameters. From a

performance perspective, lattice-based algorithms offer decent performance. For example, CRYSTALS-DILITHIUM has both signature and public key size in the low single-digit kilobytes, and speed in the low single digit millions of cycles. Finally, multivariate-quadratic algorithms may have interesting performance advantages, such as the tiny signatures of 258 bits offered by GeMSS, but they are the candidates that would benefit the most from additional security assessments, given the recurrent attacks against MQ digital signature schemes throughout the history.

In summary, if state management is possible, IETF schemes (e.g., XMSS) seem to be a very promising approach. If state management is not possible, the candidates of the second round of the NIST competition (following the prioritization described above: 1—stateless symmetric-crypto-based, 2—lattice-based, 3—MQ) seem to be a promising approach.

COST OF CRYPTOGRAPHY VS. ADOPTION

Generally, technology adoption increases as performance increases and cost decreases. We expect postquantum cryptography to follow suit. NIST recently published “The Economic Impacts of the Advanced Encryption Standard, 1996-2017” which estimated a \$250 billion impact. A key factor in the success of AES has been the phenomenal performance achieved in modern microprocessors. Galois Counter Mode (GCM) is a popular AES mode of operation used in the networking space that secures the majority of internet traffic. Consider the historic cost



of the computation of GCM, and the adoption rate in the figure below, gathered from the ICSI Notary (<https://notary.icsi.berkeley.edu/>).

In the figure above, the “performance” of running the GCM algorithm—presented as “cost in cycles per byte” where lower cost-per-byte indicates *higher* performance—corresponds to various Intel processors that launched on those dates, with a trend of continuously improving GCM performance. Notably, in 2013 there was a dramatic increase in GCM performance (indicated by a significant drop in the cost of cycles/byte), which we believe contributed to a steep rise of GCM adoption across the industry.

INTEL’S DIRECTION ON PQC AND FIRMWARE

Intel’s strategy is to continue securing platforms using cryptographically strong digital signature standards that execute efficiently. Intel will continue to be aligned with standardization organizations and will evaluate algorithms based on security assurance, hardware cost, and performance—including that in embedded cores. Until full PQC transition has completed, a hybrid solution where two or more algorithms are executed in parallel—in order to remove dependence on a single algorithm or class of algorithms, seems an interesting approach. This should allow

for more flexibility and minimize redesign time of embedded microcontroller systems should a single (class of) algorithms become infeasible to deploy for their firmware authentication.

CONCLUSION

Firmware authenticity is a key factor in the proper function and security assurance of a platform built with embedded microcontroller cores. With the need for PQC-secure cryptographic algorithms to continue to assure this authenticity, Intel is investigating possibilities currently under consideration in various standardization processes. Our priority in this selection is and will always be security. Regarding performance, we are specifically focusing on the capabilities of platforms’ embedded cores to execute PQC algorithms and looking for algorithms that have parameters allowing flexibility in fitting their execution to the capabilities of both current and planned embedded cores.

REFERENCES

1. A. Huelsing, D. Butin, S. Gazdag, J. Rijneveld, and A. Mohaisen, (2018). XMSS: eXtended Merkle Signature Scheme - Request For Comment 8391 (RFC 8391). Internet Engineering Task Force (IETF), Retrieved: 24 June, 2019, <https://tools.ietf.org/html/rfc8391>

2. V. Lyubashevsky, L. Ducas, E. Kiltz, T. Lepoint, P. Schwabe, G. Seiler, and D. Stehle, (2017). CRYSTALS-DILITHIUM - A Submission to the NIST Post-Quantum Cryptography Standardization Project. National Institute of Standards and Technology (NIST). Retrieved: 24 June, 2019, <https://pq-crystals.org/>
3. T. Prest, P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, and Z. Zhang, (2017). Falcon - A Submission to the NIST Post-Quantum Cryptography Standardization Project. National Institute of Standards and Technology (NIST). Retrieved: 24 June, 2019, <https://falcon-sign.info/>
4. A. Casanova, J.-C. Faugere, G. Macario-Rat, J. Patarin, L. Perret, and J. Ryckeghem, (2017). GeMSS - A Submission to the NIST Post-Quantum Cryptography Standardization Project. National Institute of Standards and Technology (NIST). Retrieved: 24 June, 2019, <https://www.polsys.lip6.fr/Links/NIST/GeMSS.html>
5. L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, 1996, pp. 212–219.
6. W. Beullens, B. Preneel, A. Szepieniec, and F. Vercauteren, (2017). LUOV - A Submission to the NIST Post-Quantum Cryptography Standardization Project. National Institute of Standards and Technology (NIST). Retrieved: 24 June, 2019, <https://www.esat.kuleuven.be/cosic/pqcrypto/luov/>
7. S. Samardjiska, M.-S. Chen, A. Hülsing, J. Rijneveld, and P. Schwabe, (2017). MQDSS - A Submission to the NIST Post-Quantum Cryptography Standardization Project. National Institute of Standards and Technology (NIST). Retrieved: 24 June, 2019, <http://mqdss.org/>
8. G. Zaverucha, M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, and V. Kolesnikov, (2017). Picnic - A Submission to the NIST Post-Quantum Cryptography Standardization Project. National Institute of Standards and Technology (NIST). Retrieved: 24 June, 2019, <https://microsoft.github.io/Picnic/>
9. N. Bindel, S. Akleylek, E. Alkim, P. S. Barreto, J. Buchmann, E. Eaton, and G. Zanon, (2017). qTesla - A Submission to the NIST Post-Quantum Cryptography Standardization Project. National Institute of Standards and Technology (NIST). Retrieved: 24 June, 2019, <https://qtesla.org/>
10. J. Ding, M.-S. Chen, A. Petzoldt, D. Schmidt, and B.-Y. Yang, (2017). Rainbow - A Submission to the NIST Post-Quantum Cryptography Standardization Project. National Institute of Standards and Technology (NIST).
11. P. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Foundations Comput. Sci.*, 1994, pp. 124–134, Santa Fe: IEEE Comput. Soc. Press.
12. D. Bernstein, C. Dobraunig, M. Eichlseder, S. Fluhrer, S. Gazdag, A. Hülsing, and F. Mendel, (2017). SPHINCS+ - A Submission to NIST Post-Quantum Cryptography Standardization Project. National Institute of Standards and Technology (NIST). Retrieved: 24 June, 2019, <https://sphincs.org/>

Rafael Misoczki is a cryptographer/research scientist at Intel Labs. His work is focused on post-quantum cryptography and its application to secure update, root of trust, remote attestation, and other security flows. He has a PhD from the University of Paris (Pierre et Marie Curie), with a thesis on efficient constructions for post-quantum cryptography. He also holds an MSc in electrical engineering and a BSc in computer science from the University of Sao Paulo. Contact him at rafael.misoczki@intel.com.

Sean Gulley is a principal engineer at Intel's Data Center Group responsible for anticipating and accelerating new algorithmic intensive workloads. Since joining Intel in 2001, he has focused primarily on cryptography and compression HW and SW solutions for client and data center. He has a BS in computer engineering from Tufts University and an MS in electrical engineering from Stanford University. He has over 30 U.S. patents. Contact him at sean.gulley@intel.com.

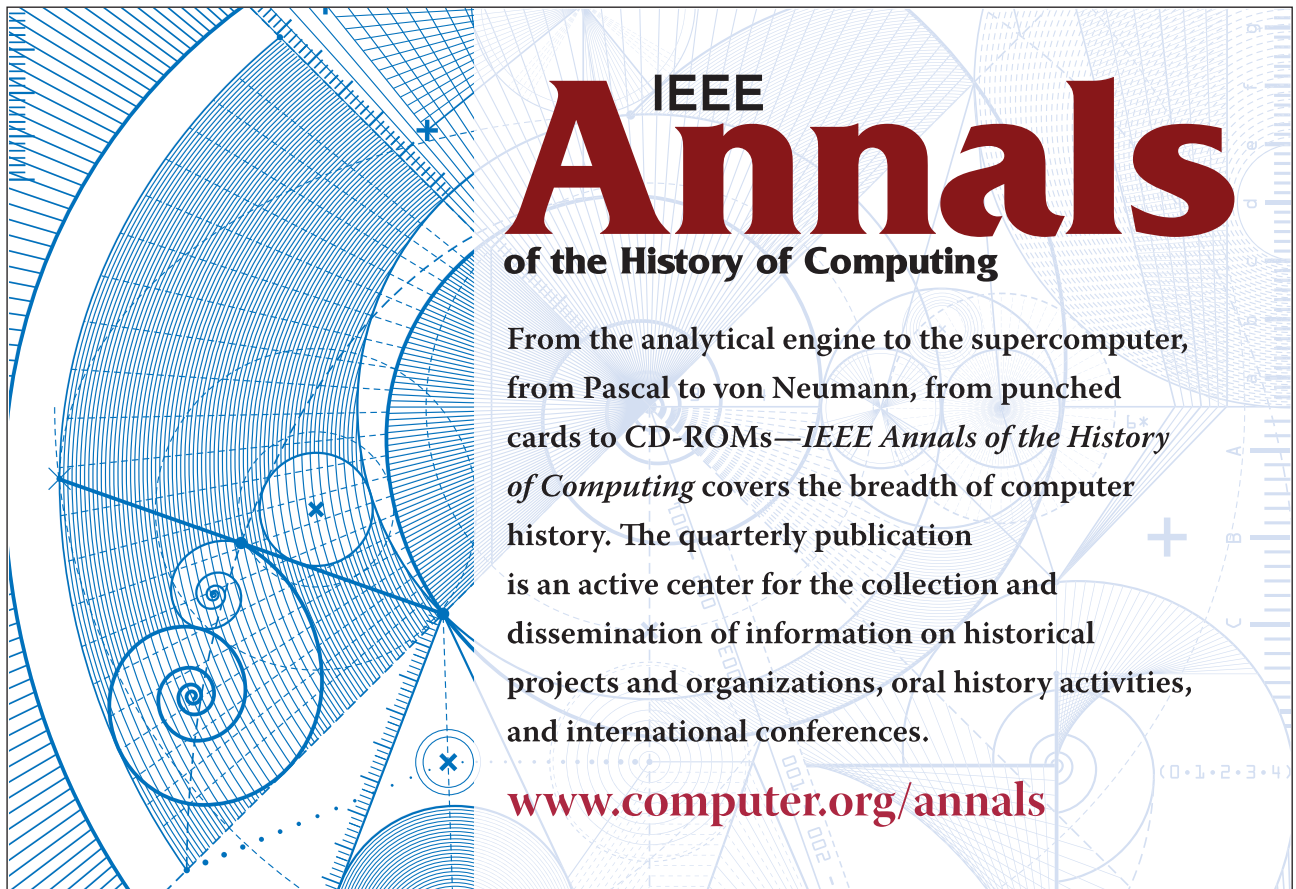
Vinodh Gopal is as senior principal engineer at Intel, working in the Data Center Group. His work includes accelerators, instruction-set extensions for x86 and architectural enhancements to processors, in applications such as cryptography, integrity, compression, and analytics over a range of products. In 2019, he won the Intel Inventor of the Year Award. Contact him at vinodh.gopal@intel.com.

Martin G. Dixon is an Intel Fellow in the Intel Product Assurance and Security (IPAS) group and director of architecture at Intel Corporation. He is responsible for guiding future research and architecture decisions to secure Intel's platforms. He has published a dozen academic papers in the field of computer architecture and holds 50 patents in the field of computer architecture and cryptography. He has a bachelor's degree in electrical and

computer engineering from Carnegie Mellon University. Contact him at martin.dixon@intel.com.

Hrvoje Vrsalovic has been involved in firmware and app-to-device interface software development in one form or another ever since being part of the original team that created Palm's WebOS in 2008. Since then, he has worked on software—and sometimes hardware—of many “smart” consumer products, particularly wearables. He recently joined Intel's IPAS group as a security architect. He has a BSc in computer science from UCSB and an MSc in electrical and computer engineering from Carnegie Mellon University. Contact him at harvey.vrsalovic@intel.com.

Wajdi K. Feghali is an Intel Fellow and the director of the Security and Algorithms Center of Innovation in the Data Center Group at Intel Corporation. He leads the development of cryptography, compression, data integrity and data de-duplication hardware and software solutions with a focus on efficient performance across Intel products. He has been granted more than 50 U.S. patents, with numerous other patents pending, and is the author of several published technical papers. He has a bachelor's degree in mathematics with a minor in computer science from the University of Ottawa. Contact him at wajdi.feghali@intel.com.

The advertisement features a background of intricate blue geometric patterns, including a large spiral on the left and various circular and linear designs on the right. The text is overlaid on this background. The title 'IEEE Annals of the History of Computing' is prominently displayed in the center. Below the title, a paragraph describes the publication's scope and purpose. At the bottom, the website URL is provided in a red font.

IEEE
Annals
of the History of Computing

From the analytical engine to the supercomputer, from Pascal to von Neumann, from punched cards to CD-ROMs—*IEEE Annals of the History of Computing* covers the breadth of computer history. The quarterly publication is an active center for the collection and dissemination of information on historical projects and organizations, oral history activities, and international conferences.

www.computer.org/annals