Post-Quantum Hybrid Security Mechanism for MIMO Systems

Yousef Qassim, Mario Edgardo Magaña, and Attila Yavuz
Oregon State University
Email: Yousef.Qassim, Mario.E.Magana, Attila.Yavuz@oregonstate.edu

***Abstract*— In this paper, we propose a post-quantum cross-layer key agreement scheme that is robust against Man in the Middle (MitM) attack and the wide deployment of quantum computers. Our security mechanism combines physical layer and cryptographic security techniques to provide best effort security. Physical layer security usually has no assumption on the eavesdropper's, Eve, computational power, nor on Eve's available information. It is unbreakable, provable, and quantifiable. However, physical layer security is limited, hard to prove, and researchers usually consider a passive attacker model. Alternatively, traditional cryptography has worked well in practice, but it is based on the assumption of limited computational power at Eve and it is vulnerable to the large-scale implementation of quantum computers.**

***Index Terms*— wireless, security, physical layer, cryptography, public-key, private-key, MIMO, MitM attack, key exchange, SPHINCS, post-quantum.**

I. Introduction

In the past decade, the world has become gradually connected and the introduction of Internet of Things became a widely used notion in research. While the advancement of technology was able to put a radio access interface on every device and provided reliable communication links, information security took the back seat. Generally, the wireless communication medium security has always been a critical issue since an unprecedented amount of sensitive and private data being transmitted over it. In conventional wireless networks, security issues are primarily handled by the higher-level layer, i.e. application layer, and rely on the computational complexity of an underlying mathematical problem known as cryptographic methods. While they have worked well in practice [1], [2], they might be difficult to implement and may be vulnerable to attacks in some cases since they require a secure channel to exchange keys or certificate management. Most importantly, the majority of public-key cryptosystems are susceptible to large deployment of quantum computers. Current methods rely either on integer factorization, discrete logarithmic, or elliptic curve discrete logarithmic problems which can be solved easily using Shor's algorithm [3].

On the other hand, physical layer security techniques exploit the characteristics of the wireless channel to improve security. It ensures data's security by requiring the latter to be a design constraint rather than a feature. By utilizing physical layer security methods, it becomes more difficult for attackers to decipher transmitted data and more robust to the increase in the adversary computational power. Moreover, physical layer security offers built-in security that is information theoretically unbreakable [4], [5]. Thus, physical layer security is not susceptible to the introduction of quantum computers. The security solutions at the physical layer can complement the cryptographic mechanisms, or work as a standalone solution for a system with strict energy requirements like the ones found in sensor networks. Although promising, physical layer security relies on assumptions about relative quality of channels. When these qualities are partially known or unknown, special handling is required [6]. Furthermore, its perfect secrecy is conditioned on the notion that channels are unknown or noisier at the adversary, which might not be true in all cases [4]. Finally, proving the security guarantee for physical layer is usually a hard task, especially for strong secrecy cases [7].

TABLE I: Security comparison between MOPRO, Diffie-Hellman + RSA, and the proposed C-MOPRO.

Algorithm	MitM Safe	Info Theoretic	Quantum Resistent	Eve Coverage	Security Loss*
MOPRO	✗	✓	✓	one	50%
DH + RSA	✓	✗	✗	none	0%
C-MOPRO	✓	✓	✓	two	0%

* with the presence of an eavesdropper near Alice or Bob.

In general, researchers focus on investigating either traditional cryptography or physical layer security and their applications. Nevertheless, there has been little to no effort in investigating a cross-layer security mechanism that combines the advantages of both directions and reduce or eliminate the disadvantages of the two schools of security. Therefore, we propose a post-quantum hybrid key agreement with device authentication security mechanism that uses a combination of physical layer security and cryptographic techniques to achieve a powerful security mechanism with reasonable overhead.

Our proposed algorithm (C-MOPRO) is based on the work presented in [8]. The authors in [8] achieved key agreement during channel establishment phase using physical layer techniques. This resulted in less communicational and computational overhead. In addition, they achieved this with a reasonable security guarantee, therefore we adopted their solution. However, our work significantly differs from their work in the following aspects: Firstly, our proposed solution assumes an active attacker model while they assume a passive model. In the active attacker model, the adversary can do more than just eavesdropping on the communication between

two legitimate users. In fact, the adversary can launch a Man in the Middle (MitM) attack where he/she can impersonate one or more of the legitimate users, or/and jam their communications. In this paper we only consider the MitM attack. In order to prevent such an attack, we implement a digital signature scheme to authenticate the legitimate users. Specifically, we choose to implement SPHINCS which is a stateless hash-based signature scheme. SPHINCS depends only on the existence of secure hash functions which makes it very adjustable and invulnerable to quantum computing [9]. Secondly, we address the issue where one of the legitimate users' keys gets jeopardized resulting in unveiling half of communicated messages to an eavesdropper. The security comparison of our proposed algorithm against other techniques is summarized in Table I.

The mechanism proposed in [8] uses complex signals to achieve key agreement between the legitimate users. Meanwhile, SPHINCS uses real value messages to authenticate users. Consequently, the main challenge here is how to sign the complex signals using SPHINCS to authenticate the users.

The rest of this paper is organized as follows: In Section II, we introduce SPHINCS digital signature and its components. Section III presents the system model and discusses MIMO precoding. In Section IV, we discuss our proposed algorithm. Then, we detail the security analysis of our proposed scheme in Section V. After that, we examine the performance of the algorithm and provide a comparison to its counterparts in Section VI. Finally, Section VII concludes the paper and states our future work.

II. Cryptographic Primitives

In light of the wide introduction of quantum computers and its consequences on modern digital signatures, current post-quantum cryptography research proposes SPHINCS as one of the best alternatives. As stated before, SPHINCS is a stateless hash-based signature scheme. In fact, one-time signature (OTS) as its basic block as in all hash-based signatures. Merkle adopted this scheme in order to construct a many-time signature scheme [10]. When a Merkle tree is used on top of OTS key pairs, the choice of an OTS key more than once should be avoided. This requires us to store some info, i.e. state, about the keys that have already been used making it impractical in some cases. To overcome this problem, Goldreich proposed a scheme that creates a tree in a way that makes the probability of choosing a previously used key significantly small [11]. However, the size of Goldreich's signature is extremely large.

SPHINCS overcomes both challenges; the state and signature size. It does that by combining Goldreich's scheme with Merkle trees and few-time signatures. The authors use Winternitz One-Time Signature (WOTS+)[1] scheme to form the Merkle tree [12]. Also, they propose HORST few-time signature scheme, which is basically a version of HORS [13] with trees, to sign the message digest. Both schemes are defined in Algorithm 1 and Algorithm 2, respectively.

[1]The authors of [9] slightly deviated from description of WOTS+ in [12].

Global parameters: Winternitz parameter $w \in \mathbb{N}$, $w > 1$, message M, security parameter $n \in \mathbb{N}$, input seed $S \in \{0,1\}^n$, $l_1 = \lceil n/log(w) \rceil$, $l = l_1 + \lfloor log(l_1(w-1))/log(w) \rfloor + 1$, $G_\lambda : \{0,1\}^n \rightarrow \{0,1\}^{\lambda n}$, $\mathcal{V} : \{0,1\}^n \rightarrow \{0,1\}^n$.

Algorithm 1 WOTS+ Signature

1: **Parameters:** $|M| = n$, bitmasks $\mathbf{r} \in \{0,1\}^{n*(w-1)}$, $c^i(x, \mathbf{r}) = \mathcal{V}(c^{i-1}(x, \mathbf{r}) \oplus r_i)$
2: **Key Generation** $(SK, PK) \leftarrow WOTS.kg(S, \mathbf{r})$: Outputs secret key SK and public key PK
- $SK = (SK_1, .., SK_l) \leftarrow G_l(S)$
- $PK = (PK_1, .., PK_l) = (c^{w-1}(SK_1, \mathbf{r}), .., c^{w-1}(SK_l, \mathbf{r}))$

3: **Signing** $\sigma_{WOTS} \leftarrow WOTS.sign(M, S, \mathbf{r})$: Outputs signature σ_{WOTS} for M under SK
- SK and PK are generated on the fly since storage(S) < storage(SK)

4: **Verifying** $PK' \leftarrow WOTS.vf(M, \sigma_{WOTS}, \mathbf{r})$: Outputs PK' that will be compared to PK in SPHINCS algorithm (returns true on equality, and false otherwise)

SPHINCS deploys a hyper-tree of height h that contains d layers of trees of height h/d [9]. In more details, each layer i has $2^{(d-1-i)(h/d)}$ trees. WOTS+ key pairs of the trees on layer $i+1$ are used to sign the roots of layer i trees. The WOTS+ key pair on layer 0 is used to sign a HORST public key. Finally, each HORST key pair is used to sign the message digest. It is worth noting that a pseudo-randomly generated index is used to choose which trees inside the hyper-tree are used and which HORST key pair is selected. Finally, in order to verify, a Merkle tree authentication path is provided as part of the signature. SPHINCS is described in Algorithm 3. For more details, readers are referred to [9].

III. System Model

In this paper, we consider a multi-input and multi-output (MIMO) wireless communication system. MIMO systems use multiple antennas at the transmitter and receiver ends to increase its capacity. They are widely deployed in multiple communication system technologies such as Wi-Fi, 3G, and 4G. MIMO systems are usually utilized through precoding, spatial multiplexing, and diversity coding. However, in this work, we only consider precoding which is explained later in this section.

A. System Setup

The system consists of two legitimate users (Alice and Bob) and an eavesdropper (Eve). The users are connected using wireless MIMO channels $\mathbf{H}_{AB}$, $\mathbf{H}_{AE}$, and $\mathbf{H}_{BE}$. This model is depicted in Fig. 1. Alice wants to communicate with Bob confidentially through $\mathbf{H}_{AB}$. Due to the broadcast nature of wireless channels, Eve can listen to the messages originated at Alice and Bob through $\mathbf{H}_{AE}$ and $\mathbf{H}_{BE}$, respectively. It is assumed that the MIMO system uses time

Algorithm 2 HORST Signature

1: **Parameters:** message length m, $t = 2^\tau$ where $\tau \in \mathbb{N}$, $k \in \mathbb{N}$ where $k\tau = m$, bitmasks $\mathbf{Q} \in \{0,1\}^{2n*logt}$, $x \in \mathbb{N} \setminus \{0\}$

2: **Key Generation** $PK \leftarrow HORST.kg(S, \mathbf{Q})$: Outputs public key PK
- $SK = (SK_1, .., SK_t) \leftarrow G_t(S)$
- A tree is constructed using $\mathbf{Q}$ where tree leaves $L_i = \mathcal{V}(SK_i)$ for $i \in [t-1]$
- PK = root node of a binary tree of height $log(t)$

3: **Signing** $(\sigma_{HORST}, PK) \leftarrow HORST.sign(M, S, \mathbf{Q})$: Outputs PK and signature σ_{HORST} for M under SK
- $SK = (SK_1, .., SK_t) \leftarrow G_t(S)$
- $M = (M_0, .., M_{k-1})$ where $|M_i| = log_2(t)$ bits for $i \in [k-1]$
- Determine x such that $k(\tau - x + 1) + 2^x$ is minimal
- σ_{HORST_i} = $(SK_{M_i}, Auth_{M_i})$ where $Auth_{M_i}$ is the lower $\tau - x$ elements of the authentication path of leaves $(A_0, .., A_{\tau-1-x})$ for $i \in [k-1]$
- σ_{HORST_k} = 2^x nodes of level $\tau - x$ binary tree

4: **Verifying** $PK' \leftarrow HORST.vf(M, \sigma_{HORST}, \mathbf{Q})$: The signature is valid if all nodes and authentication paths agree on the same root PK (i.e. $PK' = PK$)

Algorithm 3 SPHINCS Signature

1: **Parameters:** $p = max\{w - 1, 2(h + \lceil log(l) \rceil, 2log(t)\}$, $\mathbf{Q} \xleftarrow{\$} \{0,1\}^{pxn}$

2: **Key Generation** $(SK, PK) \leftarrow SPHINCS.kg(1^n)$: Outputs secret key SK and public key PK
- $SK = (SK_1, SK_2, \mathbf{Q})$ where $(SK_1, SK_2) \in \{0,1\}^n$ x $\{0,1\}^n$
- $PK = (PK_1, \mathbf{Q})$ where PK_1 = root node of a binary tree that is built on public keys of WOTS+ key pairs

3: **Signing** $\sigma_{SPHINCS} \leftarrow SPHINCS.sign(M, SK)$: Outputs signature $\sigma_{SPHINCS}$ for M under SK
- $\sigma_{SPHINCS} = (I, \sigma_{HORST}, Auth_i, \sigma_{WOTS_i})$ where I is index, σ_{WOTS_i} is WOTS+ signature per layer i, and $Auth_i$ is the authentication path per layer i

4: **Verifying** $ind \leftarrow SPHINCS.vf(M, \sigma_{SPHINCS}, PK)$: Returns true if the verification algorithm reaches to the same root node in PK_1, otherwise it returns false

division duplexing and the MIMO channel reciprocity holds in the transposed form $\mathbf{H}_{AB} = \mathbf{H}_{BA}^T$, where $[.]^T$ is the matrix transpose, along with perfect channel reciprocity. Alice, Bob, and Eve are equipped with M_A, M_B, and M_E number of antennas, respectively.

As in [8], the universal codebook containing precoding matrices and the corresponding precoding matrix indices (PMIs) is accessible to all parties Alice, Bob, and Eve. The channel capacity function used by Alice and Bob is also known to Eve. The mapping between precoding matrix and

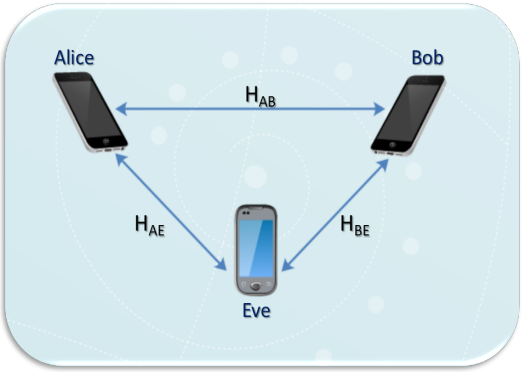

Fig. 1: System layout.

secret key sequence is a predefined public information. All parties have knowledge of this mapping in advance. Eve is assumed to be an active attacker who will falsify public discussion and/or listen to the communications between Alice and Bob but will not jam the channel.

B. MIMO Precoding

MIMO precoding is a processing technique which functions as a multi-mode beamformer to support multi-stream data transmission. By allocating appropriate transmission power to data streams, it maximizes the channel throughput. In order to achieve the optimal MIMO channel capacity, the optimal precoding matrix requires full channel state information at the transmitter (CSIT). Assuming slow frequency non-selective fading, the received signal is described by $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$, where $\mathbf{y}$ is the received signal vector, $\mathbf{H}$ is the MIMO channel matrix, $\mathbf{x}$ is the transmitted signal vector, and $\mathbf{v}$ is the white Gaussian noise vector. To obtain the optimal gain, the MIMO channel matrix $\mathbf{H}$ can be decomposed by performing the singular value decomposition (SVD) of the channel matrix as $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$, where $[.]^H$ is the Hermitian operator, $\mathbf{U}, \mathbf{V}$ are complex unitary matrices and $\mathbf{\Sigma}$ is a matrix whose diagonal elements are the singular values of $\mathbf{H}$. The optimal beam directions with perfect CSIT are matched to the channel right singular vectors $\mathbf{V}$. As a consequence, this requires the channel to be approximately constant over a considerably large period as well as a large feedback overhead. Alternatively, WiMAX and LTE systems use a codebook that consists of multiple precoding matrices and their corresponding PMIs, which yields a balance between system performance, equalizer complexity, and the feedback overhead.

The MIMO-OFDM channel matrix $\mathbf{H}$ is estimated at the receiver using the pilot symbols sent by the transmitter. Then, the suboptimal precoding matrix that maximizes the channel capacity is selected by the receiver using the following equation:

$$\max_{\mathbf{F} \in \mathcal{F}} Capacity_{\mathbf{H},\mathbf{F}} = \log_2 \det[\mathbf{I}_n + \frac{E_s}{n_s \sigma^2} \mathbf{F}^H \mathbf{H}^H \mathbf{H} \mathbf{F}] \quad (1)$$

where $\mathbf{F}$ is the precoding matrix, $\mathcal{F}$ is the universal codebook, $\mathbf{I}_n$ is the identity matrix and n is the minimum number

TABLE II: C-MOPRO Notations.

$\mathbf{G}$	M_AxM_A random unitary complex matrix
$\mathbf{r}$	M_AxN_r complex reference signal
$\mathbf{U}_{B,i}$	M_BxM_B complex unitary matrix
$\mathbf{V}^H_{B,i}$	M_AxM_A complex unitary matrix
$\mathbf{V}_{A,i}$	M_AxM_A complex unitary matrix
$\mathbf{U}^H_{A,i}$	M_BxM_B complex unitary matrix
$\breve{\mathbf{F}}$	M_Bxn_s
$\mathbf{G}_u$	n_sxn_s complex unitary matrix
$\mathbf{s}$	n_sxN_s complex matrix
(SK_B, PK_B)	Bob's secret and public keys
(SK_A, PK_A)	Alice's secret and public keys

of antennas at Alice and Bob, E_s is the total energy of the transmitted signal, n_s is the number of data elements, and σ^2 is the noise variance. Finally, the receiver sends the corresponding PMI of the suboptimal precoding matrix to the transmitter.

IV. Proposed Algorithm

In this section, the proposed algorithm C-MOPRO is detailed. Our proposed solution is based on the MOPRO scheme presented in [8]. The algorithm utilizes complex unitary rotation matrices to hide the secrecy information and exchange secret keys during the communication establishment phase. Although similar, our work differs in the following: 1) It assumes an active attacker model. 2) It addresses the Man in the Middle (MitM) attack. 3) It addresses the issue of exposing half of the secret key. Fig. 2 depicts the exchanged messages between the legitimate users and what is heard by Eve. The flow of our algorithm is detailed next and the notation used in the algorithm is defined in TABLE II.

1) Alice transmits the reference signal $\mathbf{Gr}$ to Bob to estimate the channel. Bob estimates the sub-band i averaged channel $\mathbf{H}_{AB,i}\mathbf{G}_i$ and performs SVD on $\mathbf{H}_{AB,i}\mathbf{G}_i$ to obtain $\mathbf{H}_{AB,i}\mathbf{G}_i = \mathbf{U}_{B,i}\mathbf{\Sigma}_i\mathbf{V}^H_{B,i}\mathbf{G}_i$, where $\mathbf{\Sigma}_i$ is M_BxM_A matrix.
2) Bob generates a secret key $\mathcal{K}_{Bob}$ of c-bits. Bob applies channel coding and obtains the coded sequence $\mathcal{C}_{Bob}$. Based on the codebook used, Bob divides $\mathcal{C}_{Bob}$ into $\lceil \frac{c}{p} \rceil$ groups each denoted $\mathcal{C}_{Bob,i}$.
3) Using $\mathcal{C}_{Bob,i}$ as PMI, Bob finds the corresponding precoding matrix $\mathbf{F}_{B,i}$. Bob appends random orthogonal columns to $\mathbf{F}_{B,i}$ to make it a full rank M_BxM_B complex unitary matrix $\hat{\mathbf{F}}_{B,i}$.
4) Bob transmits the rotated reference signal $\mathbf{G}_{1,i}\mathbf{r}$ to Alice, where $\mathbf{G}_{1,i} = \mathbf{U}^*_{B,i}\hat{\mathbf{F}}^H_{B,i}$ and $[.]^*$ is the matrix conjugation. Then, Alice estimates PMI of the ith sub-band from $\mathbf{H}_{BA,i}\mathbf{G}_{1,i}$.
5) Bob generates $(SK_B, PK_B) \leftarrow SPHINCS.kg(1^n)$. Bob transmits $[SPHNCS.sign(\mathbf{G}_{1,i}\mathbf{r}, SK_B), PK_B]$ to Alice. Then, Alice verifies Bob on the ith sub-band using $SPHINCS.vf(\mathbf{G}_{1,i}\mathbf{r}, \sigma_{SPHINCS}, PK_B)$.
6) Steps 3-5 are repeated for all sub-bands. Alice combines all the collected PMIs to form $\mathcal{C}_{Bob}$ and then obtain $\mathcal{K}_{Bob}$. Alice generates a secret key $\mathcal{K}_{Alice}$ of c-bits.
7) Alice applies channel coding and obtains the coded sequence $\mathcal{C}_{Alice}$ and divides $\mathcal{C}_{Alice}$ into $\lceil \frac{c}{p} \rceil$ groups each denoted $\mathcal{C}_{Alice,i}$. Using $\mathcal{C}_{Alice,i}$ as PMI, Alice finds the corresponding precoding matrix $\mathbf{F}_{A,i}$. Bob appends random orthogonal columns to $\mathbf{F}_{A,i}$ to make it a full rank M_AxM_A complex unitary matrix $\hat{\mathbf{F}}_{A,i}$.
8) Alice performs SVD on $\mathbf{H}_{BA,i}\mathbf{G}_{1,i}$ to obtain $\mathbf{H}^T_{AB,i}\mathbf{G}_{1,i} = \mathbf{V}^*_{A,i}\mathbf{\Sigma}^T_i\mathbf{U}^T_{A,i}\mathbf{G}_{1,i}$, where $\mathbf{\Sigma}_i$ is M_AxM_B diagonal matrix. Alice transmits the rotated reference signal $\mathbf{G}_{2,i}\mathbf{r}$ to Bob, where $\mathbf{G}_{2,i} = \mathbf{V}_{A,i}\hat{\mathbf{F}}^H_{A,i}$. Bob estimates PMI of the ith sub-band from $\mathbf{H}_{AB,i}\mathbf{G}_{2,i}$.
9) Alice generates $(SK_A, PK_A) \leftarrow SPHINCS.kg(1^n)$ and sends $[SPHNCS.sign(\mathbf{G}_{2,i}\mathbf{r}, SK_A), PK_A]$. Then, Bob verifies Alice on the ith sub-band using $SPHINCS.vf(\mathbf{G}_{2,i}\mathbf{r}, \sigma_{SPHINCS}, PK_A)$.
10) The steps are repeated for all sub-bands. Alice combines all the collected PMIs to form $\mathcal{C}_{Alice}$ and then obtain $\mathcal{K}_{Alice}$.
11) Alice and Bob apply a cryptographic hash function on the concatenation of Alice and Bob keys. A shared secure key is defined by $\mathcal{K}_{AB} = H(\mathcal{K}_{Alice}||\mathcal{K}_{Bob})$.
12) Alice finds the optimal precoding matrix to achieve MIMO channel capacity using: $\breve{\mathbf{F}} = \max_{\mathbf{F}\in\mathcal{F}} Capacity_{\mathbf{H},\mathbf{F}} = \log_2 \det[\mathbf{I}_n + \frac{E_s}{n_s\sigma^2}\mathbf{F}^H\mathbf{H}^H\mathbf{H}\mathbf{F}]$.

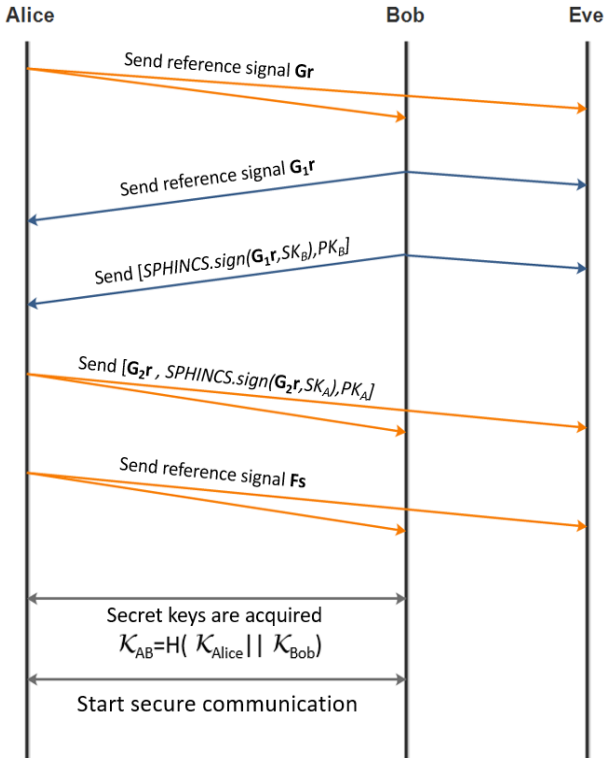

Fig. 2: C-MOPRO message exchange between Alice and Bob.

TABLE III: Overhead comparison between MOPRO, Diffie-Hellman + RSA, and the proposed C-MOPRO.

	Alice Overhead		Bob Overhead	
Algorithm	**Computation**	**Communication (in bits)**	**Computation**	**Communication (in bits)**
MOPRO	-	$n_b\|\mathbf{Gr}\|$	-	-
DH + RSA	KA: $1.Exp$ SGN: $2.Exp' + 2.Hash$	$\|p\|$ + $\|g\|$ + $\|A\|$ + $\|PK_{RSA_A}\|$ + $\|\sigma_{RSA}\|$	KA: $1.Exp$ SGN: $2.Exp' + 2.Hash$	$\|B\|$ + $\|PK_{RSA_B}\|$ + $\|\sigma_{RSA}\|$
C-MOPRO	KA: $1.Hash$ SGN: n_bC	$\|PK_{SPH_A}\|$ + $n_b\|\mathbf{Gr}\|$ + $n_b\|\sigma_{SPH}\|$	KA: $1.Hash$ SGN: n_bC	$\|PK_{SPH_B}\|$ + $n_b\|\sigma_{SPH}\|$

KA: Key agreement algorithm. SGN: Digital signature algorithm. Exp: Module exponentiation in DH. Exp': Module exponentiation in RSA. $Hash$: Hash function. p, g, A, B: Parameters for Diffie-Hellman key exchange algorithm, where $|p|$=$|g|$=3072 bits. n_b: Number of sub-bands. PK_{RSA}: RSA public key. PK_{SPH}: SPHINCS public key. σ_{RSA}: RSA signature. σ_{SPH}: SPHINCS signature. C: Cost of SPHINCS-256 signature which consists of 699494 ChaCha12 permutations [9]. For 128-bit post-quantum security: 3072-bit DH, 3072-bit RSA, SPHINCS-256, and SHA-384 are considered [14].

Alice generates $\mathbf{G}_u$ and creates $\mathbf{F} = \breve{\mathbf{F}}\mathbf{G}_u$. Alice transmits the reference signal $\mathbf{Fs}$ and Bob estimates the channel $\mathbf{H}_{AB}\mathbf{F}$.

V. Security Analysis

In this section, we discuss the security guarantee of the proposed solution. The security guarantee of C-MOPRO, DH + RSA, and MOPRO is summarized in Table I. By deploying the physical layer security mechanism, exchanging uniformly distributed secrets keys is made possible during the channel establishment phase. Furthermore, the use of the unitary rotation matrices prevents Eve from acquiring either $\mathbf{H}_{AE}$ or $\mathbf{H}_{BE}$ since only the rotated channel is used to exchange messages. This renders Eve attempts to reconstruct the complete channel between Alice and Bob useless and provides additional security to the communication channel. However, based on Eve's location there might be a risk of exposing half of the secret key bits. If Eve places itself near either Alice or Bob, then the channel experienced by Eve will be close to either one of the legitimate users. For example, if Eve placed itself close to Bob then $\mathbf{H}_{AE}\mathbf{G}_2 \simeq \mathbf{H}_{AB}\mathbf{G}_2$ and by performing PMI estimation Eve can obtain $\mathcal{K}_{Alice}$.

For physical layer security mechanism to be information theoretically unbreakable, it has to satisfy the strong secrecy condition defined as $\lim_{n\to\infty} I(W|Z^n) = 0$. This requires that the mutual information between each bit of the message W and the observed n-length cipher Z^n at Eve to be zero, i.e. no information leakage about the message when the transmitted cipher is observed by Eve [15]. To remedy this, we propose that both legitimate users should apply a universal hash function on the concatenation of both Alice and Bob keys to generate a shared key $\mathcal{K}_{AB} = H(\mathcal{K}_{Alice}||\mathcal{K}_{Bob})$. Thus, if Eve was successful in obtaining one of the legitimate users key, Eve will not be able to obtain the shared key. This is due to the fact that any small change in the hash function input will cause the output to change drastically. Nevertheless, the security of C-MOPRO can be compromised if two active attackers placed themselves near Alice and Bob simultaneously. Still, this requires the two attackers to exchange data risking alerting either Alice or Bob which might result in terminating the communication.

In addition, implementing the physical layer security mechanism allows us to authenticate the legitimate users during channel establishment phase. The wireless channel between the legitimate users becomes decodable after the transmission of the rotated reference signals and hence we can authenticate transmitted signals to prevent MitM attack. Alternatively, traditional cryptography usually authenticates and secures the channel after the channel has been established and usually does not concern itself with this process. With the rise in fear of the inevitable large-scale implementation of quantum computers, many of the digital signature schemes that rely on the integer factorization problem, the discrete logarithm problem, or the elliptic curve discrete logarithm problem can be solved easily. Therefore, we opted to implement SPHINCS to authenticate the legitimate users. The authors of SPHINCS proved its security against quantum attacks since it only depends on the usage of secure cryptographic hash functions.

Finally, it is important to note that MOPRO and C-MOPRO provide information theoretic security. The authors in [8], showed that using the rotation matrices decreases Eve's knowledge about the channel.

$$\bar{H}(\mathbf{h}_{AB}|\mathbf{h}_{AE}) \leq \bar{H}(\mathbf{h}_{AB}\mathbf{G}_2|\mathbf{h}_{AE}\mathbf{G}_2) \tag{2}$$

and

$$\bar{H}(\mathbf{h}_{BA}|\mathbf{h}_{BE}) \leq \bar{H}(\mathbf{h}_{BA}\mathbf{G}_1|\mathbf{h}_{BE}\mathbf{G}_1) \tag{3}$$

where $\bar{H}$ is the entropy and $\mathbf{h}$ is the simplified channel matrix.

VI. Performance Analysis

Overhead comparison between MOPRO, Diffie-Hellman + RSA, and the proposed C-MOPRO is detailed in TABLE III. The table shows the computation and communication overhead for Alice and Bob, respectively. In MOPRO, the generation of Alice and Bob respective keys requires no computation overhead in terms of the number of exponentiations and hash operations. Since their secret keys are embedded into the required reference signals to estimate the channel, one of the users does not acquire communication overhead. However, the other user will need to send additional $n_b\mathbf{Gr}$ reference signals to communicate its secret key securely.

Alternatively, the Diffie-Hellman + RSA algorithm requires each Alice and Bob one exponentiation to agree on a key. Additionally, it requires each Alice and Bob one

exponentiation and one hash function operation to authenticate or verify the exchanged messages. Furthermore, the communication overhead associated with Diffie-Hellman + RSA algorithm is the result from communicating Diffie-Hellman parameters, RSA public keys, and RSA signature.

On the other side, our proposed C-MOPRO algorithm requires each Alice and Bob one hash function operation to agree on a shared secret key. Also, it requires each Alice and Bob n_bC to authenticate and verify the exchanged signals. As in MOPRO, our proposed solution requires additional $n_b\mathbf{Gr}$ reference signals to transmit the second secret key. On top of that, C-MOPRO needs to communicate Alice/Bob public keys and signatures to authenticate the messages. It is important to highlight that the parameter n_b in MOPRO and C-MOPRO is a design choice and depends on the total bandwidth, the sub-band bandwidth, and the desired length of the secret key. In fact, selecting an appropriate number of sub-bands is critical since it affects the computation and communication overhead. Hence, in our future work, we aim to find the optimal n_b that results in a reasonable overall overhead and yet maintains high system capacity.

The main contributing factor in C-MOPRO overhead is due to SPHINCS which is computationally costly when compared to traditional digital signatures. Nevertheless, in the age of quantum computing, SPHINCS and other post-quantum schemes must be used instead of traditional cryptography signatures, e.g. RSA. As a matter of fact, all post-quantum hash-based signatures result in higher overhead compared to traditional signatures [9], [16]. This is the trade-off between security and performance. Other than SPHINCS overhead, C-MOPRO has a reasonable computational and communicational overhead when compared to post-quantum key exchange algorithms. This is due to the fact that the key agreement in C-MOPRO is done during the channel establishment phase and it does not require a generation of a public and private key pair to agree on a secret key. In addition, it has been established that many post-quantum key exchange protocols are computationally costly [17], [18]. For example, Supersingular Isogeny Diffie-Hellman (SIDH) key exchange which serves as a replacement to DH takes 303ms to agree on a key[2] [19]. This does not include the time needed for channel establishment and message authentication.

VII. Conclusion and future work

In this paper, we proposed the C-MOPRO algorithm which is a post-quantum hybrid security algorithm. This cross-layer security mechanism combines cryptographic techniques and physical layer security to achieve a powerful security mechanism with a reasonable overall overhead. In this scheme, the key agreement is accomplished during the channel establishment phase. Also, during this phase, we address MitM attack using SPHINCS digital signature. Furthermore, we tackle the problem where half of the secret key bits gets compromised when Eve is located near either Alice or Bob. This is done using a universal secure hash function that guarantees the security of the shared secret key even if half of the secret key bits is exposed.

As a future work, we plan to derive the exact overhead and the optimal number of sub-bands. Moreover, we will extensively simulate and evaluate C-MOPRO against MOPRO and cryptographic techniques. Finally, we plan to investigate the possibility of implementing a real world testbed and the possibility of deploying our work in real case scenario.

[2]This was measured on Macbook Pro Intel Core i5-2415M @ 2.4 GHz.

References

[1] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb 1978.

[2] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[3] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Computing 26th*, pp. 1484–1509, 1997.

[4] A. Wyner, "The wire-tap channel," *Bell Syst. Tech. J.*, vol. 54, pp. 1355–1387, 1975.

[5] S. Cheong and M. Hellman, "The gaussian wire-tap channel," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 4, pp. . 451–456, Jul. 1978.

[6] A. Yener and S. Ulukus, "Wireless physical-layer security: lessons learned from information theory," *Proceedings of the IEEE*, vol. 103, no. 10, pp. 1814–1825, Oct. 2015.

[7] W. Trappe, "The challenges facing physical layer security," *IEEE Communications Magazines*, vol. 53, no. 6, pp. 16–20, June 2015.

[8] C. Wu, P. Lan, P. Yeh, C. Lee, and C. Cheng, "Practical physical layer security schemes for mimo-ofdm systems using precoding matrix indices," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 1687–1700, Sept. 2013.

[9] D. Bernstein, D. Hopwood, A. Hulsing, T. Lange, R. Niederhagen, L. Papachristodoulou, M. Schneider, P. Schwabe, and Z. Wilcox-O'Hearn, "Sphincs: Practical stateless hash-based signatures," *EUROCRYPT*, vol. 9056, no. 8, pp. 368–397, April 2015.

[10] R. Merkle, "A certified digital signature," *Advances in Cryptology - CRYPTO '89*, vol. 435, pp. 218–238, 1990.

[11] O. Goldreich, "Two remarks concerning the goldwasser-micali-rivest signature scheme," *Advances in Cryptology - CRYPTO '86*, vol. 263, pp. 104–110, 1990.

[12] A. Hulsing, "W-OTS+ shorter signatures for hash-based signature schemes," *Africacrypt*, vol. 7918, pp. 173–188, 2013.

[13] L. Reyzin and N. Reyzin, "Better than biba: Short one-time signatures with fast signing and verifying," *Information Security and Privacy*, vol. 2384, pp. 1–47, 2002.

[14] U.S. National Security Agency, "Commercial national security algorithm suite and quantum computing FAQ," Jan. 2016.

[15] U. Maurer and S. Wolf, "Information-theoretic key agreement: From weak to strong secrecy for free," *EUROCRYPT*, vol. 1807, pp. 351–368, 2000.

[16] T. Eisenbarth, I. Maurich, and X. Ye, "Faster hash-based signatures with bounded leakage," *Selected Areas in Cryptography 20th International Conference*, pp. 223–243, August 2013.

[17] D. Jao and L. Feo, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," *Post-Quantum Cryptography 4th International Workshop*, pp. 19–34, 2011.

[18] C. Delfs and S. Galbraith, "Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$," *Codes and Cryptography Designs*, vol. 78, no. 2, pp. 425–440, 2014.

[19] R. Azarderakhsh, D. Fishbein, and D. Jao, "Efficient implementations of a quantum-resistant key-exchange protocol on embedded systems," *Citeseer*.