

QSOR: Quantum-safe Onion Routing

Zsolt Tujner^{1,2}^a, Thomas Rooijakkers¹^b, Maran van Heesch¹^c and Melek Önen²^d

¹TNO, The Hague, the Netherlands

²EURECOM, Sophia-Antipolis, France

Keywords: Tor, Anonymous Routing, Post-quantum Cryptography.

Abstract: We propose a study on the use of post-quantum cryptographic primitives for the Tor network in order to make it safe in a quantum world. With this aim, the underlying keying material has first been analysed. We observe that breaking the security of the algorithms/protocols that use long- and medium-term keys (usually RSA keys) have the highest impact in security. Therefore, we investigate the cost of quantum-safe variants. Six different post-quantum cryptographic algorithms that ensure level 1 NIST security are evaluated. We further target the Tor circuit creation operation and evaluate the overhead of the post-quantum variant. This comparative study is performed through a reference implementation based on SweetOnions that simulates Tor with slight simplifications. We show that a quantum-safe Tor circuit creation is possible and suggest two versions - one that can be used in a purely quantum-safe setting, and one that can be used in a hybrid setting.

1 INTRODUCTION

Nowadays, information available online is expanding in an unforeseen way, a vast amount of data is uploaded and shared through social media, IoT, etc. This data attracts unwanted attention and might paint a bad image of some stakeholders. Consider the case of Edward Snowden who put the National Security Agency (NSA) in the spotlight by shedding light on how the American population was wiretapped¹. When blowing the whistle on such a large scale one would aim to remain anonymous, as this act can negatively affect the career and freedom of the individual. In oppressive regimes, where the freedom of speech is abused, this is even more serious, as any type of negative speech or expressing freedom of information may be recognized as an act of treason resulting in severe punishments.

The Onion Router (Tor) (Dingledine et al., 2004)) aims to ensure the anonymity of its users when accessing or communicating over the Internet. When using Tor, messages or website connection requests

are sent through a network of relays and after multiple 'hops' reach their destination. The cryptographic schemes used today in Tor are based on hard mathematical assumptions e.g., integer factorization (Katz and Lindell, 2007). These schemes are assumed to be secure against classical adversaries, as solving them with the currently known algorithms cost exponential time. However, with a quantum computer solving these problems becomes feasible.

Contributions. We investigate the main challenges to build and maintain a quantum-safe Tor network. We first examine the keying material used in Tor and identify the impact of the compromise of each of them. We observe that the migration towards quantum-safe Tor should start with the update of cryptographic algorithms that involve long-term and medium-term keys. Such a migration naturally results in additional cost in terms of CPU and bandwidth. To evaluate the overhead resulting from the shift to post-quantum (PQ) cryptographic algorithms, we have conducted an experimental study while considering six PQ public-key encryption algorithms that are part of NIST's round 2 submissions². We observe that each scheme comes with different advantages and limitations, and that consequently there is no ideal solution that offers optimal CPU and bandwidth overhead. We further focus on a particular Tor network operation, namely cir-

^a <https://orcid.org/0000-0001-9748-0044>

^b <https://orcid.org/0000-0001-6930-0421>

^c <https://orcid.org/0000-0003-0135-3521>

^d <https://orcid.org/0000-0003-0269-9495>

¹<https://www.theguardian.com/world/interactive/2013/nov/01/snowden-nsa-files-surveillance-revelations-decoded#section/1>, Accessed on 2019-06-14

²<https://csrc.nist.gov/Projects/post-quantum-cryptography/round-2-submissions>

circuit building, and evaluate the cost of its PQ variant. This study is performed through a reference implementation based on SweetOnions³ that simulates Tor with slight modifications. We show that while the increase in CPU time is acceptable and similar among different implementations, the bandwidth overhead remains significant. For more details, the reader can refer to the full version of this paper⁴.

2 BACKGROUND

The Onion Router (Tor). Tor (Dingledine et al., 2004) is one of the most popular tools to achieve anonymity for web browsing. When a Tor user accesses a website, the encrypted traffic is routed across multiple relays. The use of multiple nodes enroute to the destination helps obfuscate the connection of users and hence achieve anonymity: Each node in the path towards the destination (named a circuit), only has information about the previous node and the next node. Messages are encrypted by the source in a layered fashion and each encryption layer is removed by one relay node. The default number of relay nodes to set up a circuit is three (entry node, middle node, exit node)⁵. Each node has to communicate information called descriptors to Directory Authorities who maintain a state of the network. The Directory Authorities vote on the network status to obtain a consensus document. The user connects to one of the Directory Authorities, fetches the consensus document and the Tor software selects a path from the available nodes. The overall Tor framework is illustrated in Figure 1. Each Tor node receives and maintains multiple cryptographic keys for different purposes. Table 1

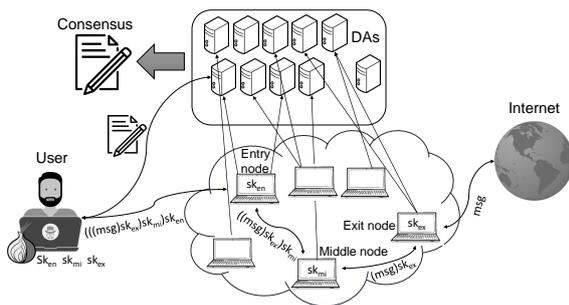


Figure 1: An overview of Tor with nine directory authorities (DAs), a bridge authority, the consensus document, Tor nodes, the symmetric keys (sk), and the message (msg).

³<https://github.com/LeonHeTheFirst/SweetOnions>, accessed on 28/11/2019.

⁴<https://arxiv.org/abs/2001.03418>

⁵<https://trac.torproject.org/projects/tor/wiki/TorRelayGuide>, accessed on 28/11/2019.

Table 1: Functionalities of keys in Tor.

Type	Key lifetime	Key name	Function
RSA	long-term	identity key	Establish relay identity
	medium-term	onion key	Decrypt cells at circuit creation. Used in <code>ntor</code> and TAP for handshakes.
	short-term	connection key	Establish TLS channels.
Curve25519	medium-term	handshake key	Handle handshakes <code>ntor</code> .
Ed25519	long-term	master identity key	Sign medium-term Ed25519 key. Never changes.
	medium-term	signing key	Replaces RSA id key, signs documents, certificates.
	short-term	link authentication key	Authenticate handshakes after circuit negotiation.

provides an overview of the asymmetric keys used in Tor with their lifetime and functionalities. Long-term keys are used at least for one year, medium-term keys are used for three to twelve months, and short-term keys have a lifespan of minutes to a maximum of one day.

Post-quantum Cryptography. The security of the current asymmetric encryption and digital signature standards mostly depend on the hardness of integer factorization (RSA) or discrete logarithm (Diffie-Hellman, Elliptic Curve Discrete Logarithm) (Katz and Lindell, 2007). As described in (Shor, 1995), such cryptographic schemes can be easily broken in polynomial time when using quantum computers. Hence, researchers are actively developing post-quantum cryptographic solutions to resist quantum attacks (Bernstein, 2009). In 2016, the National Institute of Standards and Technology (NIST) opened a call for proposals on the topic of quantum-safe cryptographic solutions for new quantum-safe standards (NIST, 2016). On January 30, 2019 the candidates for the second round were 17 asymmetric key encryption and key-establishment algorithms and 9 digital signature algorithms. The transition to quantum-safe cryptographic schemes is expected to be a lengthy process. The adoption of quantum-safe schemes results in a significant increase in bandwidth and computational cost. Developers adopt the hybrid approach whereby currently used standardized cryptographic schemes are combined with quantum-safe schemes.

Related Work. At the time of writing, two papers consider a quantum-safe Tor network: In (Ghosh and Kate, 2015), the proposed solution named HybridOR is a customized key exchange protocol. The solution is reported to be computationally more efficient compared to the currently used `ntor` protocol. In (Schanck et al., 2016), the focus is also on securely establishing the short-term keys in a quantum-safe fashion. The proposed protocol named Hybrid, uses a combination of long-term keys generated by Diffie-Hellman key exchange, and short-term keys generated by a quantum-safe scheme NTRUEncrypt. We observe that existing solutions focus on the prob-

lem of key exchange only. Furthermore, their performance study only focuses on the use of one particular quantum-safe cryptographic scheme. For example, Hybrid is evaluated using NTRUEncrypt only. Therefore there seems to be a lack of comparative study among different quantum-safe cryptographic primitives.

3 TOWARDS QUANTUM-SAFE TOR

In a quantum world, Tor users need to have the same security and anonymity guarantees as they currently have. A quantum-safe Tor network should provide these same guarantees against both quantum and classical adversaries. Current attack scenarios on Tor do not target the cryptography used in Tor but aim to exploit other potential weaknesses. However, powerful enough quantum computers will pose a new threat to Tor as cryptography becomes vulnerable for abuse by quantum adversaries. Introducing PQ cryptography to Tor must be done in order to keep cryptographic vulnerabilities off the list of attack surfaces. It is pivotal to introduce quantum-safe cryptography to the keys of the nodes.

Current Attack Scenarios. Current attacks on Tor do not target the cryptography but rather focus on vulnerabilities in Tor-related software, hidden services, bridge node discovery, disabling the network, and on generic attacks like timing. Typically, adversaries introduce new nodes to the Tor network. This is a lengthy process due to the policy of the network. New nodes are even more closely monitored than nodes already in the network for malicious patterns and if such is recognized, they are excluded from the network.

Cryptographic Attacks. We now consider attack scenarios on the keys of the nodes that a quantum adversary possesses. There are four types of keys in Tor (See Table 1) and all of these can be compromised:

- *short-term key.* By compromising a short-term key at an entry node an adversary can follow the entire circuit from sender to recipient, leading to deanonymizing the user. Since such keys are renewed at the end of TLS connections, such attack can be performed during the lifetime of its TLS connection.
- *medium-term key.* When an adversary compromises the short-term key and the medium-term key of a node, it can impersonate this node. Since a node can decrypt one layer of symmetric encryption when the messages are passed through it, the previous and next ‘hop’ in the circuit are dis-

covered. The attack has to be performed before the rotation of the medium-term and short-term keys.

- *long-term key.* The compromise of a long-term key would enable the adversary to impersonate the node and send forged descriptors to the directory nodes. This also allows gaining indefinite, full access to the node. Moreover, the adversary sees previous and consecutive ‘hops’ in the circuit with the encrypted cells.
- *symmetric key.* Symmetric keys are used to encrypt the data sent between nodes. In the current implementation of Tor, AES 128-bit is used⁶. Compromising the symmetric keys enables an adversary to decrypt layers of encryption and learn the destination of the message; anonymity is at risk. In case the attacker learns nothing but the symmetric keys, the encrypted message must be intercepted before entering the network as the TLS connection adds an extra layer of security. If an adversary does not learn all symmetric keys, but only a subset, then he cannot decrypt the message and thus, the circuit is not known, so source and destination remain anonymous.

Current attacks on Tor are carried out with colluding adversaries. If adversaries control the entry and exit nodes in the network, they can share information with each other and as a result deanonymize communicating parties. Colluding adversaries at the entry and at the exit node who have the medium-term keys will know the middle relay in a circuit. Sharing this knowledge enables them to attempt to deanonymize users, as the users using the common middle node have the greatest probability to be communicating with each other.

Migration Strategy for Quantum-safe Tor. Considering attackers’ capabilities and the lifetime of the asymmetric keys, it is most urgent to update the long-term keys to a quantum-safe alternative. Long-term keys remain unchanged for a long time-period. Hence, an adversary has more time to compromise long-term keys. The effects of compromising long-term keys are also greater, as an adversary can thereby impersonate a node. The second most urgent need, is updating the medium-term keys based on the available time period. Finally, the short-term keys must be considered, even though the attacker has limited time to compromise these keys due to the security restrictions of Tor. Furthermore, short-term keys are used with TLS, and there are works on making TLS quantum-safe (Bos et al., 2015). We do stress that

⁶<https://gitweb.torproject.org/torspec.git/tree/tor-spec.txt>, accessed on 14/06/2019.

it is crucial to update every asymmetric scheme to a quantum-safe alternative in order to enforce the security and anonymity claims of Tor. Lastly, we note that the symmetric keys must be updated to AES 256 bits to prevent ‘store now, decrypt later’-attacks (Bernstein, 2009) and ensure that users of Tor maintain life-long anonymity.

4 IMPACT OF PQ CRYPTOGRAPHY ON TOR

In this section, we investigate the impact that PQ cryptography might have on the Tor network, when following the previously suggested migration strategy. The impact of migrating all asymmetric cryptography to a quantum-safe alternative has an impact on the performance (both computational and network) and reliability of Tor. We focus, in particular, on the key exchanges as updating these has the greatest effect on the overall performance and reliability of Tor. We benchmark the PQ cryptographic schemes that have been implemented in the Open Quantum Safe library (Stebila and Mosca, 2017). We only tested the schemes that achieve level 1 NIST security (see Table 2).

System Setup. Local and virtual environments are both used. The technical specification of the notebook used for the local experiments is Dell Latitude E7240, with Intel Core i5-4310U CPU @ 2.00 - 2.60GHz processor, 8 GB RAM, Samsung SSD SM841N mSATA 128 GB for storage and Windows 10 Enterprise 64-bit operating system. Furthermore, an Ubuntu 18.04 LTS subsystem was installed. In order to emulate the Tor network, 6 virtual machines were used with Intel Core Processor (Broadwell) @ 2.4 GHz processors, 60 GB storage, a virtual network adapter, and Linux version 4.15.0 operating system.

Benchmarking Results. We performed measurements and obtained benchmarks for (i) public key, private key and ciphertext sizes (see table 3), (ii) RSA key generation, encryption and decryption, and, (iii) quantum safe key generation, encapsulation and decapsulation. To get an average result for the CPU cycle measurements, 1 000 iterations were run with each test. The number of CPU cycles corresponding to one second is 2 399 753 472.

Key lengths have an effect on network load as they are sent to the Directory Authorities who distribute them to the clients. Larger ciphertexts have a significant detrimental effect on the stability, reliability and performance of a network. From table 2, we observe that, both Frodo-640-AES and Frodo-640-SHAKE (Alkim et al., 2019b) have a prob-

Table 2: Key and Ciphertext sizes in Bytes.

Scheme	Public key	Private key	Ciphertext
RSA-1024	< 128	< 128	128
RSA-2048	< 256	< 256	256
Frodo-640-AES	9 616	19 888	9 720
Frodo-640-SHAKE	9 616	19 888	9 720
Kyber512	800	1 632	736
NewHope-512-CCA	928	1 888	1 120
NTRU-HPS-2048-509	699	935	699
Sike-p503	378	434	402

lematic ciphertext size of 9 720 bytes.

From Table 3, we observe that the lattice-based quantum-safe schemes (Kyber, NewHope, NTRU) require less CPU cycles for generating keys than RSA-1024. Kyber (Avanzi et al., 2017) and NewHope (Alkim et al., 2019a) drastically outperform the other schemes. We also note that the supersingular isogeny-based quantum-safe scheme Sike (Jao et al., 2019), even though slightly less performant than RSA-1024, outperforms RSA-2048. Key gen-

Table 3: CPU cycles for encapsulation, decapsulation and key generation averaged over 1 000 test runs.

Scheme	Encapsulation	Decapsulation	Key generation
RSA-1024	410 402	2 078 161	61 568 194
RSA-2048	730 570	5 718 858	266 140 623
Kyber512	170 856	195 106	152 973
NewHope-512-CCA	228 687	247 457	193 367
NTRU-HPS-2048-509	636 263	1 609 748	27 632 969
Sike-p503	149 691 623	159 119 760	90 800 645

eration only affects the nodes who generate them. The factor that affects both the nodes and the client is the time/computation needed to encapsulate and decapsulate messages. As opposed to key generation times, where all lattice-based schemes outperform RSA-1024, we observe that NTRU (Chen et al., 2019) requires more CPU cycles for encapsulation. Decapsulation for lattice-based implementations requires less CPU cycles than RSA-1024. Sike requires the most CPU cycles and is almost 48 times more computationally heavy than RSA-2048.

Lattice-based schemes (Kyber, NewHope, NTRU) have better performance for CPU cycles than the RSA schemes. This suggests that these are the most fit candidates for replacing classical cryptographic schemes. However, based on ciphertext sizes, Sike is the best fit as the ciphertext size fits within one Tor cell (512 bytes).

Impact. A migration of classical cryptography to quantum-safe cryptography can have a big effect on the overall availability, reliability, stability and performance of Tor. An important factor to take into account is the network load. The number of packets needed to transfer the ciphertexts increase with these schemes. The factor of computation time, on the other end, influences the response time to users. We note

that a trade-off has to be made between network load and computation time, when considering the schemes that we tested.

5 CASE STUDY: CIRCUIT CREATION IN TOR

To investigate the impact of PQ cryptography on Tor, we propose to investigate the performance of one particular protocol, namely circuit creation. Our framework uses the SweetOnions implementation⁷ which is a simplified version of the circuit creation protocol used in Tor. We consider two quantum-safe versions of this protocol: a version in which we only use post-quantum cryptography (QSO), and a hybrid version of the protocol (HSO) in which we combine the currently used cryptography with post-quantum cryptography. The reference implementation (SO) that uses standard cryptographic schemes, namely RSA, is also evaluated. We now provide a detailed description of each protocol. **Protocol descriptions** In the original SweetOnions protocol, defined in Protocol 1 for one layer, the client who aims to send a message m to node N , $N \in \mathbb{N}$, encapsulates the symmetric data encryption key using N 's public RSA key pk_N . To set up a circuit the client has to perform these steps with all the nodes in the circuit. Once the client knows the address of every node, each node between the client and destination sequentially decrypts one layer of encryption and forwards the message.

Protocol 1: Sweet Onion (SO).

Client (m)	Node (pk_N)
$K_{AES} \leftarrow_R \{0, 1\}^{256}$ $c \leftarrow Enc_{AES}(K_{AES}, m)$ $c' \leftarrow Enc_{RSA}(pk_N, K_{AES})$	$(c, c') \rightarrow$ $K_{AES} \leftarrow Dec_{RSA}(sk_N, c')$ $m \leftarrow Dec_{AES}(K_{AES}, c)$

QSO corresponds to the simple quantum-safe variant of SO: the RSA key encapsulation method (KEM) is exchanged with a post-quantum KEM (PQC). The public-private key pair of the node consists of post-quantum keys.

In the hybrid SweetOnions (HSO) protocol (Protocol 3), the RSA KEM is combined with a post-quantum KEM. The client randomly generates two symmetric encryption keys. The first key is encapsulated with RSA and the second key is encapsulated

with a PQ encryption algorithm. The actual data encryption key is the result of a simple XOR of these two symmetric keys. Hence, the receiver should perform two decapsulation operations (one with RSA and one with PQ decryption).

Protocol 2: Hybrid Sweet Onion (HSO).

Client (m)	Node (pk_N, pk_N^{PQ})
$K_{AES}^1, K_{AES}^2 \leftarrow_R \{0, 1\}^{256}$ $K_{AES} = K_{AES}^1 \oplus K_{AES}^2$ $c \leftarrow Enc_{AES}(K_{AES}, m)$ $c' \leftarrow Enc_{RSA}(pk_N, K_{AES}^1)$ $c'' \leftarrow Enc_{PQC}(pk_N^{PQ}, K_{AES}^2)$	$(c, c', c'') \rightarrow$ $K_{AES}^1 \leftarrow Dec_{RSA}(sk_N, c')$ $K_{AES}^2 \leftarrow Dec_{PQC}(sk_N^{PQ}, c'')$ $K_{AES} = K_{AES}^1 \oplus K_{AES}^2$ $m \leftarrow Dec_{AES}(K_{AES}, c)$

Experimental Results. We evaluate the performance of each protocol in terms of CPU and bandwidth consumption. The size of one Tor packet is 512 bytes. For the reference SO protocol, the underlying encryption algorithms are RSA-2048 and AES-192. For the two other protocols, the PQ cryptographic schemes studied in Section 4 are used. Experimental results are given in Table 4. In particular, we evaluate the cost of wrapping the layers of encryption, decapsulating one layer of encryption, and the overall circuit creation. The table also includes the size of one message and the number of packets needed for this protocol.

We observe that QSO always outperforms the original SO. On the other hand, while the integration of *Sike* increases the overall time significantly, the bandwidth overhead is very close to SO. Therefore, a lattice-based scheme may be considered as a potential cryptographic primitive for circuit building since it requires less CPU cycles than *Sike*. Nevertheless, the use of lattice-based schemes significantly increases the network load compared to *Sike*. Therefore, depending on the original communication cost, one can decide to choose *Sike* or a lattice-based PQC.

When using the hybrid scheme, we observe that both the computational cost and the bandwidth increase significantly. This is mainly due to the fact that HSO uses one RSA encapsulation and one encapsulation with PQC. Consequently, the cost originating from PQC for lattice-based cryptographic schemes becomes negligible when combined with RSA. Even though CPU consumption remains affordable in the hybrid implementation, the bandwidth overhead is important. The number of packets is at least doubled when switching to the hybrid solutions.

⁷<https://github.com/LeonHeTheFirst/SweetOnions>, accessed on 28/11/2019,

Table 4: The CPU cycles needed for building a circuit (averaged over 1 000 test runs) and message sizes.

Scheme	Wrap encryption layers	Remove one layer	Total circuit build	Message size (bytes)	Packets needed	Time needed
Original	5 131 765	13 714 147	46 274 206	1 223	3	0.0193s
Kyber	1 371 999	917 080	4 123 240	3 248	7	0.0017s
NewHope	1 618 934	1 119 668	4 977 938	4 832	10	0.0021s
NTRU	2 803 358	4 149 134	15 250 759	3 099	7	0.0064s
Sike	452 691 951	271 667 313	1 267 693 889	1 874	4	0.5283s
Hybrid Kyber	6 188 037	15 734 659	53 392 015	5 774	12	0.0222s
Hybrid NewHope	6 953 196	13 771 785	48 268 550	7 886	16	0.0201s
Hybrid NTRU	7 517 316	18 977 229	64 449 002	5 550	11	0.0269s
Hybrid Sike	456 441 243	275 867 016	1 284 042 291	3 938	8	0.5351s

6 CONCLUSION

In this paper, we investigated the main challenges to develop a quantum-safe Tor network and focused on the algorithms that use long-term and medium-term keys. Experimental studies show that among the six post-quantum cryptographic scheme evaluated, there is no single winning solution. Nevertheless, given the current status of the NIST standardisation process, Sike seems the most optimal one when it comes to assessing the communication overhead.

For future work, it may be interesting to test other schemes such as the code-based BIKE. Testing the remaining lattice and isogeny-based schemes is also an interesting future topic as they might have better performance measurements than the ones currently available in the Open Quantum Safe library. For field experiments, an implementation of Tor called TorLAB⁸ is available and simulates Tor on a private network of Raspberry Pis. It would be beneficial to re-create the network and extend the measurements of our research to the network load. This would ensure a more realistic study for the evaluation of expected circuit build times, since in the current setting, network latency is omitted.

REFERENCES

Alkim, E., Avanzi, R., Bos, J., Ducas, L., de la Piedra, A., Pöppelmann, T., Schwabe, P., Stebila, D., Albrecht, M. R., Orsini, E., Osheter, V., Paterson, K. G., Peer, G., and Smart, N. P. (2019a). NewHope. In *NIST Round 2 Submissions for Post-Quantum Cryptography Standardization*.

Alkim, E., Bos, J. W., Ducas, L., Longa, P., Mironov, I., Naehrig, M., Nikolaenko, V., Peikert, C., Raghunathan, A., and Stebila, D. (2019b). FrodoKEM. In

⁸<https://github.com/dws-pm/TorLAB>, accessed on 11/08/2019.

NIST Round 2 Submissions for Post-Quantum Cryptography Standardization.

Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., Schwabe, P., Seiler, G., and Stehlé, D. (2017). CRYSTALS-KYBER. In *NIST Round 1 Submissions for Post-Quantum Cryptography Standardization*.

Bernstein, D. J. (2009). *Introduction to post-quantum cryptography*. Springer Berlin Heidelberg.

Bos, J. W., Costello, C., Naehrig, M., and Stebila, D. (2015). Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem. In *Proceedings of the IEEE Symposium on Security and Privacy*.

Chen, C., Danba, O., Hoffstein, J., Hülsing, A., Rijneveld, J., Schanck, J. M., Schwabe, P., Whyte, W., and Zhang, Z. (2019). NTRU. In *NIST Round 2 Submissions for Post-Quantum Cryptography Standardization*.

Dingledine, R., Mathewson, N., and Syverson, P. (2004). Tor: The Second-Generation Onion Router. In *Proceedings of the 13TH USENIX Security Symposium*.

Ghosh, S. and Kate, A. (2015). Post-Quantum Forward-Secure Onion Routing (Future Anonymity in Today’s Budget). In *Proceedings of the International Conference on Applied Cryptography and Network Security (ACNS)*.

Jao, D., Azarderakhsh, R., Campagna, M., Costello, C., Feo, L. D., Hess, B., Jalali, A., Koziel, B., LaMacchia, B., Longa, P., Naehrig, M., Pereira, G., Renes, J., Soukharev, V., and Urbanik, D. (2019). Supersingular Isogeny Key Encapsulation. In *NIST Round 2 Submissions for Post-Quantum Cryptography Standardization*.

Katz, J. and Lindell, Y. (2007). *Introduction to Modern Cryptography*. Chapman & Hall/CRC.

NIST (2016). Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process. <https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf>.

Schanck, J. M., Whyte, W., and Zhang, Z. (2016). Circuit-extension handshakes for Tor achieving forwards se-

crecy in a quantum world. In *Proceedings on Privacy Enhancing Technologies (PETS)*.

Shor, P. W. (1995). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*.

Stebila, D. and Mosca, M. (2017). *Post-quantum Key Exchange for the Internet and the Open Quantum Safe Project*. In Roberto Avanzi, Howard Heys, editors, *Selected Areas in Cryptography (SAC) 2016*, LNCS, vol. 10532.

