THE JBBA

PEER REVIEWED RESEARCH

OPEN ACCESS
ISSN Online: 2516-3957
ISSN Print: 2516-3949
https://doi.org/10.31585/jbba-2-1-(4)2019

Evaluation of Post-Quantum Distributed Ledger Cryptography

Robert E. Campbell Sr.
Capitol Technology University, USA

Correspondence: rc@medcybersecurity.com
Received: 08 January 2019 **Accepted**: 26 February 2019 **Published**: 16 March 2019

Abstract

This paper evaluates the current cybersecurity vulnerability of the prolific use of Elliptical Curve Digital Signature Algorithm (ECDSA) cryptography in use by the Bitcoin Core, Ethereum, Bitcoin Cash, and enterprise blockchains such as Multi-Chain and Hyperledger projects Fabric, and Sawtooth Lake. These blockchains are being used in media, health, finance, transportation and government with little understanding, acknowledgment of the risk and no known plans for mitigation and migration to safer public-key cryptography. The second aim is to evaluate ECDSA against the threat of Quantum Computing and propose the most practical National Institute of Standards and Technology (NIST) Post-Quantum Cryptography candidate algorithm lattice-based cryptography countermeasure that can be implemented near-term and provide a basis for a coordinated industry-wide lattice-based public-key implementation. Commercial quantum computing research and development is rapid and unpredictable, and it is difficult to predict the arrival of fault-tolerant quantum computing. The current state of covert and classified quantum computing research and advancement is unknown and therefore, it would be a significant risk to blockchain and Internet technologies to delay or wait for the publication of draft standards. Since there are many hurdles Post-Quantum Cryptography (PQC) must overcome for standardisation, coordinated large-scale testing and evaluation should commence promptly.

Keywords: *ECDSA, blockchain, post-quantum, lattice-based cryptography, cybersecurity, distributed ledger, qTESLA, Ring Learning with Errors, critical infrastructure*

JEL Classifications: *D02, D71, H11, P16, P48, P50*

Competing Interests:
None declared.

Ethical approval:
Not applicable.

Author's contribution:
RC designed and coordinated this research and prepared the manuscript in entirety.

Funding:
None declared.

Acknowledgements:
RC would like to acknowledge Dr. Ian McAndrew for his supervision and guidance in preparing this research.

1. Introduction

Rapid advances on a global scale in Quantum Computing technologies and the threat it poses to most standardized encryption prompted NIST to put out an international call for candidate quantum-resistant public-key cryptographic algorithms to evaluate for standardization. NIST will conduct efficiency analysis on their reference platform delineated in the *Call for Proposals*; NIST invites the public to perform similar tests and compare results on additional platforms (e.g., 8-bit processors, digital signal processors, dedicated complementary metal oxide semiconductor (CMOS), etc.) and provide comments regarding the efficiency of the submitted algorithms when implemented in hardware.

This research has two goals; the first is to examine the vulnerabilities in current Asymmetric Digital Signature Cryptography (ASDC) as used in private key generation in Bitcoin Blockchain technology in the PQC era. The second goal is to independently test and evaluate candidate NIST algorithms to assist in the process of selection of acceptable candidate cryptosystems for standardisation and the proposal of potential replacement of ADSC in private key generation in blockchain and distributed ledger technology. Most blockchain and distributed ledger technologies use an asymmetric digital signature scheme for private key generation such as, ECDSA, which has been cloned often from the Bitcoin Blockchain. These digital signature schemes are being implemented in critical sectors of government and the economy. Evaluations will include cryptographic strengths and weaknesses of NIST candidate pool of submitted algorithms. It is expected that the analysis will consist of required performance parameters that include;

Public Key, Ciphertext, and Signature Size, Computational Efficiency of Public and Private Key Operations, Computational Efficiency of Key Generation, and Decryption Failures against NIST provided Known Answer Test values (KAT).

Blockchain and Distributed Ledger cryptography private key generation cyber-security concepts are poorly understood, and often misrepresented. There is a misconception that Blockchain technology can't "be hacked," resulting in a general endorsement for critical sectors and industries [1]. The author believes that the technology offers excellent cyber-security promise for many areas, but the limitations and strengths must be defined. This work examines the weakness of the ECDSA and its current vulnerability and uses in the Bitcoin Blockchain or Distributed Ledger Technology (DLT). Many industries are rapidly adopting versions or mutations of the first of the Bitcoin Blockchain technology in essential sectors such as information technology, financial services, government facilities, healthcare, and Public Health Sector seemingly, without cybersecurity due diligence, a proper comprehension of the cryptography vulnerabilities or plans for addressing quantum computing threats [2]. The ECDSA is the foundation of Public Key Infrastructure (PKI) for many Internet applications and open source projects, and it's the primary source for public-key cryptography. The second part of this paper offers the most practical and near-term first-round candidate NIST Lattice-Based Post-Quantum Cryptography solution with a recommendation for immediate coordinated (academia, the private sector, government) independent testing, verification, and validation (IV&V) and test framework for sharing results [3]. This framework aids in speeding the approval of PQC standards that are vital to global cybersecurity. The scope of this work evaluates the lattice-based digital signature scheme qTESLA, based on the verifiable hardness of the decisional Ring Learning With Errors (R-LWE) [4]. Quantum computing's threat adversely affects the cybersecurity of financial services such as payment systems, general network communications systems, business functions including cloud computing, Internet of Things (IoT) and critical infrastructure. Further, the author believes that currently estimated timelines for the availability of large-scale fault-tolerant quantum computers are underestimated due to unpredicted global progress and the veil of secrecy surrounding classified research programs led by organizations and governments around the globe. It is, therefore, essential to begin work and testing the most likely candidate algorithms for normalization.

2. Implications in this work

Current encryption systems and standards such as Ron Rivest, Adi Shamir and Leonard Adleman (RSA), Digital Signature Algorithm (DSA), and ECDSA impact everything from defense, banking, healthcare, energy, telecommunications, intelligence, Internet and the Blockchain. The compromise, disruption or non-availability of one of these sectors would severely impact the health and safety of U.S. national security, public health, safety or its economy.

Blockchain technology is a revolutionary technology that has great potential in many applications. This technology has gained global interest in all industry sectors based on cryptography-based algorithms that are considered vulnerable today but will be increasingly threatened by accelerated advances in quantum computing.

3. Significance of the findings

The time to test and validate new post-quantum cryptology is now, given it takes at least ten years to build and deliver a new public key infrastructure. The pace at which quantum computing advancements can be anticipated is uncertain. The ability to transition to post-quantum cryptology appears to be very complicated, and there are many unknowns concerning establishing, standardizing and deploying post-quantum cryptography systems. All of this must be completed before the arrival of large-scale quantum computers because the cybersecurity of many vital services will be severely degraded.

4. Bitcoin and Distributed Ledger Technology

The Bitcoin Cryptocurrency (BTC) is the first widespread application of blockchain technology. The critical elements of Blockchain and DLT have been in existence for decades, and they include fault-tolerance, distributed computing, and cryptography. Succinctly, the first iteration of this technology is a decentralized distributed database that keeps records of transactions relatively secure and in an append-only mode, where all peers eventually come to a consensus regarding the state of a transaction. The Bitcoin Blockchain like others operates in an open peer-to-peer (P2P) network, where each node can function as a client and a server at the same time. The nodes in the system are connected over TCP/ IP and once a new node is connected that node broadcast peer IP addresses via Bitcoin address messages. Each address maps to a unique public and private key; these keys are used to exchange ownership of BTCs among addresses. A Bitcoin address is an identifier of 26 to 35 alphanumeric characters [5] . Since the advent of BTC along with its choice of a data structure, called a block, modified blockchain technologies, makes use of different data structures such as Directed Acyclic Graph (DAGs). Therefore, recent versions of the newest blockchains can longer accurately be called blockchains, and it is more appropriate to use the term Distributed Ledger (DL) that applies to all version of the blockchain. Presently, according to Crypto-Currency Market Capitalizations [6], there are more than 2000 alternate cryptocurrencies, and most make use of the Bitcoin Blockchain or are clones with minor differences in the private key generation cryptography and structure. The primary configuration changes include the underlying hash function, block generation times, data structures and method of distributed consensus. However; the critical task of generating private keys in blockchains remains unchanged across most blockchain adaptions, and this work asserts that the foundation of the current cryptocurrency markets and all the private and public sectors using this technology are vulnerable to the same cybersecurity weaknesses.

5. ECDSA, libsecp256k1 and OpenSSL

The ECDSA algorithm is part of public-key cryptography and is also the cryptography the Bitcoin blockchain uses to generate the public and private keys. The ECDSA is used in critical infrastructure, secure communications over the Internet, cellular and Wi-Fi and in many blockchain forks in use today. Specifically, the Bitcoin blockchain uses the ECDSA and the Koblitz curve *secp256k1* [7] which have significant weaknesses which include general algorithm structure, side-channel attacks, and threats from quantum computers. The Koblitz Curve was not adopted for standardisation by NIST due to the non-random structure of the algorithm. The Bitcoin creator selected a non-NIST P-256 approved curve to serve as a source of entropy. Entropy is defined in this case as the randomness inserted by an operating system or application for use in cryptography that requires random data. OpenSSL is an open-source software library used in BTC technology and ECDSA applications to secure communications and many critical infrastructures. OpenSSL [8] provides software Pseudo Random Number Generator (PNRG) based on a variety and type of hardware and software sources. Its core library is written in the C programming language. The process starts once the Bitcoin Core client is installed, and the user receives a set of ECDSA key pairs, called Addresses. The PRNG starts in the state unseeded and this state; it has zero entropy. A call to RAND bytes is made, and it will transfer automatically into the state seeded with a presumed entropy of 256 bits and is feed to the PRNG through a call to RAND add. The keys generated from this process are necessary to transfer BTC from one Address to the other. Next, the client needs to sign a specific message (called Transaction) with the private key of the user. The public key is used to check if the given user has rights to BTC [9].

The ECDSA algorithm relies on generating a random private key used for signing messages and a corresponding public key used for checking the signature. The bit security of this algorithm depends on the ability to compute a point multiplication and the inability to calculate the multiplicand given the original and product points.

The Koblitz curve secp256k1 is non-verifiably random and is defined by Standards for Efficient Cryptography Group (SECG), instead of the NIST 186-3 DSS Standard using the elliptic curve secp256r1. The security of the ECDSA algorithm and protocols relies on a source of distributed random bits.

6. Fault Attack on Bitcoin's Elliptic Curve with Montgomery Ladder Implementation.

This Montgomery Ladder Fault Attack method is a fault attack on elliptic curve scalar product algorithms and can be used when the (y-coordinate) is not used. The bit security of the elliptic curve parameters in most cases can be significantly reduced. The Fault attack is a robust side-channel technique that is used to break ECDSA cryptographic schemes. The idea is to inject a fault during the computations of implementation and to use the faulty outputs to deduce information on the secret key stored in the secure component [10]. Table 1 gives the resultant bit security after the Montgomery Ladder Fault Attack.

Table 1: Curve parameter security according to Montgomery Ladder Fault Attack [10]

Values *secp*	P1363 IPSEC	X9.62 X9.63	NIST	Strength	Security
256k1	c/c	c/r		128	**50**
256r1	c/c	r/r	r	128	121

The bold font indicates the scep256k1 security is below 2^{60} since these computations can be easily performed with classical computers. The mention 'r' denotes parameters explicitly recommended in the standard, while the mention 'c' denotes parameters in conformance with the standard. The column "Strength" refers to the standard. Clearly, implementations without protections, the attacker can compute the discrete logarithm in the twist with a cost of 2^{50} operations and retrieve the secret scalar for n = 256.

7. Algorithm Security Strength

Breaking a cryptographic algorithm can be defined as defeating some aspect of the protection that the algorithm is intended to provide. For example, a block cipher encryption algorithm that is used to protect the confidentiality of data is broken if, with an acceptable amount of work, it is possible to determine the value of its key or to recover the plaintext from the ciphertext without knowledge of the key.

The approved security strengths for federal applications are 128, 192 and 256 bits. Note that a security strength of fewer than 128 bits is no longer approved because quantum algorithms reduce the bit security to 64 bits. NIST Special Publication 800-57 Part 1 Revision 4: Recommended for Key Management as shown in Table 2 [11]. The Fault Attack on Bitcoin's Elliptic Curve with Montgomery Ladder Implementation yields security strength of only 50 bits as shown in Table 1.

8. NIST and Post-Quantum Cryptography

In December 2016, NIST formally announced its Call for Proposals (Request for Nominations for Public-Key Post-Quantum Cryptographic Algorithms), [12]. This call solicited

Table 2: Comparison of conventional and quantum security levels of typical ciphers [12].

Algorithm	Key Length	Effective Key Strength / Security Level	
		Conventional Computing	Quantum Computing
RSA-1024	1024 bits	80 bits	0 bits
RSA-2048	2048 bits	112 bits	0 bits
ECC-256	256 bits	128 bits	0 bits
ECC-384	384 bits	256 bits	0 bits
AES-128	128 bits	128 bits	64 bits
AES-256	256 bits	256 bits	128 bits

proposals for post-quantum digital signature as well as public-key encryption and Key Encapsulation Mechanism (KEM)/Encryption for evaluation. In response, there were 82 total submissions, and 69 were accepted, and five withdrew. The results and categories included 19 Signatures and 45 KEM Encryption. The Signature category which produces private keys included five Lattice-based submissions, and this work focuses on qTESLA's submission which is based on the verifiable hardness of the decisional Ring Learning With Errors (R-LWE) problem [4]. Public Key Systems based on R-LWE is computationally superior over LWE systems because of reduced overhead, greater capacity for message space and smaller public key sizes.

9. Selected algorithm for test and evaluation: qTESLA

The author's considerations for the selection qTESLA, are "reasonable" key and ciphertext sizes, and to a lesser extent the number of CPU cycles required for encryption, decryption, and verification, and potential incorporation into constrained devices such as smartphones and emerging IoT devices. Additional considerations included trust, metrics, parameters, migration, compatibility, and efficient and secure implementation. This submission utilizes two approaches for parameter generation. The first approach is called "heuristic qTESLA," and it uses heuristic method parameter generation and the second approach is called "provably-secure qTESLA,"

Table 3: Adapted from The NIST Post-Quantum Crypto "Competition" [13].

Level	Security Description
I	At least as hard to break as AES128 (exhaustive key search)
II	At least as hard to break as SHA256 (collision search)
III	At least as hard to break as AES192 (exhaustive key search)
IV	At least as hard to break as SHA384 (collision search)
V	At least as hard to break as AES256 (exhaustive key search)

and its parameter generation is provably-secure. qTESLA includes five parameter sets that correspond to two security levels located in Table 3.

Security levels:

A. Heuristic qTESLA:
- qTESLA-I: NIST's security category 1.
- qTESLA-III-speed: NIST's security level 3 (option for speed).
- qTESLA-III-size: NIST's security level 3 (option for size).

B. Provably-secure qTESLA:
- qTESLA-p-I: NIST's security category 1.
- qTESLA-p-III: NIST's security category 3 [4].

The security of lattice-based systems is provably secure under worst-case hardness assumptions. In the author's view, it is not

Table 4: Description and bounds of all the system parameters [4]

Parameter	Description	Requirement
λ	security parameter	-
q_h, q_s	number of hash and sign queries	-
n	dimension (n – 1 is the poly. degree)	power of two
σ, ξ	standard deviation of centered discrete Gaussian distribution	$\sigma = \xi/\sqrt{2\ln 2}$
k	#R-LWE samples	-
q	modulus	$q = 1 \mod 2n$, $q > 4B$ For provably secure parameters $q^{nk} \geq \lvert\Delta S\rvert \cdot \lvert\Delta L\rvert \cdot \lvert\Delta H\rvert$ $q^{nk} \geq 2^{4\lambda+nkd}\, 4q_s^3\, (q_s + q_h)^2$
h	# of nonzero entries of output elements of Enc	$2^h \cdot \binom{n}{h} \geq 2^{2\lambda}$
L_E, η_E L_S, η_S	bound in checkE bound in checkS	$\eta_E \cdot h \cdot \sigma$ $\eta_S \cdot h \cdot \sigma$
B	interval of randomness is chosen during signing	$B \geq \frac{k \cdot n\sqrt{M} + 2L_{S-1}}{2\left(-1-\sqrt[kn]{M}\right)}$, near a power of two
d	number of rounded bits	$\left(i - \frac{2L_E+1}{8^d}\right)^{kn} \geq 0.3$, $d > \log_2(B)$
b_{GenA}	number of blocks requested to SHAKE128 for GenA	$b_{GenA}\, \varepsilon\mathbb{Z} > 0$
$\lvert\Delta H\rvert$ $\lvert\Delta S\rvert$ $\lvert\Delta L\rvert$		$\sum_{j=0}^{h} \sum_{i=0}^{h-j} \frac{\binom{k}{2}\binom{n}{i}^{2^{2i}} (k_j^n - 2i)^{2j}}{(4(-B-L_s)+1)^n (2^d+1)^{nk}}$
δz δw δ_{keygen}	acceptance probability of z acceptance probability of w acceptance probability of key pairs	experimentally experimentally experimentally
sig size pk size sk size	theoretical size of signature theoretical size of public key theoretical size of secret key	experimentally experimentally experimentally
$\varkappa$	output length of hash function H and input length of GenA, PRF_1, PRF_2, Enc and ySampler	$\varkappa \geq \lambda$

Table 5: Parameters for each of the proposed heuristic and provably-secure parameter sets with $q_b = 2^{128}$ and $q_s = 2^{64}$; $M = 0.3$ [4]

Parameter	qTESLA-I	qTESLA-III-speed	qTESLA-III-size	qTESLA-p-I	qTESLA-p-III
λ	95	160	160	95	160
$\varkappa$	256	256	256	256	256
n	512	1024	1024	1024	1024
σ, ξ	23.78, 27.9988	10.2, 12	8.49, 9.9962	8.5, 10	8.5, 10
k	1	1	1	4	5
q	4205569 $\approx 2^{22}$	8404993 $\approx 2^{22}$	4206593 $\approx 2^{22}$	485978113 $\approx 2^{29}$	1129725953 $\approx 2^{30}$
h	30	48	48	25	40
L_E, η_E	1586, 2.223	1147, 2.34	910, 2.23	554, 2.61	901, 2.65
L_S, η_S	1586, 2.223	1233, 2.52	910, 2.23	554, 2.61	901, 2.65
B	2^{20} -1	$2^{21} - 1$	2^{20} - 1	$2^{21} - 1$	2^{23} - 1
d	21	22	21	22	24
b_{GenA}	19	38	38	108	180
|ΔH|				$\approx 2^{435.8}$	$\approx 2^{750.9}$
|ΔS|				$\approx 2^{23551.6}$	$\approx 2^{51199.7}$
|ΔL|				$\approx 2^{94208.0}$	$\approx 2^{2560000}$
δ_w	0.31	0.38	0.25	0.33	0.34
δz	0.44	0.56	0.37	0.78	0.81
δ_{sign}	0.14	0.21	0.09	0.26	0.28
δ_{keygen}	0.45	0.60	0.39	0.59	0.44
sig size	1376	2848	2720	2848	6176
pk size	1504	3104	2976	14880	39712
sk size	1216	2112	2112	4576	12320
classical bit hardness	104	178	188	132	247
quantum bit hardness	97	164	169	123	270

likely that current PQC will be direct replacements for current standards and will likely impact the entire category of Internet protocols, such as Transport Layer Security (TLS) and Internet Key Exchange (IKE).

System parameters can be viewed in Table 4 and Table 5.

10. Informal Signature Scheme

Informal descriptions of the algorithms that give rise to the signature scheme qTESLA are shown in Algorithms 1, 2 and 3. These algorithms require two basic terms, namely, B-short and well-rounded, which are defined below. Let q, L_E, L_S, and d be system parameters that denote the modulus, the bound constant for error polynomials, the bound constant for the secret polynomial, and the rounding value, respectively. An integer polynomial y is B-short if each coefficient is at most B in absolute value. An integer polynomial is w well-rounded if w is $([q/2] - L_E)$-short and $[w]_L$ is $(2^{d-1} - L_E)$-short, where $[w]_L$ denotes the unique integer in $(-2^{d-1}, 2^{d-1}] \subset Z$ such that $w = [w]_L$ modulo 2^d. Also, $[w]_M$ is the value represented by all but the d least significant bits of $(w - [w]_L)$. Let $R = Z[x]/(x^n + 1)$ and $R_q = Z_q[x]/(x^n + 1)$. The hash oracle H() maps from {0, 1}* to H, where H denotes the set of polynomials $c \in R$ with coefficients in {-1, 0, 1} with exactly h nonzero entries.

Algorithm 1: Informal description of the key generation.

Require - , n/a

Ensure: Secret key $sk = (s; e_1, \ldots, e_k, a_1, \ldots, a_k)$, and public key $pk = (a_1, \ldots, a_k, t_1, \ldots t_k)$

1. $a_1, \ldots, a_k \leftarrow R_q$ invertible ring elements.
2. Choose $s \in R$ with entries from D_σ. Repeat step if the h largest entries of s sum to L_S.
3. For $i = 1, \ldots, k$: Choose $e_i \in R$ with entries from D_σ. Repeat step at iteration i if the h
4. largest entries of e_i sum to L_E.
5. For i = 1, …, k: Compute $t_i \quad a_i s + e_i \in R_q$.
6. Return $sk = (s; e_1, \ldots e_k; a_1, \ldots a_k)$ and $pk = (a_1, \ldots a_k, t_1, \ldots, t_k)$.

Algorithm 2: Informal description of the signature generation.

Require: Message m, secret key $sk = (s; e_1, \ldots e_k, a_1, \ldots a_k)$

Ensure: Signature (z; c)

1. Choose y uniformly at random among B-short polynomials in R_q.
2. $c \leftarrow H([a_1 y]_M, ..., [a_k y]_M, m)$.
3. Compute $z \leftarrow y + sc$.
4. If z is not $(B - L_S)$-short then retry at step 1.
5. For $i = 1, ..., k$: If $a_i y - e_i c$ is not well-rounded then retry at step 1.
6. Return (z, c).

Algorithm 3: Informal description of the signature verification.

Require: Message m, public key $pk = (a_1, \ldots, a_k, t_1, \ldots, t_k)$, and signature (z, c)

Ensure: "Accept" or "reject" signature

1. If z is not (B - L_S)-short then return reject.
2. For $i = 1, \ldots, k$: Compute $w_i \leftarrow a_i z - t_i c \in R_q$.
3. If $c \neq H([w_1]_M, \ldots, [w_k]_M, m)$ then return reject.
4. Return accept [4].

Performance of post-quantum qTESLA algorithms analysis

To evaluate the performance of the provided implementations written in portable C, the author ran benchmarking suite on three machines powered by: (i) an Intel® Core™ i7-6500 CPU @ 2.50 GHz x 4 (Skylake) processor (see table 4**)** (ii) an Intel® Core™ i5-6400T CPU @ 2.20GHz (VMWARE)(Haswell) processor (see table 5) (iii) an Intel® Core™ i7-2630QM CPU @ 2.00GHz × 8 (Haswell) (see table 6) all running Ubuntu 18.04.1 LTS. For compilation, GCC version 7.3.0 was used in all test.

11. Analysis

The author argued that the uncertainties had not been appropriately addressed. For example, there is the possibility that additional quantum algorithms or techniques will be developed, which will lead to new and unanticipated attacks. Also, it is difficult to calculate the impact of those programs that are highly classified, and its performance characteristic is not public. Rapid and unpredictable advancements in quantum computing, are endangering or making current encryption schemes obsolete. It has been established that the most significant threat posed by quantum computers is directed towards current RSA, ECC digital signature scheme systems on which Bitcoin, Distributed Ledger and much of Internet-based technology uses.

It has been settled that the current RSA and ECC based public key cryptography are broken, and the AES cryptography is adversely reduced in bit security by quantum computing era. It is the author's view that recommendations such as doubling the AES key size need to be examined while considering the constraints of present systems. Current AES-128 is reduced to 64-bit security, and AES-256 would have 128-bit security.

An example of the impact of doubling the key size for AES-256 to AES-512 is not well documented and verified. This alternative algorithm (AES-512) would most likely use input block size and a key size of 512-bits. An increasing number of rounds and key schedule would adversely impact performance constraints, especially for constrained devices. The higher the key size, the more secure the ciphered data, but also the more rounds needed. In the hardware perspective, a bigger key size also means a larger area and power consumption due to more operations that need to be done. More focus and examination need to be done for AES in the PQC era, especially for constrained devices.

The author specifically, examined the ECDSA that are in use in Bitcoin and Distributed Ledger technologies. Secondly, evaluated NIST Candidate PQC for standardisation and possible replacement in blockchain and other public key cryptography Internet-based technologies. Table 6 gives the ECDSA (P-256) parameters used as the benchmark for comparison regarding the number of quantum security bits, and the size of the public key, secret key and signature key as an independently controlled variable. According to NIST, the use of schemes with less than 112-bit security is deprecated and will eventually be disallowed for use by U.S. government institutions to handle sensitive data. It is noted that that speed at which the encryption and decryption occurs is also an important parameter.

Table 6: ECDSA; signature and key sizes are given in bytes [4].

Software/ Scheme	Computation Assumption	Bit Security	Key Size (bytes)	Signature Size (bytes)
ECDSA (P-256)	Elliptic Curve Discrete Logarithm	128	pk: 64 sk: 96	64

Table 7: Intel® Core™ i7-6500 (Skylake) CPU @ 2.50 GHz x 4

Scheme	Keygen	Sign	Verify	Total (sign + verify) median
qTESLA-I	1321.3	402.4	82.6	485
qTESLA-III-speed	2987.6	551	168.8	719.8
qTESLA-III-size	5042.8	1035.8	170.4	1206.2
qTESLA-p-I	5370.1	1033.2	423.4	1456.6
qTESLA-p-III	25791.8	4223.2	2134	6357.2
Scheme	**Keygen**	**Sign**	**Verify**	**Total (sign + verify) average**
qTESLA-I	1501.7	557.3	87.1	644.4
qTESLA-III-speed	3349.9	747.2	172.9	920.1
qTESLA-III-size	5329.7	1448.6	171.8	1620.4
qTESLA-p-I	5545.3	1328.9	428	1756.9
qTESLA-p-III	27570.3	5254.8	2156.4	7411.2

Table 8: Intel® Core™ i5-6400T CPU @ 2.20GHz (VMWARE)

Scheme	Keygen	Sign	Verify	Total (sign + verify) median
qTESLA-I	1460	461	88.7	550.0
qTESLA-III-speed	3217	634.8	180.8	815.7
qTESLA-III-size	5367	1219.7	181.7	1401.4
qTESLA-p-I	6316	1187.2	446.5	1633.7
qTESLA-p-III	29961	4730.5	2260	6990.6
Scheme	**Keygen**	**Sign**	**Verify**	**Total (sign + verify) average**
qTESLA-I	1786	664	107	772
qTESLA-III-speed	3998	898	212	1110
qTESLA-III-size	618	1718	206	1925
qTESLA-p-I	6898	1595	520	2116
qTESLA-p-III	31280	5952	2412	8364

Table 9: Intel® Core™ i7-2630QM CPU @ 2.00GHz × 8

Scheme	Keygen	Sign	Verify	Total (sign + verify) median
qTESLA-I	1729.3	494	105.7	599.7
qTESLA-III-speed	3900.5	708.6	223.2	931.8
qTESLA-III-size	6047	1350.2	220.5	1570.7
qTESLA-p-I	6987.2	1328.2	563.8	1892
qTESLA-p-III	36254.2	5204.5	2858	8062.5
Scheme	**Keygen**	**Sign**	**Verify**	**Total (sign + verify) average**
qTESLA-I	1972	672	108	780
qTESLA-III-speed	4367.9	929	224.4	1153.4
qTESLA-III-size	6994.3	1858.8	225.2	2084
qTESLA-p-I	7343	1683	5689	2252
qTESLA-p-III	3739	6430	2882	9312

The following results cannot be compared directly with the vendor qTESLA's submitted results, but; specific observations can be made with alternative applications and platforms. It is the author's view that if the key sizes are not manageable and practical for use in conventional and constrained devices, then the time or speed becomes less critical metric compared to key size.

Table 7, Table 8 and Table 9 gives the results of the independent tests on respective platforms and performance is measured (in thousands of cycles) of the reference implementation. Results for the median and average (in the first and second table respectively) are rounded to the nearest 10^3 cycles. Signing is performed on a message of 59 bytes.

12. Recommendations

The PQC Standardisation process is complex, arduous and requires coordinated involvement (academia, private and public sector) and requires significant IV&V before formalization. Successful PQC must be resistant to both classical and quantum attacks. Multiple tradeoffs will have to be considered such as security, performance, key size, signature size, and side-channel resistance countermeasures. Other important considerations are the capability to migrate into new and existing applications such as TLS, IKE, code signing, PKI infrastructure.

It is necessary to begin a coordinated international campaign to mitigate the uncertainties of breakthroughs and the unknowns regarding classified programs. The aim should include, information sharing between the academic, public and private sector toward the common goal.

It is critical to devise and initiate the incorporation of cutting edge yet practical PQC to prevent a disastrous impact on global privacy, security, and economy before the arrival of large-scale fault-tolerant quantum computing.

13. Conclusion

qTESLA's submission for NIST Security Categories I and III as tested on platforms described in this work are more than two orders of magnitude larger for the public-key for qTESLA-p-1 (128-bit security) and qTESLA-p-III (192-bit security). The qTESLA-p-1 secret key is 56 times the size of ECDSA's secret key and qTESLA-p-III is two orders of magnitude larger.

It is essential to come to a consensus on how to assess quantum security. Currently, there is not a clear agreement on the best way to measure quantum attacks. It is, nevertheless, fundamental that work continues with alternatives that will produce smaller key sizes, comparable to the current ECDSA algorithms. The major drawback with qTESLA is the large key sizes which make it unlikely to be accepted in its current configuration. However, there is ongoing research being done to make it potentially a more viable candidate, both by reducing the key sizes and providing more efficient implementations (see tables 7, 8, 10).

The qTESLA's "Heuristic" submission for NIST Security Categories I and III are qTESLA-I, qTESLA-III-space, and qTESLA-III-size. The vendor claims that their heuristic approach is the security level of an instantiation of a scheme by the hardness level of the instance of the underlying lattice problem. Also, the claim is that it corresponds to these parameters regardless of the tightness gap of the provided security reduction if the corresponding R-LWE instance is intractable.

These claims and the necessary proof are beyond the scope of this work and cannot be independently verified and validated and is not the author's aim. It is important to note that; the results of qTESLA's heuristic algorithm were captured and are analyzed against its provably secure submissions. The heuristic algorithms were tested on the same platforms identified in the provably secure submission. qTESLA-I's public-key size vs. qTESLA-p-1's public-key size is a reduction of 90%. The secret key size at the same bit security level is reduced by 60%, and the signature size is reduced by 52%. Observations for public keys; qTESLA-III-size vs. qTESLA-p-III is reduced by 92%; secret key size reduction is 66%; signature size reduction is 56% (see Table 10).

The difference in the heuristic key sizes are dramatically reduced and compares more favorably to ECDSA (P-256) parameters. While the heuristic values are dramatically reduced compared to the provably secure values, the key sizes are still large compared to current standard ECDSA (P-256) sizes. For

Table 10: qTESLA Public-Key, Secret key, and Signature Size

Scheme (Bytes)	Public-key	Secret key	Signature Size
qTESLA-I	1504	2112	1376
qTESLA-III-speed	3104	4160	2848
qTESLA-III-size	2976	4160	2720
qTESLA-p-I	14880	5184	2848
qTESLA-p-III	39712	12352	6176

example; the best result for the secret key size for qTESLA-III-size (4160) vs. ECDSA (P-256) secret key size (96) is a 4233% increase and would prove problematic in existing systems.

14. Future Work

The author selected qTESLA's submission which is 1 of 5 NIST Candidate PQC digital signature schemes. Additional work needs to be done in verifying and validating and testing vendors results. Concrete PQC parameters for testing and validation need to be created for the promotion of a baseline. The parameters should be modified to determine the best tradeoffs while maintaining required security. Moreover, the organization of guidelines and standards are necessary for the wider cryptography community to aid in PQC standardisation create efficient, high-quality implementations.

Continued measurements of current PQC scheme implementations should be performed, such as performance and memory usage on the ARM and CMOS platforms. Many embedded devices have ARM and CMOS architecture and have limited computational and memory resources. NIST currently plans a Post-Quantum Cryptography Round 2 call tentatively schedule in 2019 and will offer additional opportunities for IV&V and research.

References:

[1] S. . M, A. H. D, M. . M, P. . P and S. . Balaji, "Decentralized digital voting application," *International Journal of Advance Research, Ideas and Innovations in Technology,* vol. 4, no. 3, pp. 1725-1728, 2018

[2] E. . Feig, "A Framework for Blockchain-Based Applications.," , 2018. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1803.html. [Accessed 7 1 2019]

[3] D. Moody, L. Feldman and G. Witte, "Securing Tomorrow's Information Through Post-Quantum Cryptography", *Csrc.nist.gov*, 2019. [Online]. Available: https://csrc.nist.gov/publications/detail/itl-bulletin/2018/02/securing-information-through-post-quantum-cryptography/final. [Accessed 7 1 2019].

[4] E. Alkim, N. Bindel, J. Buchmann, Ö. Dagdelen, E. Eaton, G. Gutoski, J. Krämer, and F. Pawlega, "Revisiting TESLA in the Quantum Random Oracle Model," *Post-Quantum Cryptography Lecture Notes in Computer Science*, pp. 143–162, 2017. [Accessed 7 1 2019].

[5] G. O. Karame, "On the Security and Scalability of Bitcoin's Blockchain," , 2016. [Online]. Available: https://dl.acm.org/citation.cfm?id=2976756. [Accessed 7 1 2019].

[6] "Cryptocurrency Market Capitalizations," , . [Online]. Available: https://coinmarketcap.com/currencies/bitcoin-cash/. [Accessed 8 1 2019].

[7] N. T. Courtois, G. . Song and R. . Castellucci, "Speed Optimizations in Bitcoin Key Recovery Attacks," *Tatra mountains mathematical publications,* vol. 67, no. 1, p. 103, 2016.

[8] J. Ooms, "Toolkit for Encryption, Signatures and Certificates Based onOpenSSL," , 2016. [Online]. Available: https://cran.r-project.org/web/packages/openssl/index.html. [Accessed 7 1 2019].

[9] J. A. Dev, "Bitcoin mining acceleration and performance quantification," , 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6900989. [Accessed 30 12 2018]

[10] P.-A. R. L. D. R. F. V. Fouque, "Fault Attack on Elliptic Curve with Montgomery Ladder Implementation," 2008

[11] "NIST Special Publications - NIST Computer Security ...," , . [Online]. Available: http://csrc.nist.gov/publications/PubsSPs.html. [Accessed 7 1 2019].

[12] L. .Chen, S. P. Jordan, Y.-K. . Liu, D. . Moody, R. C. Peralta, R. A. Perlner and D. C. Smith-Tone, "Report on Post-Quantum Cryptography | NIST," , 2016. [Online]. Available: https://nist.gov/publications/report-post-quantum-cryptography. [Accessed 30 12 2018].

[13] D. Moody, "The NIST Post-Quantum Crypto "Competition" "The Ship Has Sailed"," in *Asiacrypt 2017*, Hong Kong, 2017.

