IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 October 2022, accepted 2 December 2022, date of publication 7 December 2022, date of current version 23 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3227394

RESEARCH ARTICLE

Improving Bitcoin's Post-Quantum Transaction Efficiency With a Novel Lattice-Based Aggregate Signature Scheme Based on CRYSTALS-Dilithium and a STARK Protocol

YUNJIA QUAN
Charlotte Country Day School, Charlotte, NC 28226, USA
e-mail: angelinaquan2024@gmail.com

This work was supported in part by the Pioneer Academics Research Program.

ABSTRACT This paper proposes a novel lattice-based aggregate signature (LAS) scheme that brings post-quantum security to the Bitcoin system without sacrificing its transaction efficiency. Bitcoin currently employs Elliptic Curve Digital Signature Algorithm (ECDSA), which is insecure against the emerging quantum technology, so post-quantum signature schemes like the proposed LAS will become necessary in the near future. However, most of the post-quantum signatures schemes have large signature sizes which decrease Bitcoin's efficiency. Even CRYSTALS-Dilithium, the most prominent post-quantum signature scheme chosen by the National Institute of Standards and Technology (NIST), has this adverse limitation: it would cause Bitcoin's transaction efficiency to fall by 17 times from 2759.36622 transactions per block (tpb) to 159.48374 tpb. The existing signature schemes are unable to resolve this efficiency problem for Bitcoin. We crafted a novel LAS scheme based on CRYSTALS-Dilithium and a zero-knowledge Scalable Transparent Arguments of Knowledge (STARK) protocol to tackle this problem. The proposed LAS scheme takes full advantage of signature aggregation using the STARK protocol and Dilithium's easy and fast implementation, thus generating signatures with post-quantum security and small signature sizes which are critical to transaction efficiency. Our proofs convey the correctness, compactness, and post-quantum security of our scheme in the quantum random oracle model, and our implementation in Python conveyed that the proposed scheme would only decrease Bitcoin's transaction efficiency by 3 times, a significant improvement from using Dilithium and other lattice-based aggregate signature schemes. Our proposed scheme has many advantages over the existing schemes and will become very valuable to Bitcoin.

INDEX TERMS Digital signatures, post-quantum cryptography, lattice-based aggregate signature scheme, CRYSTALS-dilithium, STARK protocol, bitcoin blockchain.

I. INTRODUCTION

Digital signatures are essential to the security of Bitcoin and blockchain technology. Every valid Bitcoin transaction has to be signed by the sender and then recorded on the Bitcoin blockchain. Digital signatures ensure the security of Bitcoin transactions and enables robust and efficient methods of verification.

Currently, Bitcoin utilizes ECDSA to generate digital signatures that are secure against attacks from classical computers, but quantum computers can break ECDSA easily. Therefore, we investigate CRYSTALS-Dilithium, a Lattice-Based Digital Signature Scheme that can be utilized in a post-quantum setting. Dilithium was selected by NIST as the primary post-quantum digital signature algorithm, so it will likely be used for Bitcoin and other important areas of application [1].

The problem, however, is that Dilithium and other post-quantum signature schemes generate signatures that

The associate editor coordinating the review of this manuscript and approving it for publication was Diana Gratiela Berbecaru.

are much larger than signatures generated by ECDSA. An increase in signature size should result in an increase in the Bitcoin block size, but since Bitcoin has a block size limit of 1MB, this increase in block size will not be possible. Instead, we will have to decrease the number of transactions per block to maintain the 1MB block size, which will cause transaction efficiency to fall.

Since using Dilithium decreases Bitcoin's post-quantum transaction efficiency (measured by the number of transactions per block), we wish to craft an aggregate signature scheme that will increase Bitcoin's transaction efficiency from that of Dilithium. Using the proposed signature scheme, we hope to generate one compact signature for multiple transactions in the same Bitcoin block.

A. RELATED WORKS

Digital signatures are essential to the security of Bitcoin and blockchain technology. Every valid Bitcoin transaction has to be signed by the sender and then recorded on the Bitcoin blockchain. Digital signatures ensure the security of Bitcoin transactions and enables robust and efficient methods of verification.

Currently, Bitcoin utilizes ECDSA to generate digital signatures that are secure against attacks from classical computers, but quantum computers can break ECDSA easily. Therefore, we investigate CRYSTALS-Dilithium, a Lattice-Based Digital Signature Scheme that can be utilized in a post-quantum setting. Dilithium was selected by NIST as the primary post-quantum digital signature algorithm, so it will likely be used for Bitcoin and other important areas of application [1].

The problem, however, is that Dilithium and other post-quantum signature schemes generate signatures that are much larger than signatures generated by ECDSA. An increase in signature size should result in an increase in the Bitcoin block size, but since Bitcoin has a block size limit of 1MB, this increase in block size will not be possible. Instead, we will have to decrease the number of transactions per block to maintain the 1MB block size, which will cause transaction efficiency to fall.

Since using Dilithium decreases Bitcoin's post-quantum transaction efficiency (measured by the number of transactions per block), we wish to craft an aggregate signature scheme that will increase Bitcoin's transaction efficiency from that of Dilithium. Using the proposed signature scheme, we hope to generate one compact signature for multiple transactions in the same Bitcoin block.

B. RELATED WORKS

Ever since lattice-based cryptography was introduced in 1996, there have been many studies concerning lattice-based aggregate signatures.

There are two types of aggregate signatures: general and sequential. Sequential aggregate signature schemes require a strict order to exist among individual signatures, and most of the sequential aggregate signature schemes are developed based on the Rivest-Shamir-Adleman (RSA) problem or bilinear maps, which makes them vulnerable to quantum attacks. There are, however, sequential aggregate lattice-based signature schemes with quantum-security based on lazy verification [2], FALCON based trapdoor functions [3], or NTRUSign [4], but in these schemes, each user has to verify the signature of the user prior to them which decreases the efficiency of the schemes.

On the other hand, in 2012, Zhang et al introduced a homomorphic technique for unordered lattice-based aggregate signature scheme, but all signers in this scheme have to have the same public key in order for the i^{th} signer to sign a message m_j without the j^{th} signer knowing anything about it [5]. In 2014, Jing et al also proposed a lattice-based homomorphic unordered aggregate signature scheme that reduces signature length and increases efficiency [6]. However, a problem for both of these schemes is that every signature of the aggregate signature can sign messages on behalf of other signers without their consent. This is exceedingly detrimental for Bitcoin transactions, for users may lose a tremendous amount of money to adversaries.

In 2016, Bansarkhani et al proposed the first unordered interactive lattice-based aggregate signature scheme that is provably secure in the random oracle model [7], and in 2018, Lu et al proposed an unordered lattice-based aggregate signature scheme based on the intersection method [8]. However, these schemes are not applied to other protocols such as a STARK protocol to produce zero-knowledge proofs that protects the privacy of Bitcoin transaction data.

There have been many protocols and services that can protect Bitcoin anonymity. For example, Bao et al proposed Lockmix, a secure mix service for Bitcoin [9], and in 2012, Bitansky et al proposed zero-knowledge Succinct Non-interactive Argument of Knowledge (SNARK) [10]. However, the STARK protocol has advantage over these other protocols and services because STARK is capable of aggregating signatures with a zero-knowledge proof while preserving post-quantum security, which a lot of other services and protocols lack the ability to do so. Thus, we wish to utilize a STARK protocol alongside our scheme.

Since there are little lattice-based aggregate signature schemes that can produce a zero knowledge proof along with a fast implementation process and small signature size, we seek to devise a scheme that will achieve this and improve Bitcoin's transaction efficiency.

C. RESEARCH AIMS AND SCOPE

The aim of this study is to craft a lattice-based aggregate signature scheme in order to improve the transaction efficiency of Bitcoin and gain post-quantum security and to identify the impacts of transitioning from ECDSA to CRYSTALS-Dilithium on Bitcoin's transaction efficiency.

This paper deals with a lattice-based aggregate digital signature scheme, proofs of the scheme's security, and the scheme's implementation. These fall under the studies of cryptography, mathematics, and computer science, respectively. Knowledge about post-quantum lattice-based cryptography was used to construct the scheme, mathematical knowledge such as abstract algebra was used to write the proofs of the scheme's construction, and coding skills helped implement the scheme and obtain results.

The scope of the study, however, is limited to signature schemes for Bitcoins, and not other cryptocurrencies. We also focused on lattice-based signatures specifically due to their efficient functionality for post-quantum Bitcoin and NIST's recommendation and did not investigate other post-quantum signatures. We also chose to involve a STARK protocol to produce zero-knowledge proofs a did not apply our LAS scheme with other protocols.

D. OUR CONTRIBUTION

In this paper, we calculated how shifting from ECDSA to CRYSTALS-Dilithium would affect Bitcoin's transaction efficiency. We investigated different transaction types (P2PKH, P2PK, and P2SH) and computed their transaction sizes under the two different signature schemes. With that, we found the transaction efficiency of the two schemes.

After our discovery that CRYSTALS-Dilithium will decrease Bitcoin's transaction efficiency by 17 times, we devised our lattice-based aggregate signature scheme. We formulated the algorithms in the scheme based on Dilithium and applied it with a STARK protocol. We used proofs of compactness, correctness, and post-quantum security to show that our scheme is compact, correct, and secure in the quantum random oracle model because our scheme is based on the hardness of the Module Learning With Errors (MLWE) and Module Short Integer Solution (MSIS) problems.

We implemented our scheme with a set of parameters described in the paper and obtained the result that our proposed scheme only decreases Bitcoin's transaction efficiency by 3 times, which is a notable improvement from Dilithium. We also compared our scheme with several other lattice-based aggregate signature schemes and found that our scheme is the most efficient. Therefore, in a post-quantum era, our scheme will be important for Bitcoin.

E. PAPER ORGANIZATION

The rest of this paper is organized as follows. Section II introduces preliminaries regarding different signature schemes and their role in the Bitcoin system. Starting from the basis of a signature scheme, we delve into aggregate signature schemes, proofs for signature schemes, and CRYSTALS-Dilithium. We also present the zero knowledge STARK protocol that we utilize later with our LAS construction. Section III shows our calculations of how Bitcoin's transaction efficiency is expected to fall when transitioning to CRYSTALS-Dilithium. Section IV concerns the specifics of our LAS construction and proofs for our construction. Section V is the implementation of our LAS scheme and compares our scheme's efficiency to other post-quantum schemes: our proposed scheme is more efficient and does not cause Bitcoin's transaction efficiency to fall as much. We also discuss the strengths and shortcomings of our proposed scheme. Section VI summarizes our work and concludes the discussion of our proposed LAS scheme.

II. PRELIMINARIES

A. BITCOIN AND BLOCKCHAIN

Bitcoin is a digital, secure, and decentralized medium of exchange that utilizes a peer-to-peer electronic cash system without relying on trust: the Bitcoin system employs blockchain technology to record transactions on blocks and then utilizes cryptographic methods such as digital signature schemes to construct transactions. Transactions are the foundations of the Bitcoin system, for every component of Bitcoin is designed to ensure the safety and efficiency of transactions.

Bitcoin has one blockchain, which is a global ledger of transactions that are maintained across several computers linked in a peer-to-peer network. Each block in the blockchain is composed of a block header and different transactions, and the size of a transaction is determined by the input, output, and scripts of that transaction, which we will discuss in Section III. There are many elements to a transaction, and the elements that we are the most concerned with are the public key, private key, hash, and signature, for they are most frequently utilized in digital signature schemes and transactions.

Bitcoin utilizes a hash algorithm to ensure data integrity, public keys corresponding to Bitcoin user addresses, private keys for users to prove ownership over their Bitcoins, and signatures for users to sign transactions and verify the Bitcoin transactions.

In this study, we are also concerned with Bitcoin's scalability problem: Bitcoin's limited capability to process large amounts of transactions in a short period of time [12]. Bitcoin has a block size limit of 1MB due to this scalability problem, and Bitcoin can only process 3-7 transactions per second. When using a post-quantum signature scheme, the increase in signature size would either cause the block size to increase or the transactions per block will decrease. The latter one will take place, for the block size limit may not be surpassed, and transaction efficiency will be sacrificed. Thus, we seek to craft a lattice-based signature scheme that will bring down the post-quantum Bitcoin signatures' size and increase the transaction efficiency.

B. SIGNATURE SCHEMES

Signatures in Bitcoin transactions, are very important, as they ensure that the transaction is verified, authentic, and legitimate, and signature schemes are used to generate unique signatures for different transactions. We will start with the basic definition of a signature scheme [13].

*Definition 1 (Signature Scheme): A signature scheme consists of three algorithms: Key Generation (**KeyGen**), Signing (**Sign**), and Verification (**Verify**). In particular:*

- ***KeyGen**: A private key sk and a public key pk are chosen.*
- ***Sign**: With the secret key sk, a user can sign the transaction m by using this algorithm to generate a signature σ*
- ***Verify**: Given the public key pk, a transaction m, and a signature σ, anyone can verify whether this signature is the corresponding signature to this transaction.*

The signature scheme described above is the backbone of more complex signature schemes. Most signature schemes have these three algorithms or a variation of them, and those schemes may have additional algorithms for different purposes. Since we are constructing an aggregate signature scheme, we will explain how an aggregate signature scheme differ from this basic signature scheme.

*Definition 2 (Aggregate Signature Scheme): An aggregate signature scheme is a tuple of five algorithms. Three of the algorithms (**KeyGen, Sign, and Verify**) are the same as the algorithms in the most basic signature scheme, and two additional algorithms are defined below.*

Let $PK = (pk_1, pk_2, \ldots, pk_n)$, $M = (m_1, m_2, \ldots, m_n)$, and $\Sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$.

- ***AggSig**: Take n different transactions M, their corresponding signatures Σ (generated by Sign), and the corresponding public keys PK, the user will get one aggregated signature σ_{agg} as output.*
- ***AggSigVerify**: Given one aggregate signature σ_{agg}, the user can verify this signature with PK and M.*

Every aggregate signature scheme has to preserve compactness, correctness, and unforgeability, which we define below.

*Definition 3 ([13] Compactness): Let π = (**KeyGen, Sign, SignAgg, Verify, AggSigVerify**) be an aggregate signature scheme. π is compact if there exists a polynomial $f(x)$ and a negligible function $g(x)$ such that for every security parameter λ and sets of messages $M = \{m_1, m_2, \ldots, m_n\}$, we get that*

$$Pr[|\sigma_{ag}| \leq f(\lambda, \log(n))] = 1 - g(\lambda) \tag{1}$$

where Pr is the probability of the indicated inequality holding, $|\sigma_{ag}|$ is the bit length of σ_{ag}, and the negligible function $g(x)$ is defined below.

Definition 4 ([14] Negligible Function): a function $f(x)$: $\mathbb{Z} \to \mathbb{R}$ is negligible if for every positive integer a, we have

$$\lim_{n\to\infty} f(n)n^a = 0 \tag{2}$$

In other words, for all positive integer a, there exists $m \in \mathbb{Z}_{\geq 1}$ such that all integers $n \geq m$ satisfies $|f(n)| \leq \frac{1}{n^a}$.

*Definition 5 ([13] Correctness): Let π = (**KeyGen, Sign, SignAgg, Verify, AggSigVerify**) be an aggregate signature scheme. π is correct if for all security parameters $\lambda \in \mathbb{N}$ and any number of messages $n \in \mathbb{N}$, we have*

$$Pr[AggSigVerify(PK, M, AggSig(PK, M, \Sigma)) = 1] = 1 \tag{3}$$

where everything is defined as in Equation 1.

*Definition 6 ([13] Unforgeability): Let π = (**KeyGen, Sign, SignAgg, Verify, AggSigVerify**) be an aggregate signature scheme. π is unforgeable if for any admissible adversaries $\mathcal{A}$ we have*

$$Pr[UE[\lambda, \mathcal{A}] = 1] = g(\lambda) \tag{4}$$

- *An admissible adversary $\mathcal{A}$ is an adversary that returns a verifying set of PK and M and a signature σ^* such that...*
    1) *For some arbitrary $i^* \in [n] = \{1, 2, \ldots, n\}$, $pk^* = pk_{i^*}$*
    2) *For all $i^* \in [n]$, m_{i^*} was never added to the signing oracle of the adversary*
- *$UE[\lambda, \mathcal{A}]$ is an unforgeability experiment of the signature scheme:*
    1) *Public parameters are generated with λ*
    2) *KeyGen generates the secret key sk* and public key pk**
    3) *The admissible adversary generates (PK, M, σ_{ag}) with the public parameters and public key pk^* through $\mathcal{A}^{Sign(sk^*, \cdot)}$*
    4) *The experiment outputs AggSigVerify(PK, M, σ_{ag})*

C. CRYSTALS-DILITHIUM: A LATTICE-BASED DIGITAL SIGNATURE SCHEME

CRYSTALS-Dilithium is a lattice-based digital signature scheme that consists of three algorithms: Key Generation, Signing, and Verification, and the security of the scheme is based on the hardness of the MLWE and MSIS problems on a lattice. Dilithium follows a Schnorr framework with a rejection-sampling step, making the signature size relatively small. All operations in Dilithium are done in the ring $R = \mathbb{Z}_q[X]/(X^{256} + 1)$ with $q = 2^{23} - 2^{13} + 1$, and SHAKE-256 is used for hashing. Moreover, all sampling (the process of choosing arbitrary values for variables in a certain range) done in Dilithium is uniform [15].

- **KeyGen**: The public key pk and private key sk are generated ($pk = (A, t_1)$, $sk = (s_1, s_2)$) where
    - A is a random matrix of size 4×4 that is stored in the Number Theoretic Transform (NTT) Domain Representation for the sake of easy and fast implementation. See explanation in Section II-C1. Note that the matrix sizes can vary for different Dilithium security levels.
    - s_1 and s_2 are chosen from $[-n, n]^4$ and $[-n, n]^4$ respectively
    - t_1 is the highest order bit of $t = As_1 + s_2$
- **Sign**: Given the secret key sk and a message M, the algorithm outputs a signature $\sigma = (z, h, c)$ where
    - c is the hash of the highest order bit of Ay for some arbitrary y in $[-\gamma, \gamma]^5$
    - $z = y + cs_1$
    - h is a carry bit hint vector, meaning that it is a 1-bit hint that allows us to recover the highest bit of

$Ay - cs_2$ without $-ct_0$ - we just need h and $Ay - cs_2 + ct_0$ - because $high(Ay - cs_2) = high(Ay - cs_2 + ct_0)$; high is the function that gives the highest order bit and t_0 is the lowest order bit of t

Moreover, the following conditions must be satisfied. If any of the conditions are not satisfied, we need to choose our y again:

- $\|z\| < \gamma - \beta$ for $\beta = \max \|cs_2\|$
- The lowest order bit of $(Ay - cs_2) < \gamma - \beta$

- **Verify**: Takes the private key sk, a message M, and a signature σ and outputs a boolean expression that shows whether the signature is valid or not. The expression is as follows

$$\|z\| \leq \gamma - \beta \text{ and } c = \text{hash of } Az - ct \text{ and the number of 1's in } h \leq \omega \tag{5}$$

where ω is a constant set by us.

See Figure 1 below for a comprehensive overview of CRYSTALS-Dilithium in psuedocode.

```
Gen
$\rho \leftarrow \{0,1\}^{256}$
$K \leftarrow \{0,1\}^{256}$
$(\mathbf{s}_1, \mathbf{s}_2) \leftarrow S_\eta^\ell \times S_\eta^k$
$\mathbf{A} \in R_q^{k\times\ell} := \mathsf{ExpandA}(\rho)$          // $\mathbf{A}$ is stored in NTT Domain Representation
$\mathbf{t} := \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$
$(\mathbf{t}_1, \mathbf{t}_0) := \mathsf{Power2Round}_q(\mathbf{t}, d)$
$tr \in \{0,1\}^{384} := \mathsf{CRH}(\rho \,\|\, \mathbf{t}_1)$
return $(pk = (\rho, \mathbf{t}_1), sk = (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0))$

Sign$(sk, M)$
$\mathbf{A} \in R_q^{k\times\ell} := \mathsf{ExpandA}(\rho)$          // $\mathbf{A}$ is stored in NTT Domain Representation
$\mu \in \{0,1\}^{384} := \mathsf{CRH}(tr \,\|\, M)$
$\kappa := 0, (\mathbf{z}, \mathbf{h}) := \perp$
while $(\mathbf{z}, \mathbf{h}) = \perp$ do
    $\mathbf{y} \in S_{\gamma_1 - 1}^\ell := \mathsf{ExpandMask}(K \,\|\, \mu \,\|\, \kappa)$
    $\mathbf{w} := \mathbf{A}\mathbf{y}$
    $\mathbf{w}_1 := \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)$
    $c \in B_{60} := \mathsf{H}(\mu \,\|\, \mathbf{w}_1)$
    $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$
    $(\mathbf{r}_1, \mathbf{r}_0) := \mathsf{Decompose}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)$
    if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ or $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$ or $\mathbf{r}_1 \neq \mathbf{w}_1$, then $(\mathbf{z}, \mathbf{h}) := \perp$
    else
        $\mathbf{h} := \mathsf{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$
        if $\|c\mathbf{t}_0\|_\infty \geq \gamma_2$ or the # of 1's in $\mathbf{h}$ is greater than $\omega$, then $(\mathbf{z}, \mathbf{h}) := \perp$
    $\kappa := \kappa + 1$
return $\sigma = (\mathbf{z}, \mathbf{h}, c)$

Verify$(pk, M, \sigma = (\mathbf{z}, \mathbf{h}, c))$
$\mathbf{A} \in R_q^{k\times\ell} := \mathsf{ExpandA}(\rho)$          // $\mathbf{A}$ is stored in NTT Domain Representation
$\mu \in \{0,1\}^{384} := \mathsf{CRH}(\mathsf{CRH}(\rho \,\|\, \mathbf{t}_1) \,\|\, M)$
$\mathbf{w}_1' := \mathsf{UseHint}_q(\mathbf{h}, \mathbf{A}\mathbf{z} - c\mathbf{t}_1 \cdot 2^d, 2\gamma_2)$
return $[\![\|\mathbf{z}\|_\infty < \gamma_1 - \beta]\!]$ and $[\![c = \mathsf{H}(\mu \,\|\, \mathbf{w}_1')]\!]$ and $[\![\#$ of 1's in $\mathbf{h}$ is $\leq \omega]\!]$
```

FIGURE 1. Psuedocode overview of CRYSTALS-Dilithium scheme [15].

1) NTT DOMAIN REPRESENTATION

In order to work with matrix A, we use implementation via Number Theoretic Transform (NTT) in our selected ring, for it is very efficient. NTT is a variation of Fast Fourier Transform (FFT): instead of working in the complex field, we work in the finite field $\mathbb{Z}_q$.

First, we choose a prime $q \equiv 1 \pmod{512}$ so that there is an element r in the group $\mathbb{Z}_q^*$ that is a 512-th root of unity. Then, we get that $x^{256} + 1 = (x - r)(x - r^3) \ldots (x - r^{511})$. By the Chinese Remainder Theorem, we get the following isomorphism:

$$Z_q \rightarrow \prod_{i=0}^{255} \mathbb{F}_q[X]/(X - r^{2i+1})$$

Now, any polynomial $a \in \mathbb{Z}_q[X]/(X^{256} + 1)$ can be represented by $(a(r), a(r^3), \ldots, a(r^{511}))$, and thus the product of polynomials is coordinate-wise. Therefore, the polynomial multiplications that involve matrix A can be computed easily with the help of the FFT, which is defined below.

Definition 7 ([16] Fast Fourier Transform (FFT)): Let $f \in \mathbb{Q}[x]/(\phi)$. For our purpose, let $\phi = x^{256} + 1$ and let Ω_ϕ be the set of complex roots of ϕ. Since $\phi(x) = \prod_{\zeta \in \Omega_\phi}(x - \zeta)$ by Equation 6, we get that $FFT_\phi(f)$, the fast Fourier transform of f with respect to ϕ, can be denoted below:

$$FFT_\phi(f) = (f(\zeta))_{\zeta \in \Omega_\phi}$$

As stated above, polynomial addition, subtraction, multiplication, and division modulo ϕ can be computed very efficiently in FFT because we can perform them by considering their components.

In short, Dilithium is a lattice-based signature scheme that is easy to implement, fast, compact, and quantum-resistant.

D. STARK PROTOCOL: SCALABLE TRANSPARENT ARGUMENTS OF KNOWLEDGE

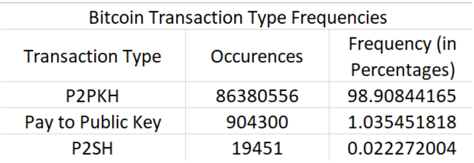
A Scalable Transparent Argument of Knowledge (STARK) protocol is a hash-based verification method that uses minimal resources and that provides post-quantum security. We use this protocol with the proposed LAS scheme in order to ensure its security and produce zero-knowledge proofs alongside the signatures. The STARK protocol is defined below by a tuple of three algorithms: (**Setup, Prove, Verify**)

- **Setup**: Outputs the public parameters pp randomly when given a security parameter λ and Prog is the any program that utilizes public randomness and outputs algebraic intermediate representations of the input such that $Prog\{0, 1\}^* \rightarrow \{0, 1\}$
- **Prove**: Takes pp, a statement $stmt$, and the set of signatures Σ such that $Prog(stmt \| \Sigma) = 1$. Generates a proof π as output.
- **Verify**: Takes pp, a statement $stmt$, and a proof π as input. Output is a boolean variable that denotes whether this proof is valid or not.

Note that the STARK.Setup is transparent beacuse it only relies on public randomness and satisfies standard security in a random oracle including completeness and knowledge extraction [17], [18].

III. BITCOIN's EXPECTED TRANSACTION EFFICIENCY USING ECDSA VS CRYSTALS-DILITHIUM

The size of signatures generated by CRYSTALS-Dilithium differ for different security levels. In this paper, we consider Dilithium2 (Dilithium with a NIST security level of 2) which offers enough quantum security and has a relatively small signature size and public key size. Dilithium2 has a signature



Bitcoin Transaction Type Frequencies		
Transaction Type	Occurences	Frequency (in Percentages)
P2PKH	86380556	98.90844165
Pay to Public Key	904300	1.035451818
P2SH	19451	0.022272004

FIGURE 2. Frequency of different transaction types in Bitcoin.

size of around 2420B, which is about 33 times more than the size of signatures generated by ECDSA - which is approximately 72.5 bytes. It is obvious that the Bitcoin transaction efficiency measured in transactions per block would have to decrease with Dilithium instead of ECDSA, but we wish to find out the extent of that decrease in order to compare to how our LAS scheme would decrease the transaction efficiency. To the best of our knowledge, there has not been a study on Bitcoin's transaction efficiency using CRYSTALS-Dilithium compared to using ECDSA.

In our calculations, we consider three types of Bitcoin transactions: Pay-to-Public-Key-Hash (P2PKH), Pay-to-Public-Key (P2PK), and Pay to Script Hash (P2SH), for these three are the most frequently seen transactions [19]. With data on the frequencies of different transaction types in Bitcoin [20], we create the following table.

For each transaction type, we calculate their transaction size with two inputs and two outputs when using ECDSA and when using CRYSTALS-Dilithium with the following set of equations:

- For all transaction size S_T, we get that

$$S_T = S_I + S_O$$

where S_I is the size of the input with scripts and S_O is the size of the output with scripts
- **P2PKH**: Let s be the signature size and let pkh be the size of the public key hash.

$$\begin{aligned} S_I &= 40 + 1 + 1 + s + pkh \\ S_O &= 8 + 1 + pkh \end{aligned}$$

- **Pay to Public Key**: Let s be the signature size and let pk be the size of the public key.

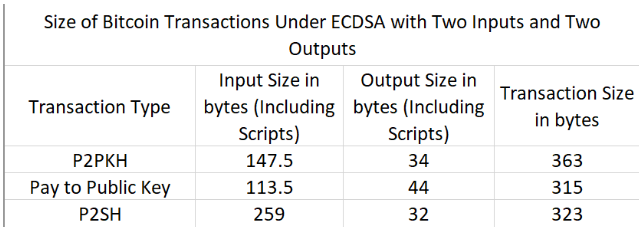
$$\begin{aligned} S_I &= 40 + 1 + 1 + s \\ S_O &= 8 + 1 + 2 + pk \end{aligned}$$

- **P2SH**: Let s be the signature size and pk be the public key size.

$$\begin{aligned} S_I &= 3 + 2 \times s + 40 + 2 \times (pk + 1) + 1 \\ S_O &= 32 \end{aligned}$$

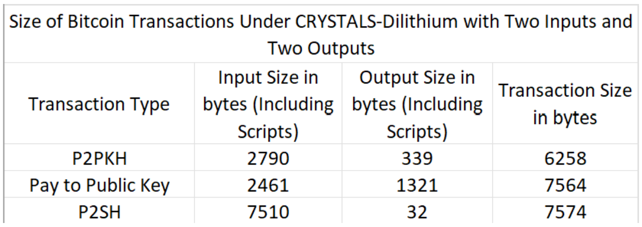
With the equations above, we get the following tables of values of transaction sizes.

Now, we combine the tables above to obtain the increase in Bitcoin block size if we did not decrease the number of transactions per block in compensation. By adding the products of the frequency of each transaction type and their corresponding transaction size, we get the average transaction size. Then, an average block size is calculated by adding 85 with 2759.12 times the transaction size, since there is an average of 2759.12 transactions per block.



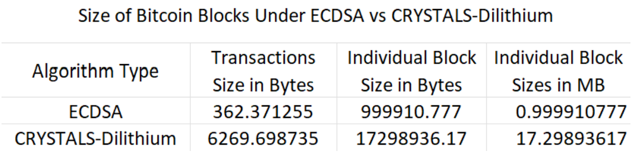
Size of Bitcoin Transactions Under ECDSA with Two Inputs and Two Outputs			
Transaction Type	Input Size in bytes (Including Scripts)	Output Size in bytes (Including Scripts)	Transaction Size in bytes
P2PKH	147.5	34	363
Pay to Public Key	113.5	44	315
P2SH	259	32	323

FIGURE 3. Bitcoin transaction sizes using ECDSA.



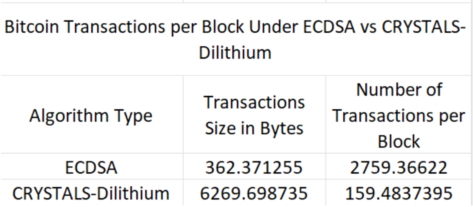
Size of Bitcoin Transactions Under CRYSTALS-Dilithium with Two Inputs and Two Outputs			
Transaction Type	Input Size in bytes (Including Scripts)	Output Size in bytes (Including Scripts)	Transaction Size in bytes
P2PKH	2790	339	6258
Pay to Public Key	2461	1321	7564
P2SH	7510	32	7574

FIGURE 4. Bitcoin transaction sizes using CRYSTALS-Dilithium.



Size of Bitcoin Blocks Under ECDSA vs CRYSTALS-Dilithium			
Algorithm Type	Transactions Size in Bytes	Individual Block Size in Bytes	Individual Block Sizes in MB
ECDSA	362.371255	999910.777	0.999910777
CRYSTALS-Dilithium	6269.698735	17298936.17	17.29893617

FIGURE 5. Average Bitcoin block size using ECDSA vs CRYSTALS-Dilithium.



Bitcoin Transactions per Block Under ECDSA vs CRYSTALS-Dilithium		
Algorithm Type	Transactions Size in Bytes	Number of Transactions per Block
ECDSA	362.371255	2759.36622
CRYSTALS-Dilithium	6269.698735	159.4837395

FIGURE 6. Average Bitcoin transactions per block using ECDSA vs CRYSTALS-Dilithium.

Now, we decrease the number of transactions per block as a remedy for the increase in block size so that the final block size would stay at 1MB. The average number of transactions per block using CRYSTALS-Dilithium is calculated below.

Finally, as seen in Figure 6, the transaction efficiency decreased by around 17 times under Dilithium compared to ECDSA.

IV. A LATTICE-BASED AGGREGATE SIGNATURE (LAS) SCHEME BASED ON CRYSTALS-DILITHIUM AND A STARK PROTOCOL

A. MAIN IDEA

The LAS scheme is a lattice-based aggregate scheme that will aggregate signatures on a Bitcoin block so that the

Bitcoin transaction efficiency in a post-quantum era would increase. Therefore, we want to make sure that LAS maintains post-quantum security with Dilithium and a STARK protocol and that our LAS scheme can successfully generate one compact signature for different transactions.

B. CONSTRUCTION

Let polynomial ring $\mathbb{Z}_q[X]/(X^{256}+1)$ with $q = 2^{23} - 2^{13} + 1$ be the ring that we perform all calculations over. We construct LAS through five algorithms: (**LAS.KeyGen**, **LAS.Sign**, **LAS.Verify**, **LAS.AggSig**, **LAS.AggSigVerify**).

- **LAS.KeyGen**() $\rightarrow$ ($pk = (A, t_1), sk = (s_1, s_2)$): defined the same way as KeyGen of CRYSTALS-Dilithium in Section II-C
- **LAS.Sign**(sk, M) $\rightarrow$ ($\sigma = (z, h, c)$): defined the same way as Sign of CRYSTALS-Dilithium in Section II-C
- **LAS.Verify**($pk, M, \sigma = (z, h, c)$) $\rightarrow$ (0/1): The verification process for z and c is defined the same way as in Section II-C. The verification condition for h is as follows:

$$\|h\| = \sqrt{\sum_{i \in |h|} h_i^2} \le \omega^{\frac{1}{2}} \tag{6}$$

- **LAS.AggSig**(PK, M, Σ) $\rightarrow$ (σ_{ag}): $\sigma_i = (z_i, h_i, c_i)$ and thus we generate the aggregate signature by considering each element of the tuple individually.

$$(z_{\text{ag}}, c_{\text{ag}}) = \sum_{i=1}^{n} v_i \times (z_i, c_i) \tag{7}$$

Note that $v_i = H(PK\|M) \in \mathbb{B}_{\beta_1}$ where H is the hash function SHAKE-256, which maps anything to $\{0, 1\}^*$ so that the output would have a smaller size than the input.

$$h_{\text{ag}} = \sum_{i=1}^{n} h_i \tag{8}$$

Such that $\|h_{\text{ag}}\| = \sum_{i=1}^{n} \|h_i\|$ because we add h_i in $\mathbb{Z}$ and not $\mathbb{B}$

- **LAS.AggSigVerify**($PK, M, \sigma_{\mathbf{ag}}$) $\rightarrow$ (0/1): Takes in the set of public key, messages, and the aggregate signature, and output a boolean variable which verifies whether a signature is valid or not. In order to perform calculations, we parse $\sigma_{\text{ag}} = (z_{\text{ag}}, h_{\text{ag}}, c_{\text{ag}})$. The conditions to be checked for the boolean variable are as follows:

$$z_{\text{ag}} \le (\gamma - \beta) \sum_{i=1}^{n} v_i \tag{9}$$

$$c_{\text{ag}} = \sum_{i=1}^{n} v_i \times H(w_i \| \mu_i) \tag{10}$$

  - $w_i = High(Ay_i)$
  - $\mu_i = H_{CR}(H_{CR}(\rho \| High(t_i)) \| M)$ for H_{CR} is a collision resistant hash function that maps any input to $\{0, 1\}^{384}$

$$\|h_{\text{ag}}\| \le n\omega^{\frac{1}{2}} \tag{11}$$

Recall the STARK protocol in Section II-D. Now, we want to use that protocol in conjunction with the LAS construction. We get the following procedure for key generation, signature generation, and verification:

1) Use LAS.KeyGen and LAS.Sign to generate the variables needed
2) $Prog((n, M, PK, LAS.pp, \Sigma))$
$= \prod_{i=1}^{n} LAS.Verify(LAS.pp, M_i, pk_i, \sigma_i)$ where
   - pp is the set of public parameters for LAS
   - n is the number of signatures that we want to aggregate
   - M, PK, Σ is the set of messages, public keys, and signatures, respectively
3) $STARK.Setup(1^{\lambda}, Prog)$ gives the public parameters pp for the overall scheme
4) $STARK.Prove(pp, stmt, \Sigma)$ generates a proof π when $stmt = (n, M, PK, Las.pp)$
5) $STARK.Verify(pp, stmt, AggSig) = LAS.AggSigVerify$ $(pp, n, M, PK, AggSig)$ with $AggSig = LAS.AggSig$ (PK, M, Σ) confirms whether an aggregated signature is valid or not.

See Figure 7 below for the overarching framework of the LAS scheme.

1) PROOFS FOR CONSTRUCTION

In order to prove that the proposed LAS scheme is a valid aggregate signature scheme, we must prove that it possesses compactness, correctness, and unforgeability in the quantum random oracle model. Moreover, we will discuss the lattice problem that LAS is based on.

Theorem 8 (Compactness): Suppose that the parameters $\lambda_1, \beta, \beta_1, \omega = f(\lambda) \ge 0$ are set as any polynomial functions in λ, then the aggregate signature scheme in Construction IV-B is compact (Definition 1).

Proof:

Let λ be the security parameter, n be the number of signatures to aggregate, and everything else defined as in Section IV-B. we parse σ_i for $i \in [n]$ into (z_i, h_i, c_i). We bound the norm of $\sigma_{\text{ag}} = (z_{\text{ag}}, h_{\text{ag}}, c_{\text{ag}})$ using the triangle inequality:

$$\|z_{\text{ag}}\| = \left\| \sum_{i \in [n]} z_i \cdot v_i \right\| \le Nn\|z_i\|\|v_i\| = 256n(\lambda_1 - \beta)(\beta_1)$$

$$\|c_{\text{ag}}\| = \left\| \sum_{i \in [n]} c_i \cdot v_i \right\| \le Nn\|c_i\|\|v_i\| = 256n(60)(\beta_1)$$

$$\|h_{\text{ag}}\| \le n\omega^{\frac{1}{2}}$$

Therefore, the size of $\sigma_{\text{ag}} = (z_{\text{ag}}, h_{\text{ag}}, c_{\text{ag}})$ can be expressed by $f(\lambda, n)$ if $\lambda_1, \beta, \beta_1, \omega = f(\lambda) \ge 0$. □

Theorem 9 (Correctness): The aggregate signature scheme in Construction IV-B is correct (Definition 3).

Proof: Let λ be the security parameter, n be the number of signatures to aggregate, and everything else defined just as in Section IV-B. Similar to the compactness proof, we parse all σ into (z, h, c), and consider the verification process of

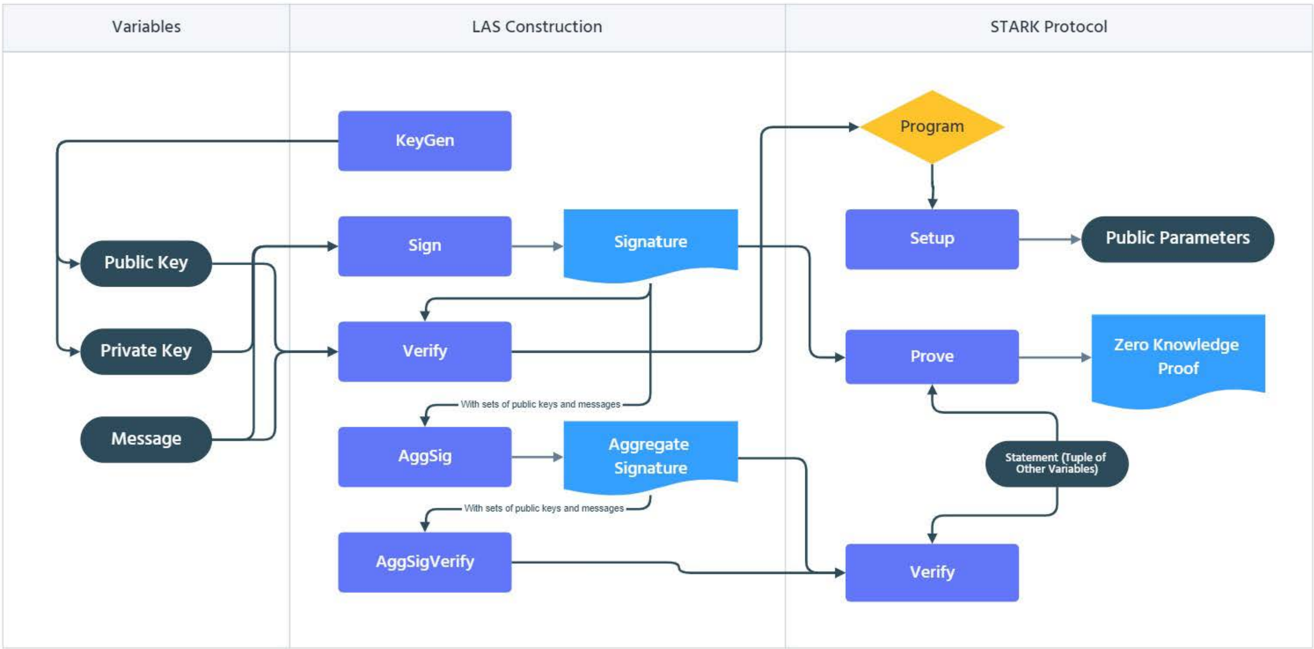


FIGURE 7. LAS scheme framework.

z, h, c individually.

$$z_{\mathrm{ag}} \le (\gamma - \beta)\sum_{i=1}^{n} v_i$$

$$\sum_{i=1}^{n} v_i \cdot z_i \le (\gamma - \beta)\sum_{i=1}^{n} v_i$$

Since each $z_i \le \gamma - \beta$ based on our construction, we get that the inequality above must hold.

Next,

$$c_{\mathrm{ag}} = \sum_{i=1}^{n} v_i \times H(w_i \| \mu_i)$$

$$\sum_{i=1}^{n} c_i \cdot v_i = \sum_{i=1}^{n} v_i \times H(w_i \| \mu_i)$$

From our construction, we know that $c_i = H(w_i \| \mu_i)$, and thus the equation above must hold true.

Finally,

$$h_{\mathrm{ag}} = \sum_{i=1}^{n} h_i \Rightarrow \|h_{\mathrm{ag}}\| = \sum_{i=1}^{n} \|h_i\| \|h_{\mathrm{ag}}\| \le n\omega^{\frac{1}{2}}$$

Since $\|h_i\| \le \sqrt{\omega}$, the equation above must hold. □

Theorem 10 (Security/Unforgeability): The aggregate signature scheme in Construction IV-B possesses SUF-CMA (Strong Unforgeability under Chosen Message Attacks) in the QROM (Quantum Random Oracle Model).

Proof: The standard evaluation method of security for digital signature schemes is the UF-CMA (Unforgeability under Chosen Message Attacks) security, and an even stronger security notion is the Strong UF-CMA (SUF-CMA). In the classical random oracle model, it can be easily shown that the LAS is SUF-CMA secure, for it is based on the hardness of the MLWE and MSIS problems. However, we also need to take into consideration the security of the LAS in the quantum random oracle model (QROM). Since we know that CRYSTALS-Dilithium and the STARK protocol are quantum secure [15], [21], we use their security proofs to prove the security of LAS.

By the SUF-CMA security of Dilithium, we can assume the SUF-CMA security of the **KeyGen**, **Sign**, and **Verify** algorithms of the LAS. By our method of generating and verifying the aggregate signature, we used the technique of preventing rogue attacks and chosen message attacks. Moreover, we used a quantum-resistant STARK protocol to generate a zero-knowledge proof. Therefore, even when a quantum adversary can query the hash function on a superposition of inputs about the aggregate signature, they would not be able to break the scheme. Thus, we get that our **AggSig** and **AggSigVerify** have SUF-CMA in QROM.

The mathematical proof of what we outlined above is as follows:

We want to show that an arbitrary adversary $\mathbb{A}$ has only a negligible advantage in breaking the LAS scheme, where the advantage of an adversary is defined as the difference between the adversary's probability of breaking the scheme and the probability that the system can be broken by guessing. The advantage of A in SUF-CMA is defined as follows

$$\begin{aligned} Adv^{\text{SUF-CMA}}(\mathbb{A}) \le\ & 2Pr[G_2 \Rightarrow 1] \\ & + Q_S \cdot 2^{-\alpha+1} + \kappa Q_S \cdot \epsilon_{\mathrm{zk}} \\ & - Pr[G_1 \Rightarrow 1] - 2^{-\epsilon+1} \end{aligned} \tag{12}$$

- Game G_1 computes the signatures on the message using a simulation algorithm *Sim*, which outputs a distribution that has statistical distance of at most ϵ_{zk}
- Game G_2 produces a boolean variable evaluated by the expression $c = H(W\|M)$
- Q_s is the number of classical queries to the signing oracle **Sign**
- κ is a positive integer constant (the value does not matter for our security proof, as it gets canceled out later on)
- $\alpha = 255$ is the bits of minimum entropy of our construction scheme
- ϵ_{zk} is the maximum statistical distance between (W, c, Z) and $(W', c', Z') \leftarrow Trans(sk)$ where *Trans* is a transcript oracle that returns a real interaction between the prover and verifier

Inequality 12 can be rewritten as follows for the proposed LAS scheme [22]:

$$Adv^{\text{SUF-CMA}}(\mathbb{A}) \le Adv^{\text{MLWE}}(\mathbb{A}) + Adv^{\text{SelfTargetMSIS}}(\mathbb{A}) + Adv^{\text{MSIS}}(D) + 2^{-\alpha+1} \qquad (13)$$

such that D is a probability distribution such that $D : Z_q \to [0, 1]$ and α is the bits of minimum entropy.

Inequality 13 also uses the MLWE and MSIS problem as well as the concept of SelfTargetMSIS, which we will explain below. For the proposed LAS scheme, we assume the hardness of the MLWE and MSIS problem, and SelfTargetMSIS is the assumption that new message forgery is based on.

Lemma 11: (MLWE Problem) Let D be a probability distribution such that $D : Z_q \to [0, 1]$, the advantage of a quantum adversary $\mathbb{A}$ solving the Module Learning with Errors problem (MLWE) problem over ring Z_q:

$$Adv^{MLWE}(\mathbb{A}) = |Pr[\mathbb{A}(A, t) \to 1] - Pr[\mathbb{A}(A, As_1 + s_2) \to 1]| \qquad (14)$$

where A is the (4,4) matrix defined in our construction, and t, s_1, s_2 are all defined just as in our construction.

Lemma 12: (MSIS Problem) Let D be a probability distribution such that $D : Z_q \to [0, 1]$, the advantage of a quantum adversary $\mathbb{A}$ solving the Module Short Integer Solution (MSIS) problem over ring Z_q:

$$Adv^{MSIS}(\mathbb{A}) = Pr[0 < \|y\|_\infty \le \lambda \wedge [I|A] \cdot y = 0|] \qquad (15)$$

where A is the (4,4) matrix defined in our construction, and $y = \mathbb{A}(A)$.

Lemma 13: (SelfTargetMSIS Problem) Let $H : 0, 1^* \to c_i$ be a cryptographic hash function, the advantage of a quantum adversary $\mathbb{A}$ to forge a new message:

$$Adv^{SelfTargetMSIS}(A) = Pr[\|y\|_\infty \le \lambda \wedge H([I|A] \cdot y\|M) = c_{agg}] \qquad (16)$$

where A is the (4,4) matrix defined in our construction, and $y = \mathbb{A}^{|H\rangle}(A)$.

In order to prove that the advantage of a quantum adversary $\mathbb{A}$ is negligible under SUF-CMA, we want the right hand side of Inequality 13 to also be negligible.

Since we assumed the hardness of MLWE and MSIS problems and we set $\alpha = 255$, we get that the right hand side can be simplified to just $Adv^{SelfTargetMSIS}(\mathbb{A})$. It suffices to prove that $Adv^{SelfTargetMSIS}(\mathbb{A})$ is negligible.

$Adv^{SelfTargetMSIS}(\mathbb{A})$ concerns the situation when an adversary receives a random (A, t) and outputs a valid pair of messages and valid signatures: $M, \sigma_{ag} : (z_{ag}, h_{ag}, c_{ag})$. This means that the following conditions must be met:

- $z_{ag} \le (\lambda - \beta) \sum_{i=1}^{n} v_i$
- $c_{ag} = \sum_{i=1}^{n} v_i \times H(w_i\|m_i)$
- $\|h_{ag}\| \le n\omega^{\frac{1}{2}}$

The second condition, in other words, requires us to show that $H(w_i\|m_i) = c_i$. we can rewrite w_i as $UseHint(h_i, Az_i - c_i t_1 \cdot 2^d, 2\lambda_2)$. By Lemma 1 in [15], we get that $w_i = Az_i - c_i t_1 \cdot 2^d + u$. Then, we know that the right hand side is equal to $Az - c_i t + u'$ where $u' = ct_0 + u$ because $t_1 = HighBit(t)$. Now, we can rewrite w_i as follows:

$$w_i = Az - c_i t + u' = [A|t|I] \cdot \begin{bmatrix} z \\ -c \\ u' \end{bmatrix}$$

Therefore, we rewrite the whole expression involving c_{ag} as follows:

$$c_{ag} = \sum_{i=1}^{n} v_i \times H(M\|[A|t|I] \cdot \begin{bmatrix} z \\ -c \\ u' \end{bmatrix}) \qquad (17)$$

By the hardness of MLWE, we get that $(A, t = As_1 + s_2)$ is indistinguishable from (A, t) where t is randomly sampled, so this is exactly what we want for $Adv^{SelfTargetMSIS}$ by Lemma 16.

However, note that since we are proving SUF-CMA and not just UF-CMA, we have to consider the case where the adversary $\mathbb{A}$ sees a signature (z_{ag}, h_{ag}, c_{ag}) for M and then only changes (z_{ag}, h_{ag}). In this case, we have $w_i' = UseHint(h_i', Az_i' - c_i t_1 \cdot 2^d, 2\lambda_2)$. Notice that we can apply the same calculations to get that $Adv^{SelfTargetMSIS}(\mathbb{A})$ is negligible.

Thus, we can claim that the proposed LAS scheme construction has SUF-CMA in QROM.

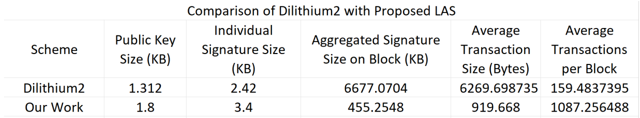
□

V. IMPLEMENTATION

A. SETTING THE PARAMETERS

There are a few parameters in the proposed LAS scheme that we need to consider when implementing the scheme.

- For the Ring $\mathbb{Z}_q[X]/(X^n + 1)$ that we perform operations in, we set $q = 2^{23} - 2^{13} + 1$ and $n = 256$ in order to ensure that Theorem 9, 8, and 10 always hold.
- In order to make LAS quantum-resistant, we set the size of matrix A to always be $(4, 4)$. This gives sufficient security in a classical random oracle model and in QROM.



Comparison of Dilithium2 with Proposed LAS

Scheme	Public Key Size (KB)	Individual Signature Size (KB)	Aggregated Signature Size on Block (KB)	Average Transaction Size (Bytes)	Average Transactions per Block
Dilithium2	1.312	2.42	6677.0704	6269.698735	159.4837395
Our Work	1.8	3.4	455.2548	919.668	1087.256488

FIGURE 8. Comparison of Dilithium2 and the proposed LAS scheme.

- The following constants are always the same in order that the construction of LAS be reasonable under all circumstances:

$$d = 14, \quad \lambda_1 = \frac{q-1}{16} = 523776, \ \lambda_2 = \frac{\lambda_1}{2} = 261888$$

- The variables that may be varied for different implementations are β, β_1, ω, η. For implementation and security purposes, we set them as follows:

$$\beta = 175, \quad \beta_1 = 250, \ \omega = 120, \ \eta = 2$$

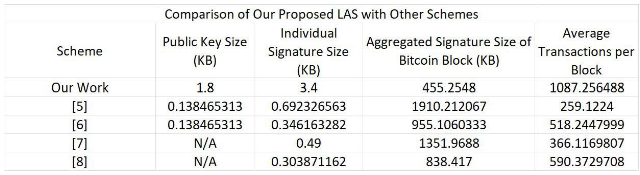
B. RESULTS

With the parameters that we set, we implemented the proposed scheme in Python and acquired a MIT license on Github [23]. We obtained the following results:

The proposed scheme was shown to only decrease the Bitcoin transaction efficiency (average transactions per block) by 3 times from ECDSA while Dilithium decreased its transaction efficiency by 17 times. Therefore, our scheme will become very useful when quantum computers are popularized, for it is not only post-quantum secure but also increases Bitcoin's transaction efficiency from CRYSTALS-Dilithium.

We also compared our proposed scheme with other existing lattice-based aggregate signature schemes mentioned in Section I-B using the same parameters.

As seen in Figure 9, although the individual signature size of our scheme is larger, our aggregate signatures have a smaller size than other schemes because of our aggregation method. Thus, our scheme has a higher efficiency - Bitcoin blocks using our scheme can contain more transactions per block - so our proposed scheme has great potential to be adopted by post-quantum Bitcoin.



Comparison of Our Proposed LAS with Other Schemes

Scheme	Public Key Size (KB)	Individual Signature Size (KB)	Aggregated Signature Size of Bitcoin Block (KB)	Average Transactions per Block
Our Work	1.8	3.4	455.2548	1087.256488
[5]	0.138465313	0.692326563	1910.212067	259.1224
[6]	0.138465313	0.346163282	955.1060333	518.2447999
[7]	N/A	0.49	1351.9688	366.1169807
[8]	N/A	0.303871162	838.417	590.3729708

FIGURE 9. Comparison of our proposed LAS scheme with other schemes.

C. ANALYSIS: STRENGTHS AND SHORTCOMINGS

The proposed LAS scheme has several advantages:

- Small aggregate signature size: through our LAS scheme, we generated aggregate signatures for Bitcoin blocks such that the block size is smaller than that of blocks using Dilithium.
- Useful for Bitcoin in post-quantum setting: since the LAS scheme preserved quantum security against adversaries, this scheme is very beneficial to post-quantum Bitcoin.
- Easy implementation and practicality: NTT offers efficient implementation in constant time.
- Our proposed scheme generates a zero-knowledge proof along with each aggregate signature, which protects users' privacy
- Our scheme is unordered and does not require each user to verify the validity of the signature of the user before them

Overall, our proposed scheme is very efficient and possesses several advantages over the other prior schemes in literature - it's also a significant improvement from the primary algorithm for post-quantum signature schemes selected by NIST. Therefore, our scheme present itself as important to Bitcoin in a post-quantum era.

On the other hand, there are also a few areas of improvement for the LAS scheme:

- Despite that we reduced the aggregate signature size of a Bitcoin block, the scheme can be further improved so that it decrease the individual signature size even more.
- Future work can also be done on our construction of the aggregate signature and the verification process of the aggregate signature to find an even more efficient process of generating and verifying the aggregate signature.
- Another area of research is to extend our signature scheme to other cryptocurrencies such as Ethereum and understand what modifications are necessary.

VI. CONCLUSION

This study proposes a novel lattice-based aggregate signature scheme that can increase Bitcoin's transaction efficiency in a post-quantum setting. We calculated that the Bitcoin transaction efficiency would fall by 17 times when using CRYSTALS Dilithium, the primary post-quantum signature scheme, compared to using ECDSA. Thus, we decided to construct a new aggregate signature scheme based on CRYSTALS-Dilithium and a STARK protocol so that our scheme possesses the benefits of both the algorithm and the protocol. We implemented our scheme and found that with our scheme, Bitcoin's transaction efficiency would merely decrease by 3 times from using ECDSA. Our scheme could significantly improve Bitcoin's transaction efficiency in a post-quantum world, and it also presents many other benefits such as easy implementation, protection of privacy, practicality, and strong security.

REFERENCES

[1] Computer Security Resource Center of NIST. (Jan. 2017). *Selected Algorithms 2022—Post-Quantum Cryptography*. [Online]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022

[2] Z. Yanhua, H. Yupu, J. Mingming, and X. Lili, "Lattice-based sequential aggregate signatures with lazy verification," *J. China Univ. Posts Telecommun.*, vol. 22, no. 6, pp. 36–44, Dec. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1005888515606914

[3] Z. Wang and Q. Wu, "A practical lattice-based sequential aggregate signature," in *Proc. Int. Conf. Provable Secur.* Cham, Switzerland: Springer, Oct. 2019, pp. 94–109. [Online]. Available: https://dl.acm.org/doi/abs/10.1007/978-3-030-31919-9_6

[4] R. E. Bansarkhani and J. Buchmann, "Towards lattice based aggregate signatures," in *Proc. Int. Conf. Cryptol. Afr.* Cham, Switzerland: Springer, May 2014, pp. 336–355. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-06734-6_21

[5] P. Zhang, J. Yu, and T. Wang, "A homomorphic aggregate signature scheme based on lattice," *Chin. J. Electron.*, vol. 21, no. 4, pp. 701–704, 2012.

[6] Z. Jing, "An efficient homomorphic aggregate signature scheme based on lattice," *Math. Problems Eng.*, vol. 2014, pp. 1–9, Nov. 2014. [Online]. Available: https://www.hindawi.com/journals/mpe/2014/536527/

[7] R. E. Bansarkhani and J. Sturm, "An efficient lattice-based multisignature scheme with applications to Bitcoins," in *Proc. Int. Conf. Cryptol. Netw. Secur.* Cham, Switzerland: Springer, Nov. 2016, pp. 140–155. [Online]. Available: https://www.semanticscholar.org/paper/An-Efficient-Lattice-Based-Multisignature-Scheme-to-Bansarkhani-Sturm/f4f72663a768a2d1a196b8f222804e6cb8098f78

[8] X. Lu, W. Yin, Q. Wen, Z. Jin, and W. Li, "A lattice-based unordered aggregate signature scheme based on the intersection method," *IEEE Access*, vol. 6, pp. 33986–33994, 2018. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8386429

[9] Z. Bao, W. Shi, S. Kumari, Z.-Y. Kong, and C.-M. Chen, "Lockmix: A secure and privacy-preserving mix service for bitcoin anonymity," *Int. J. Inf. Secur.*, vol. 19, no. 3, pp. 311–321, Jun. 2020. [Online]. Available: https://link.springer.com/article/10.1007/s10207-019-00459-6

[10] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer, "From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again," in *Proc. 3rd Innov. Theor. Comput. Sci. Conf.*, 2012, pp. 326–349. [Online]. Available: https://dl.acm.org/doi/10.1145/2090236.2090263

[11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Bus. Rev.*, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[12] The Guardian. (Jun. 2022). *Bitcoin Block Size Limit—What is it?* [Online]. Available: https://guardian.ng/saturday-magazine/brand-intelligence/bitcoin-block-size-limit-what-is-it/

[13] D. Boneh and S. Kim, "One-time and interactive aggregate signatures from lattices," Stanford, CA, USA, Tech. Rep., 2020. [Online]. Available: https://crypto.stanford.edu/~skim13/agg_ots.pdf

[14] D. Boneh and V. Shoup, *A Graduate Course in Applied Cryptography*. Philadelphia, PA, USA: Univ. of Pennsylvania, Sep. 2017. [Online]. Available: https://www.cis.upenn.edu/~sga001/classes/cis331f19/resources/bs_ch2.4.pdf

[15] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-dilithium: A lattice-based digital signature scheme," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2018, no. 1, pp. 238–268, Feb. 2018, doi: 10.13154/tches.v2018.i1.238-268.

[16] D. Soni, K. Basu, M. Nabeel, N. Aaraj, M. Manzano, and R. Karri, "FALCON," in *Hardware Architectures for Post-Quantum Digital Signature Schemes*. Cham, Switzerland: Springer, 2021, pp. 31–41, doi: 10.1007/978-3-030-57682-0_3.

[17] E. Ben-Sasson, I. H. Y. Bentov, and M. Riabzev, "Scalable, transparent, and post-quantum secure computational integrity," *Cryptol. ePrint Arch.*, Mar. 2018. [Online]. Available: https://eprint.iacr.org/2018/046.pdf

[18] E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward, "Aurora: Transparent succinct arguments for R1CS," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, May 2019, pp. 103–128. [Online]. Available: https://eprint.iacr.org/2018/828.pdf

[19] M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. Sebastopol, CA, USA: O'Reilly Media, 2014.

[20] (2022). *QuantaBytes: A Survey of Bitcoin Transaction Types*. [Online]. Available: https://www.quantabytes.com/articles/a-survey-of-bitcoin-transaction-types

[21] I. Khaburzaniya, K. Chalkias, K. Lewi, and H. Malvai, "Aggregating and thresholdizing hash-based signatures using STARKs," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, May 2022, pp. 393–407, doi: 10.1145/3488932.3524128.

[22] E. Kiltz, V. Lyubashevsky, and C. Schaffner, "A concrete treatment of Fiat–Shamir signatures in the quantum random-oracle model," presented at the Annu. Int. Conf. Theory Appl. Cryptograph. Techn. Cham, Switzerland: Springer, Apr. 2018, pp. 552–586, doi: 10.1007/978-3-319-78372-7_18.

[23] Y. Quan, "A new lattice-based aggregate signature scheme based on crystals-dilithium and a stark protocol," GitHub Repository, Tech. Rep., 2022. [Online]. Available: https://github.com/angelinaquan/A-New-Lattice-Based-Aggregate-Signature-Scheme-Based-on-CRYSTALS-Dilithium-and-a-STARK-Protocol

YUNJIA QUAN is with the Charlotte Country Day School. She is an United States of America Junior Math Olympiad Qualifier and is an International Science and Engineering Fair Finalist. Her research interests include cryptography, topology, and biology. She received the Outstanding Award for First Place from the International Mathematical Modeling Challenge.