Secure Physical Layer Voting

Nirnimesh Ghose, Bocan Hu, Yan Zhang, and Loukas Lazos
Dept. of Electrical and Computer Engineering, University of Arizona
Email: {nghose, bocanhu, yanzhang, llazos}@email.arizona.edu

***Abstract*—Distributed wireless networks often employ voting to perform critical network functions such as fault-tolerant data fusion, cooperative sensing, and reaching consensus. Voting is implemented by sending messages to a fusion center or via direct message exchange between participants. However, the delay overhead of message-based voting can be prohibitive when numerous participants have to share the wireless channel in sequence, making it impractical for time-critical applications.**

In this paper, we propose a *fast* PHY-layer voting scheme called PHYVOS, which significantly reduces the delay for collecting and tallying votes. In PHYVOS, wireless devices transmit their votes simultaneously by exploiting the subcarrier orthogonality of OFDM and without explicit messaging. Votes are realized by injecting energy to pre-assigned subcarriers. We show that PHYVOS is secure against adversaries that attempt to manipulate the voting outcome. Security is achieved without employing cryptography-based authentication and message integrity schemes. We analytically evaluate the voting robustness as a function of PHY-layer parameters. We extend PHYVOS to operate in ad hoc groups, without the assistance of a fusion center. We discuss practical implementation challenges related to multi-device frequency and time synchronization and present a prototype implementation of PHYVOS on the USRP platform. We complement the implementation with larger scale simulations.

***Index Terms*—Physical-layer security, voting, OFDM, wireless, data fusion.**

I. Introduction

Distributed wireless networks fundamentally rely on the principle of cooperation. Nodes often share information to coordinate network functions and improve the fault-tolerance of distributed operations. As an example, cooperative spectrum sensing is known to improve the detection of licensed user activity in dynamic spectrum access (DSA) [1]. Data fusion is also widely used in wireless sensor networks (WSNs) for improving the performance of target detection, target tracking, and distributed sensing [2].

For many cooperative functions, binary voting algorithms increase fault-tolerance at relative low cooperation overhead. In binary voting, a community of distributed entities shares binary decisions ("yes" or "no") on a parameter of interest (e.g., channel state, presence of a target). A combining decision rule is applied to collectively determine the decision outcome. This rule is based on some form of majority voting, plurality or threshold, to achieve the desired level of reliability. Typically, binary votes are casted using a messaging scheme, in which 1-bit votes are carried by individual messages. However, message-based voting incurs relatively high voting delay. In this work, *we define the voting delay as the time period between the initiation of the voting process with the transmission of the first vote by any of the participants, until all votes have been received at the tallier*. The tallying time is not accounted as part of the voting delay. For message-based voting, each 1-bit vote is carried by a packet that contains PHY layer and a MAC layer headers. Moreover, verifying the voter authenticity and protecting the integrity of binary votes via digital signatures and message authentication codes, requires additional packet fields. All additional fields (headers, message authentication codes, digital signature) increase the overall transmission time per vote. Further, voters must sequentially access the shared wireless channel to cast their votes. Most popular channel access protocols include anti-collision mechanisms (e.g., backoff process) that further increase the voting delay to cast multiple votes. For time-critical applications, a high voting delay could be unacceptable [8], [11].

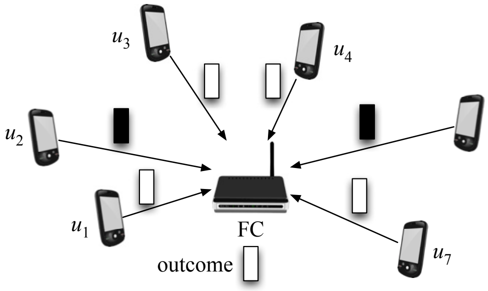


Fig. 1: The PHYVOS voting scheme.

As an example, consider the cooperative spectrum sensing mechanism proposed for DSA networks [1]. To accurately determine spectrum opportunities, secondary users sense licensed channels and submit state information ("busy" or "idle") to a fusion center (FC). The FC applies a combining decision rule (e.g., majority voting) to reliably determine the state of each channel. Existing federal regulations mandate that channel sensing must occur *every two seconds* [11], which leads to the frequent repetition of the fusion process. At such frequency, the time delay of message-based voting becomes problematic as the number of participants increases. Similar time and scalability constraints are encountered in control applications of networked multi-agent systems, where the consensus time requirement could be even more stringent [8].

To address the poor delay scalability of message-based voting, we present a *secure* and *fast* voting scheme called PHYVOS that implements voting at the PHY layer. The basic principle of PHYVOS is shown in Fig. 1. Wireless devices exploit the subcarrier orthogonality in the widely adopted orthogonal frequency division modulation (OFDM), to simultaneously cast their votes to an FC within just a few symbols. PHYVOS yields two distinct advantages relative to message-based voting. First, participants do not have to sequentially access the shared channel to cast their votes. This feature leads to significant delay savings, as delays due to contention and sequential access are eliminated. Second, votes do not carry long headers and cryptographic signatures that prolong the message transmission time. Therefore, PHYVOS drastically reduces the delay of voting, while maintaining a high security level. Implementing secure voting at the PHY layer involves new security and implementation challenges.

- Voting at the PHY layer is susceptible to false vote insertion and vote modification attacks, similar to message-based voting. An adversary can alter the voting outcome by exploiting the open nature of the wireless medium and manipulating the transmitted signals at the PHY layer. Without access to cryptographic primitives such as digital signatures and message authentication codes, securing the voting process is particularly challenging.
- The superposition of simultaneous transmissions from spatially-separated senders (voters) to a combined OFDM signal requires intricate transmitter and receiver designs [9], [27]. Senders must be synchronized in frequency and time to achieve symbol alignment at the receiver. Maintaining accurate synchronization in distributed systems could incur prohibitive coordination overheads [27].

Our Contributions: We design PHYVOS, a PHY-layer voting scheme that reduces the voting delay by several orders of magnitude compared to message-based voting. In PHYVOS, the voting delay, defined as the time required to cast votes, is reduced by exploiting the subcarrier orthogonality of OFDM to simultaneously cast votes from multiple participants. Vote tallying is performed at an FC that receives multiple votes as a single OFDM symbol. We further present a fully distributed version that allows every participant compute the vote tally, without the assistance of an FC. To overcome the challenges related to decoding simultaneous transmissions from multiple senders, binary votes are casted by adding energy to designated subcarriers. No transmission of preambles and headers is required, as the receiver does not demodulate the OFDM signal. Simple energy detection suffices. Moreover, relying on energy detection rather than message decodability for vote casting strengthens the security of our scheme, as it is generally hard to "erase" energy from a channel [10], [24].

We study the robustness of PHYVOS against an external and an internal adversary. The former attempts to modify votes by inserting energy into various subcarriers without knowing the subcarrier allocation. The latter is aware of any group secrets used to assign subcarriers, but not of pairwise secrets. PHYVOS guarantees the integrity of the voting outcome. We show that an active adversary who attempts to modify the casted votes, cannot flip the voting outcome at the FC with overwhelming probability. Also, the adversary cannot inject additional votes at the FC. We improve voting robustness by incorporating the transmission of multiple OFDM symbols to cast a single vote, thus realizing a repetition code. Since OFDM symbols have very short duration, a repetition code is still far more efficient than messaging. We analytically evaluate the voting robustness as a function of the relevant system parameters under a secret and an open vote model. We discuss practical implementation challenges of PHYVOS related to frequency and time synchronization. We present a prototype implementation of PHYVOS on the NI USRP platform. We complement the implementation with larger scale simulations and demonstrate the PHYVOS robustness to external and internal attacks.

TABLE I: Notation

Notation	Definition
A	Administrator
R	Tallier
M	Number of voting participants
$u_1, u_2, \ldots, u_M$	The M voting participants
$f_1, f_2, \ldots, f_N$	The N OFDM subcarriers
$v_1, v_2, \ldots, v_M$	Votes of each participant
ℓ	Number of symbol votes per voting round
$v_i(n)$	Symbol vote of u_i at the n^{th} symbol
$\mathcal{T}$	Voting outcome computed by the tallier
γ	Decision threshold for the voting outcome
μ	Voting margin

PHYVOS is compatible with any wireless standard that is based on OFDM. This includes 802.11a/g/n/ac, WiMAX, UWB, DVB, and others. PHYVOS requires no hardware modifications of the OFDM TX/RX circuitry. Participants cast votes by transmitting regular OFDM symbols, and the RX can decipher votes at the FFT module of the OFDM receiver.

Paper Organization: In Section II, we present the system, communication, and adversary models. Section III describes PHYVOS. In Section IV, we analyze the security of PHYVOS under internal and external adversaries. A fully distributed version of PHYVOS without an FC is presented in Section V. In Section, VI, we compare the overhead of PHYVOS with the overhead of message-based voting. Practical considerations and experimental verification of PHYVOS' performance are presented in Section VII. In Section VIII, we discuss related work and conclude in Section IX.

II. Notation and Models

A. Notation Summary

Table I summarizes the most frequently used notation.

B. Entities

The following entities are involved in the voting process:

- The *administrator* (A) is responsible for initializing the participants and the tallier with relevant cryptographic quantities, after verifying their identities.
- The M *participants* $u_1, u_2, \ldots, u_M$ cast M votes $v_1, v_2, \ldots, v_M$ to the tallier. Each vote reflects a binary choice.
- The *tallier* (R) is responsible for verifying and tallying the votes of all the participants by computing the voting outcome $(\mathcal{T})$.
- The *adversary* attempts to alter the voting outcome by injecting his own signals during the voting process.

In most applications, A and R could be the same entity such as the fusion center shown in Fig. 1.

C. Voting Model

During the voting process, M participants cast M votes $v_1, v_2, \ldots, v_M$ to the tallier. For ease of illustration, we analyze the case where binary votes are casted, i.e., $v_i \in$

$\{0,1\}$,$\forall i$. The tallier R computes the voting outcome according to a threshold decision rule.

$$\mathcal{T} = \begin{cases} 1, & if \quad \sum_{i=1}^{M}(-1)^{v_i} \ < \ \gamma \\ 0, & if \quad \sum_{i=1}^{M}(-1)^{v_i} \ \geq \ \gamma. \end{cases} \tag{1}$$

The value of γ is application-dependent. As an example, by setting $\gamma = 0$, a plurality rule is implemented. Other values of γ allow for more relaxed or stricter agreement. The voting process must satisfy the requirements of correctness and robustness defined as follows.

Definition 1 (Correctness): In the absence of attacks, all votes must be unambiguously recorded and tallied. That is, the voting must be error-free.

Definition 2 (Robustness): A voting scheme is said to be robust against active attacks and faults, if the estimated outcome $\hat{\mathcal{T}}$ at the tallier equals the true outcome $\mathcal{T}$ computed by tallying the vote intend of all participants.

Robustness is a weaker requirement than accuracy, because it can be satisfied even if some votes are incorrectly tallied. However, robustness is sufficient for the intended applications of PHY-layer voting. We emphasize that other well-known voting requirements such as receipt-freeness [29], are beyond the scope of the envisioned applications of PHY-layer voting.

D. Communication Model

We consider a one-hop communication topology, where every participant is either within the communication range of the tallier (star topology), or within one hop of each other (complete graph). Therefore votes are directly casted without a relay. Participants cast their votes to the tallier using an OFDM system with N orthogonal subcarriers, denoted by $f_1, f_2, \ldots f_N$. Participants could be at varying distances from the tallier. Moreover, participants and the tallier are synchronized to a time-slotted system with a maximum synchronization error of Δt, which depends on clock drifts and multipath. Note that time synchronization is already required for other network functions such as media access control.

Each participant must meet a minimum SNR requirement to cast a vote. This assumption is also true for message-based voting, where a sufficiently high SNR must be achieved to perform error-free decoding. As our method relies on energy detection, no other requirements are placed on the channel model. Different channels (e.g., AWGN, Rayleigh, Rician) could model the participant-to-tallier communications. If a participant's channel has an SNR below the required threshold for vote detection due to destructive interference, for all practical purposes this participant is no longer part of the voting. Finally, the channel state is assumed to be difficult to predict without being very close (within a few wavelengths) of the receiver, and without the transmission of preambles. This is true for most multipath scenarios, as it has been demonstrated by several works (e.g., [10], [12], [19], [24]).

E. Adversary Model

The adversary aims at flipping the voting outcome $\hat{\mathcal{T}}$ computed at the tallier. The adversary could be *external* or *internal*. An external adversary is unaware of any cryptographic primitives used to initialize participants. An internal adversary on the other hand, is a legitimate participant with access to any group secrets. We assume that the adversary does not launch denial-of-service (DoS) attacks that prevent the computation of any voting outcome (e.g. by eliminating the votes of every participant). Such an attack is easily detectable. The adversary is loosely synchronized to the tallier with the same synchronization error as the rest of the participants. Two different voting models are considered with respect to the secrecy of the vote intent of each participant:

Secret vote model: In the secret vote model, the adversary is not aware of the vote intent of the participants.

Open vote model: In this model, the adversary is aware of the vote intent of the targeted participants. The vote intent can be determined by some side-channel information. For instance, in a spectrum sensing application for CRNs, the vote intent of an honest participant can be determined by performing spectrum sensing on a nearby location.

III. PHYVOS: Physical Layer Voting

The key principle of PHYVOS is to simultaneously cast votes by injecting energy on designated subcarriers. An adversary attempting to modify a vote on subcarrier f_i, would have to "erase" the signal received by the tallier on f_i and simultaneously inject energy on some other subcarrier. This is generally a hard problem that requires knowledge of the signal transmitted at f_i, the precise time that the signal was transmitted, the signal propagation delay, and precise channel state information [7], [10], [24]. This knowledge needs to be collected and synchronized for all voters. PHYVOS consists of four phases: the setup phase, the vote request phase, the vote casting phase, and the tallying phase.

A. Setup Phase

In the setup phase, the administrator initializes the tallier and the M participants. If the tallier and the administrator are the same entity, only the M participants need to be initialized. The initialization process is as follows.

Key generation: The administrator $\mathcal{A}$ executes a probabilistic key generator algorithm $KeyGen(1^{\tau}) \rightarrow K$. This algorithm takes as input a security parameter τ, and outputs a master key K. $\mathcal{A}$ derives $M+1$ additional keys from K with $K_{perm} = H_{perm}(K)$ and $K_{vote,i} = H_{vote}(K, i)$ for $i = [1..M]$, where H_{perm} and H_{vote} are cryptographic hash functions.

Key assignment: $\mathcal{A}$ loads K_{perm} and $K_{vote,i}$ with $i \in [1..M]$ to R. It also loads K_{perm} and $K_{vote,i}$ to each u_i. At the end of the setup phase, R shares one pairwise key $K_{vote,i}$ with each u_i and a common key K_{perm} with all u_is.

B. Vote Request Phase

In the vote request phase, the tallier synchronizes all participants for simultaneous voting. This phase is necessary to ensure that delay overhead gains are achieved by the simultaneous vote casting. Periodic or on-demand voting can

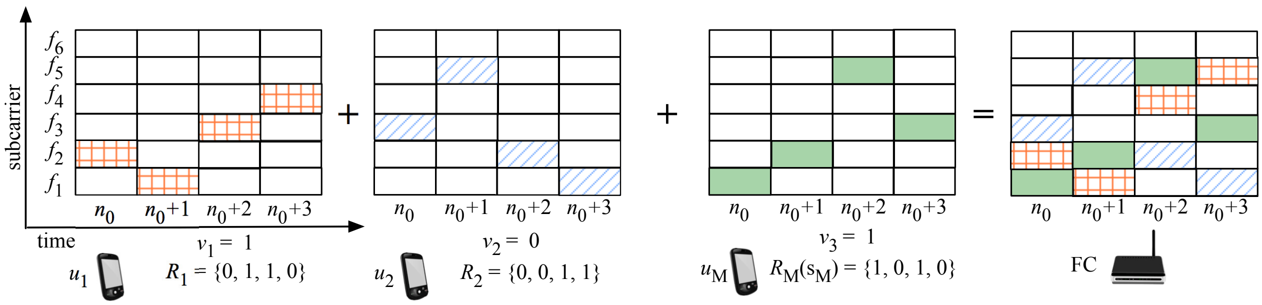


Fig. 2: The vote casting phase for M participants voting over N subcarriers (here $N = 2M$).

be employed to request a vote. In periodic voting, participants exploit their synchronization to a common time-slotted system to cast their votes at fixed time intervals without an explicit request from the tallier. This operation mode is suitable for periodic network operations. In on-demand voting, the tallier broadcasts a vote request synchronization message to all participants to initiate the voting process.

C. *Vote Casting Phase*

During the vote casting phase, participants simultaneously cast their votes to the tallier. Each vote v_i consists of a series of ℓ symbol votes $v_i(n_0), v_i(n_0+1), \ldots, v_i(n_0+\ell-1)$ casted over ℓ consecutive time slots. The ℓ symbol votes operate as a repetition code to improve the robustness of vote casting in the presence of an adversary. To cast a symbol vote $v_i(n)$ at the n^{th} time slot, a participant u_i is assigned two subcarriers $f^0_{u_i}(n)$ and $f^1_{u_i}(n)$. One subcarrier is used to cast a "no" vote whereas the other is used to cast a "yes" vote. We note that in the absence of an adversary, a single subcarrier is sufficient to cast a binary vote. However, the adversary could easily modify the vote that corresponds to energy absence by injecting energy on the alternative subcarrier. Therefore, we adopt a two-subcarrier solution.

Moreover, the subcarriers assigned to each participant are permuted per time slot to hide the assignment from the adversary. This is achieved by applying a pseudo-random permutation on the subcarrier assignment. Finally, for a given assignment $f^0_{u_i}(n)$ and $f^1_{u_i}(n)$ to participant u_i, the mapping to "yes" and "no" votes is randomized by the application of a pseudo-random binary sequence shared between u_i and R. This prevents an internal adversary from determining the subcarrier that corresponds to a specific vote. Formally, vote casting involves the following steps:

1) **Subcarrier assignment:** Each participant u_i applies pseudo-random function

$$\Pi_F : \{0,1\}^{\tau} \times [1..N] \times \mathbb{Z}^+ \rightarrow [1..N],$$

to map subcarrier with index p during slot n, to subcarrier $\Pi_F(K_{perm}, p, n)$. Participant u_i is assigned subcarriers

$$\begin{aligned} f^0_{u_i}(n) &= f_{\Pi_F(K_{perm},(2i-1),n)}, \\ f^1_{u_i}(n) &= f_{\Pi_F(K_{perm},2i,n)}. \end{aligned}$$

2) **Pseudo-random sequence generation:** Each participant u_i applies pseudo-random generator function

$$\Phi : \{0,1\}^{\tau} \times \mathbb{Z}^+ \rightarrow \{0,1\}$$

to generate a binary sequence $R_i = \{r_i(1), r_i(2), \ldots\}$ with $r_i(n) = \Phi(K_{vote,i}, n)$.

3) **Symbol vote casting:** Let voting casting be initiated at slot n_0. To cast a vote $v_i \in \{0,1\}$, a participant u_i generates ℓ symbol votes $v_i(n_0) = v_i(n_0+1) = \ldots = v_i(n_0+\ell-1) = v_i$. Each $v_i(n)$ is represented by an OFDM symbol with the following values per subcarrier

$$x_k(n) = \begin{cases} \alpha_y, & f^{v_i(n)\oplus r_i(n)}_{u_i}(n) \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where α_y is a randomly selected modulation symbol and $n_0 \leq n < n_0 + \ell$. Note that the placement of energy of either $f^0_{u_i}(n)$ or $f^1_{u_i}(n)$ is based on the XOR between the vote value $v_i(n)$ and the random bit $r_i(n)$.

The vote casting phase for three participants and six subcarriers is shown in Fig. 2. In Step 1, participants apply the pseudo-random permutation to obtain the subcarrier assignment. For the first four time slots, the subcarrier permutations are $\{f_6,f_2,f_3,f_4,f_1,f_5\}$, $\{f_1,f_3,f_5,f_6,f_4,f_2\}$, $\{f_3,f_1,f_6,f_2,f_5,f_4\}$ and $\{f_5,f_4,f_2,f_1,f_6,f_3\}$. Participant u_1 is assigned $\{f^0_{u_1}, f^1_{u_1}\} : \{(f_6,f_2),(f_1,f_3),(f_3,f_1),(f_5,f_4)\}$, participant u_2 is assigned $\{(f_3,f_4),(f_5,f_6),(f_6,f_2),(f_2,f_1)\}$ and participant u_3 is assigned $\{(f_1,f_5),(f_4,f_2),(f_5,f_4),(f_6,f_3)\}$. In Step 2, each participant u_i generates the pseudo-random sequence for slots $1-4$. For u_1, $r_1 = \{0,1,1,0\}$, for u_2, $r_2 = \{0,0,1,1\}$,, and for for u_3, $r_3 = \{1,0,1,0\}$,. In Step 3, participants cast votes at the designated subcarriers. In our example, u_1 wants to cast a "yes" vote ($v_1 = 1$). He XORs v_1 with r_1 and determines the active subcarriers as $\{f_2, f_1, f_3, f_4\}$ for slots $1, 2, 3$, and 4, respectively. The active subcarriers for other participants are similarly determined. The symbol votes arrive (almost) time aligned at the tallier such that OFDM symbols are formed as shown in Fig. 2.

D. *Vote Tallying Phase*

In the vote tallying phase, the tallier computes the voting outcome $\mathcal{T}$ according to the threshold rule in (1). To infer the votes of each participant, the tallier computes the FFT of the digitized baseband OFDM signal to separate the spectral components to each of the subcarriers. The tallier then uses an energy detector at each output of the FFT block to detect the transmitted symbol votes. Note here that *no symbol demodulation is necessary to determine the presence of energy*. At time n, a symbol vote $v_i(n)$ is computed only if the detected average power is beyond a threshold γ_D on only one of the two designated subcarriers. In any other case, the symbol vote is recorded in error. Formally, for a participant u_i, the recovery of v_i at the tallier is performed as follows.

1) **Energy detection:** Sample the FFT output of subcarriers $f^0_{u_i}(n)$ and $f^1_{u_i}(n)$ assigned to u_i and compute the average received power over L samples:

$$p^0_{u_i}(n) = \frac{1}{L}\sum_{i=1}^{L}|y_j(i)|^2, \;\; p^1_{u_i}(n) = \frac{1}{L}\sum_{i=1}^{L}|y_{j+1}(i)|^2, \tag{3}$$

with $n_0 \le n < n_0 + \ell$.

2) **Extract symbol votes:** The symbol votes $\hat{v}_i(n)$ are computed by XORing the subcarrier superscript were energy was detected with the pseudo-random sequence shared between u_i and the tallier to correctly map the subcarrier index to the vote value.

$$\hat{v}_i(n) = \begin{cases} 0 \oplus r_i(n), & \text{if } p^0_{u_i}(n) > \gamma_D, \;\; p^1_{u_i}(n) \le \gamma_D \\ 1 \oplus r_i(n), & \text{if } p^0_{u_i}(n) \le \gamma_D, \;\; p^1_{u_i}(n) > \gamma_D \\ e, & \text{otherwise.} \end{cases} \tag{4}$$

with $n_0 \le n < n_0 + \ell$.

3) **Compute the final vote:** The final vote $\hat{v}_i$ is computed by discarding all inconclusive symbol votes.

$$\hat{v}_i = \begin{cases} 0, & \text{if } \sum_{n=n_0, v_i(n)\neq e}^{n_0+\ell-1}(-1)^{v_i(n)} > 0 \\ 1, & \text{if } \sum_{n=n_0, v_i(n)\neq e}^{n_0+\ell-1}(-1)^{v_i(n)} < 0 \\ e, & \text{otherwise.} \end{cases} \tag{5}$$

4) **Compute the final voting outcome:** The final voting outcome $\hat{\mathcal{T}}$ computed according to:

$$\hat{\mathcal{T}} = \begin{cases} 1, & if \;\; \sum_{i=1}^{M}(-1)^{\hat{v}_i} \;<\; \gamma \\ 0, & if \;\; \sum_{i=1}^{M}(-1)^{\hat{v}_i} \;\ge\; \gamma. \end{cases} \tag{6}$$

The voting outcome $\hat{\mathcal{T}}$ is estimated by tallying all votes using eq. (1), where the vote values v_i have been substituted by their estimates $\hat{v}_i, i \in [1..M]$. The tallying operation is shown in the example of Fig. 2. For participant u_1, the tallier detects an average power over γ_D on subcarriers $\{f_2, f_1, f_3, f_4\}$. By XORing the output $\{1, 0, 0, 1\}$ with the random sequence $R_1 = \{0, 1, 1, 0\}$, it obtains the symbol votes $\hat{v}_1(n_0) = 1$, $\hat{v}_1(n_0 + 1) = 1$, $\hat{v}_1(n_0 + 2) = 1$, and $\hat{v}_1(n_0 + 3) = 1$, indicating a final vote $\hat{v}_1 = 1$. Similarly, participant u_2 uses random sequence $R_2 = \{1, 1, 0, 1\}$ to compute $v_2 = 0$. The vote computation proceeds in parallel for all participants. The voting outcome is estimated to be $\hat{\mathcal{T}} = 1$.

IV. Security Analysis

In this section, we evaluate the robustness of PHYVOS under the external and internal adversary model.

A. External Adversary

Under the external adversary model, the adversary is unaware of the cryptographic keys K_{perm} and $K_{vote,i}$ used to permute the subcarrier assignment per time slot and also randomize the symbol votes. Therefore, his best strategy is to inject energy on randomly selected subcarriers. Let us consider the vote v_i of u_i, consisting of ℓ symbol votes $v_i(n_0)$, $v_i(n_0 + 1), \ldots, v_i(n_0 + \ell - 1)$,. To successfully cast v_i, the adversary must guess the subcarrier of u_i that dictates the vote opposite to v_i for ℓ symbol votes. Even if the subcarrier is correctly guessed, the adversary cannot "erase" the energy injected by the legitimate participant on the complementary subcarrier. Erasure of the modulation symbol a_y transmitted by u_i requires the a priori knowledge of a_y, knowledge of the channel between the voter and the tallier as well as the adversary and the tallier, and precise synchronization between the voter and the adversary [7]. We note that u_i randomly selects a_y for each symbol vote. Moreover, the channel between u_i and tallier rapidly decorrelates with the distance from u_i. Unless the adversary is very close to u_i, the channel between u_i and tallier is unpredictable [21].

Without the opportunity to flip votes, the adversary can flip the voting outcome if he nullifies a sufficient number of in favor votes to overcome the decision threshold γ. Vote nullification occurs, if energy is present on both subcarriers assigned to a participants over the ℓ symbol votes. Let the adversary inject energy on $J \le N$ subcarriers of his choice in order to flip the voting outcome $\mathcal{T}$. Without loss of generality assume that votes in favor of $\mathcal{T}$ outnumber the votes against $\mathcal{T}$ by a voting margin μ. The probability of flipping the outcome to an estimate $\hat{\mathcal{T}} \neq \mathcal{T}$ is given in Proposition 1.

Proposition 1: Let participants cast M votes over $2M \le N$ subcarriers by transmitting ℓ symbol votes. Let an external adversary inject energy on $J \le N$ subcarriers. The probability of flipping the voting outcome is

$$\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] = \sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \Pr[\mathbf{Z} = z], \tag{7}$$

where $n_1 = \frac{M+\mu}{2}$ denotes the number of votes in favor of $\mathcal{T}$ and $n_2 = \frac{M-\mu}{2}$ denotes the votes against $\mathcal{T}$.

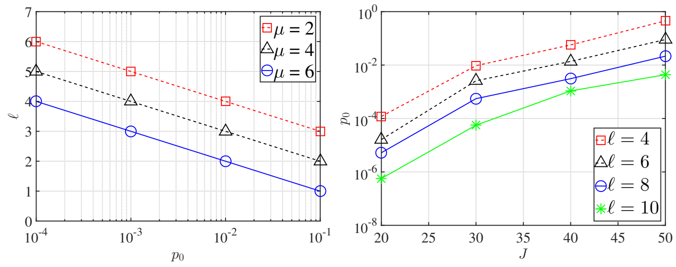
$$\begin{aligned} \Pr[\mathbf{Z} = z] \;\; = \;\; & \sum_{x} \left(\binom{n_1}{x}\binom{n_2}{x-z} \left(\frac{\binom{N-2x+z}{J-2x+z}}{\binom{N}{J}} \right)^{\ell} \right. \\ & - \left(\sum_{w=1}^{\min\{n_1-x, J-x\}} \sum_{k=0}^{w-z} \binom{n_1 - x}{w} \right. \\ & \quad \left. \binom{n_2 - x + z}{k} \frac{\binom{N-n_1-n_2-4x+2z}{J-w-k-2x+z}}{\binom{N-x}{J-x}} \right) \\ & \quad \left. \left(\frac{1}{\binom{N}{J}} \right)^{\ell} \right). \end{aligned} \tag{8}$$

Proof: The proof is provided in Appendix A. ■

Selecting the Security Parameter ℓ: Proposition 1 allows us to select the number of symbol votes ℓ to guarantee robustness with a desired probability p_0. The following corollary yields a lower bound on ℓ such that $\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] \le p_0$.

Corollary 1: For an external adversary, $\Pr\,[\hat{\mathcal{T}} \neq \mathcal{T}] \le p_0$ if

$$\ell > \lceil \frac{1}{\log \frac{1}{C_1}} \log \frac{C_0}{p_0} \rceil, \tag{9}$$



(a) (b)

Fig. 3: (a) Minimum number of symbol votes ℓ to guarantee robustness p_0 for an external adversary for va, (b) minimum number of symbol votes ℓ to guarantee robustness p_0 for $\mu = 4$ and various J.

where

$$C_0 = \sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \left(\binom{n_1}{x}\binom{n_2}{x-z}\right), \quad (10)$$

$$C_1 = \sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \frac{\binom{N-2x+z}{J-2x+z}}{\binom{N}{J}}. \quad (11)$$

Proof: The proof is provided in Appendix B. ■

From Corollary 1, we observe that the required number of symbol votes ℓ drops linearly with the logarithm of p_0. This is also confirmed by Fig. 3(a), which shows ℓ as a function of p_0, for a total for 20 participants voting over 52 subcarriers. In Fig. 3(a), we set $J = 12$, $\gamma = 0$ (plurality rule) and varied the vote margin μ. We observe that a relatively small number of symbol votes ($\ell < 5$) allows us to achieve high levels of robustness for relatively small margins.

An obvious tactic for the adversary is to increase the number of attacked subcarriers. In Fig. 3(b), we show the number of symbol votes required to achieve a desired robustness level for various J, when the vote margin is fixed to $\mu = 4$. We observe that if small number of subcarriers are attacked, the achieved robustness is high for small ℓ. The adversary's success increases with J at the expense of increased presence over the various subcarriers.

To prevent the adversary from flipping the voting outcome via vote nullification, the tallier can reject the voting outcome if the fraction of nullified votes exceeds a certain threshold. This threshold can be defined to exceed the expected number of nullified votes under unintentional interference. In Section VII, we explore this prevention method by determining the pmf for the number of nullified votes due to the imperfections of the wireless channel. The pmf is used to select the threshold for rejecting the voting outcome.

B. Internal Adversary

An internal adversary could be any malicious participant aiming at manipulating the voting outcome. Such an adversary has knowledge of the key K_{perm} used for the subcarrier assignment. Therefore, it can target particular subcarriers to nullify votes of certain participants. Note that we do not consider the case where the adversary compromises the credentials (pairwise keys) of several participants by, for example, gaining access to the participants' devices. In this case, the adversary can impersonate the compromised participants and cast votes on their behalf. For all practical purposes, such impersonations cannot be authenticated using cryptographic methods, and can only be detected using radio fingerprinting methods. Such attacks are possible against message-based voting systems as well, and cannot be defended by standard cryptographic methods of authentication and message integrity.

Modifying a Single Vote: We first analyze the modification of vote v_i of a targeted participant u_i. Let u_i initiate its voting at time slot n_0 by submitting ℓ symbol votes. Although the adversary is aware of the subcarriers assigned to u_i, he is unaware of the pseudo-random sequences used to map the subcarriers to the "yes/no" votes. Without access to the pairwise key $K_{vote,i}$, the adversary can at best guess the subcarrier where energy must be injected to emulate a "yes" or a "no" vote. We consider two possible adversary strategies. In the first strategy, the adversary randomly selects one of u_i's subcarriers to emulate a target vote. In the second strategy, the adversary nullifies vote v_i by injecting energy on both subcarriers assigned to u_i.

Strategy 1: In the first strategy, the adversary A emulates the voter behavior by injecting energy to either $f^0_{u_i}$ or $f^1_{u_i}$. Let A target the casting of $v_i = 0$. To successfully cast v_i, he can guess the subcarrier mapping with success probability 0.5, for every symbol vote. The adversary can still hope to nullify the vote of u_i (i.e., change the value of $\hat{v}_i(n)$ from $\hat{v}_i(n) = v_i$ to $\hat{v}_i(n) = e$). According to (5), to nullify $\hat{v}(i)$, all symbol votes $\hat{v}_i(n_0), \hat{v}_i(n_0+1), \ldots, \hat{v}_i(n+0+\ell-1)$ must be nullified. This is equivalent to guessing the subcarrier index used by u_i to cast each of the ℓ symbol votes. As the subcarrier carrying each symbol vote is selected pseudo-randomly and independently per symbol vote, the probability of nullifying $\hat{v}_i$ becomes:

$$\begin{aligned}\Pr[\hat{v}(i) = e] &= \Pr[\hat{v}_i(n_0) = e, \ldots, \hat{v}_i(n_0 + \ell - 1) = e] \\ &= 0.5^{\ell}. \quad (12)\end{aligned}$$

Note that eq. (12) is true even if the value of $\hat{v}_i$ is known a priori because the index of the subcarrier carrying $\hat{v}_i(n)$ is XORed with $r_i(n)$ (see eq. (2)). From (12), we can select ℓ to drive $\Pr[\hat{v}(i) = e]$ to any desired level.

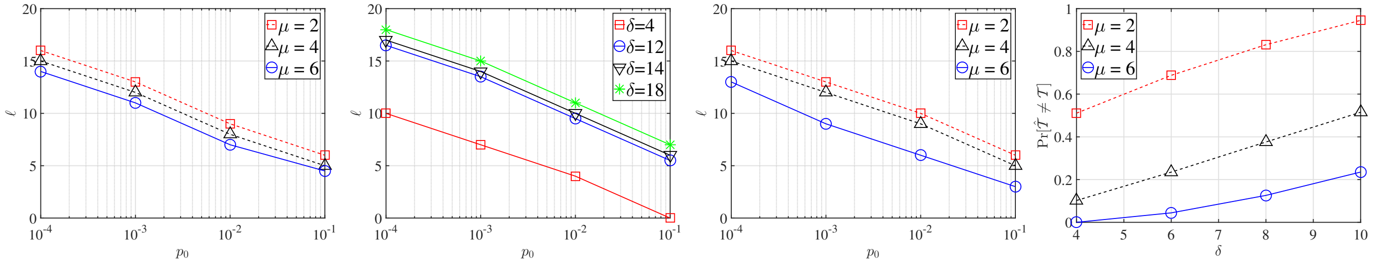
Modifying the Voting Outcome: We now analyze the probability of modifying the voting outcome under the secret vote model and the open vote model stated in Section II-E.

Proposition 2: Let an internal adversary attempt to nullify the votes of δ participants and let $p = \Pr[v(i) = e]$ denote the probability of nullifying a singe vote, as given by (12). Under the secret vote model, an internal adversary following Strategy 1 can flip the voting outcome for a decision threshold γ and a margin μ with probability

$$\begin{aligned}\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] &= \sum_{i=\mu-\gamma}^{\delta} HG(n_1, M, i, \delta) \\ &\quad \sum_{z=\mu-\gamma}^{i} \sum_{x=z}^{\min\{i, \frac{\delta+z}{2}\}} \frac{\binom{i}{x}\binom{\delta-i}{x-z}}{\binom{\delta}{2x-z}} B(2x-z, \delta, p),\end{aligned}$$

where $n_1 = \frac{M+\mu}{2}$ denotes the number of votes in favor of $\mathcal{T}$.

Proof: The proof is provided in Appendix C. ■



(a) Strategy 1 (b) Strategy 1 (c) Strategy 2 (d) Strategy 2

Fig. 4: Minimum number of symbol votes ℓ to guarantee robustness p_0 (a) under the secret vote model for Strategy 1, (b) for $\mu = 4$ and various δ, under the secret vote model and for Strategy 1, (c) under the open vote model for Strategy 2, (d) under the secret vote model for Strategy 2.

Proposition 3: Under the open vote model, an internal adversary following Strategy 1 can flip the voting outcome for a decision threshold γ and a margin μ with probability

$$\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] = \sum_{i=\mu-\gamma}^{\delta} B(i, \delta, p). \tag{13}$$

where δ denotes the number of votes that the adversary attempts to nullify, with $\delta \leq n_1$.

Proof: The proof is provided in Appendix D. ■

Selecting the Security Parameter ℓ: Propositions 2 and 3 allow us to select the number of symbol votes ℓ to guarantee robustness with a desired probability. Suppose we want to limit $\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] \leq p_0$. Then, we can select ℓ to guarantee p_0, as shown in Corollaries 2 and 3.

Corollary 2: For the secret vote model, $\Pr\left[\hat{\mathcal{T}} \neq \mathcal{T}\right] \leq p_0$ if

$$\ell > \left\lceil \frac{1}{\log 2} \log \frac{\delta \sum_{i=\mu-\gamma}^{\delta} HG(n_1, N, i, \delta) \sum_{z=\mu-\gamma}^{i} \frac{1}{z}}{p_0} \right\rceil.$$

Proof: The proof is provided in Appendix E. ■

Corollary 3: For the open vote model, $\Pr\left[\hat{\mathcal{T}} \neq \mathcal{T}\right] \leq p_0$ if

$$\ell \geq \left\lceil \frac{1}{\log 2} \log \frac{n_1}{(\mu - \gamma) p_0} \right\rceil.$$

Proof: The proof is provided in Appendix F. ■

From Corollaries 2 and 3, we observe that the required number of symbol votes ℓ drops linearly with the logarithm of p_0. This is also attested by the plots in Fig. 4, which show the required ℓ as a function of p_0, for various margins μ and number of attacked votes δ (to demonstrate the linear relationship of ℓ with the logarithm of p_0, the ceiling function has not been applied). In Fig. 4, a total of 20 participants were considered and the voting threshold γ was set to zero (plurality rule). Finally, δ was set to the number of positive votes.

Fig. 4(a) considers the secret vote model under Strategy 1. As μ increases, fewer symbol votes are necessary to provide the same robustness. However, without knowing the vote intend, the adversary nullifies both "yes" and "no" votes, thus making it harder to close the vote margin. In Fig. 4(b), we plot ℓ as a function of p_0 for different δ and for $\mu = 4$ under the secret vote model. If few votes are attacked (small δ), the achieved robustness is high for relatively small ℓ. When δ increases, a larger ℓ is needed to achieve the same robustness. However, the adversary's gains diminish beyond a certain δ. As more "yes" votes are initially corrupted, the number of remaining "yes" and "no" votes is balanced, thus becoming equally likely to nullify votes of both types with the increase of δ. Such nullification does not close the voting margin. Fig. 4(c) considers the open vote model under Strategy 1. Comparing to Fig. 4(a), we observe that a higher ℓ is necessary to provide the same level of robustness when compared to the secret vote model. This is because the adversary only attacks participants that intend to cast votes in favor of $\mathcal{T}$.

Strategy 2: In the second strategy, the adversary injects energy on both subcarriers assigned to a targeted participant to nullify the participant's vote with certainty. This strategy comes at the expense of increased presence (many subcarriers are attacked). The probability of flipping the voting outcome with Strategy 2 is expressed in Propositions 4 and 5 for the secret and the open vote models, respectively.

Proposition 4: Under the secret vote model, an internal adversary following Strategy 2 can flip the voting outcome for a decision threshold γ and a margin μ with probability

$$\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] = \sum_{z=\mu-\gamma}^{\min\{n_1,\delta\}} \sum_{x=\lceil \frac{\delta+z}{2} \rceil}^{\min\{n_1,\delta\}} \frac{\binom{n_1}{x}\binom{n_2}{\delta-x}}{\binom{M}{\delta}}, \tag{14}$$

when attempting to nullify δ votes.

Proof: The proof is provided in Appendix G. ■

Proposition 5: Under the open vote model, an internal adversary following Strategy 2 can flip the voting outcome for a decision threshold γ and a margin μ with certainty, or $\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] = 1$, when injecting energy in $J \geq \mu - \gamma$ subcarriers.

Proof: The proof is provided in Appendix H. ■

Fig. 4(d) shows the probability of flipping the voting outcome as a function of J for various μ, under the secret vote model. This probability decreases with μ because the adversary must nullify more in favor votes. Moreover, it increases with J. Note that the probability of flipping the voting outcome no longer depends on the security parameter ℓ. This is because the adversary injects energy over both subcarriers assigned to a targeted participants, and therefore nullifies the targeted vote with certainty, irrespective of ℓ.

When Strategy 2 is employed under the open vote model, the voting outcome can be flipped with certainty by attacking a number of votes equal to the vote margin. This is because the energy injection is limited to the subcarriers of participants that intend to cast in favor votes. Nullifying μ of those votes is sufficient to close the voting margin.

Subcarrier sequence preloading: To cope with an internal adversary following Strategy 2, we design a method for concealing the subcarrier assignment between participants. Without knowledge of the subcarriers assigned to others, an internal adversary becomes equivalent to an external one. He can only blindly inject energy on various subcarriers hoping to nullify in favor votes and flip the voting outcome. To hide the subcarriers used by each participant, we modify the setup and vote casting phases as follows.

Setup Phase: In the setup phase, the administrator preloads relevant quantities to the participants and the tallier.
Key generation: The administrator generates keys K_{perm} and $K_{vote,i}$ as described in Section III-A.
Key assignment: The administrator preloads $K_{vote,i}$ to each participant u_i. The administrator preloads $K_{vote,i}$ and K_{perm} to the tallier.
Subcarrier sequence preloading: The administrator computes the subcarrier assignment for each participant u_i by applying pseudo-random function

$$\Pi_F : \{0,1\}^\tau \times [1, N] \times \mathbb{Z}^+ \rightarrow [1..N],$$

to map subcarrier with index p during slot n, to subcarrier $\Pi_F(K_{perm}, p, n)$. For each participant, it computes

$$F_{u_i} = \{(f^0_{u_i}(1), f^1_{u_i}(1)), (f^0_{u_i}(2), f^1_{u_i}(2)), \dots, (f^0_{u_i}(n), f^1_{u_i}(n)\}$$

where, $f^0_{u_i}(j) = f_{\Pi_F(K_{perm},(2i-1),j)}$, and $f^1_{u_i}(j) = f_{\Pi_F(K_{perm},2i,j)}$. Sequence F_{u_i} is preloaded to participant u_i[1].

Vote Casting Phase: The vote casting phase remains the same as in Section III-C, with the exception of skipping the subcarrier assignment step. By preloading the subcarrier sequence at each participant, an internal adversary u_i cannot infer the subcarrier assignment of any other participant. The adversary is only aware of his own sequence F_{u_i}. Without access to K_{perm}, the adversary can only select the subcarriers where energy is injected at random. In this case, the robustness of PHYVOS under an internal adversary model becomes equivalent to the robustness of PHYVOS under an external adversary, as it is analyzed in Section IV-A. Note that a formula adjustment is needed in Proposition 1 to account for the reduction in the number of subcarriers unknown to the adversary. Since u_i is aware of his own subcarrier assignment, it selects to inject energy to J out of the remaining $N-2$ subcarriers (as opposed to J out of N as stated in Proposition 1). Nevertheless, the robustness computation follows along the same lines as in Proposition 1 and therefore, it is omitted.

The subcarrier sequence preloading comes at the expense of extra storage at each participant, which is linear to the number of voting rounds. The storage required to support $\mathcal{L}$ voting rounds with ℓ symbol votes per round is equal to $2\lceil\log_2 N\rceil\ell\mathcal{L}$ bits (each voting round consists of ℓ symbol votes casted in one of the two subcarriers indexed by $2\lceil\log_2 N\rceil$ bits). For example, a sequence of 80 Kbytes would support 10^5 voting rounds over 64 subcarriers.

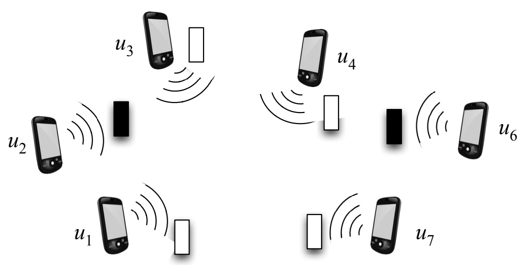


Fig. 5: The PHYVOS distributed voting scheme. Wireless devices cast their votes to each other using orthogonal subcarriers. Each participant tallies all votes and computes the voting outcome.

V. Voting Without a Centralized Tallier

In this section, we design an implementation of PHYVOS without a centralized tallier. The scenario is depicted in Fig. 5. A set of six participants co-located within the same collision domain cast their votes. Each participant acts as a tallier by independently tallying the votes casted by other participants and computing the voting outcome. All participants end up with the same voting outcome estimate $\hat{\mathcal{T}}$. To maintain the parallel nature of our PHY-layer voting technique, participants must be capable of simultaneously cast votes and performing the tallying operation. This entails the simultaneous transmission and reception over the OFDM band, that is the operation of each participant in full duplex (FD) mode. We outline two transceiver solutions that enable this concurrent transmission and reception. The first solution exploits self-interference cancellation (SIC) techniques to enable the FD mode. The second solution explores principles similar to OFDMA to allow for the simultaneous vote casting from multiple participants

A. Full Duplex OFDM

Recent advances on SIC techniques have shown that it is feasible to transmit and receive over the same frequency band [6], [28]. This is achieved by suppressing a significant portion of self interference, using a combination of antenna-based SIC, signal inversion, and RF/digital interference cancellation. In these techniques, the transmitted signal is subtracted from the received signal such that the former does not occupy the dynamic range of the ADC, allowing for the decoding of the incoming signal. For OFDM systems, FD can be realized by independently reducing self-interference at each subcarrier using narrowband cancellation techniques [28], [33].

The operating characteristics of PHYVOS, make the adoption of SIC based FD OFDM easier than its use for the communication of messages. First, each transmitter injects a signal on a single subcarrier, leaving the rest of the subcarriers empty. Thus, the self-interference in other subcarriers is small and primarily limited to the adjacent subcarriers. Applying SIC on the specific subcarrier used to cast a vote further reduces the interference on other subcarriers. Moreover, no signal decoding is necessary. Determination of votes is performed by detecting energy at the output of the FFT block. An imperfect cancellation at subcarrier $f^j_{u_i}$ used by a participant u_i to cast a vote v_i does not affect the tallying of v_i at u_i. Participant u_i is already aware of his own voting intend and does need to decode the symbol transmitted on $f^j_{u_i}$ to determine v_i.

[1] If preloading is not possible, the sequence F_{u_i} can be generated by the tallier that stores K_{perm}. The tallier can securely communicate F_{u_i} to a participant using $K_{vote,i}$.

B. OFDMA

If participants are not equipped with SIC-capable transceivers, FD operation can be achieved by applying OFDMA. Assuming that the transceivers can concurrently operate their transmission and reception radio chains, they can rely on frequency separation to enable the simultaneous vote transmission and reception. Using the adjacent subcarrier method (ASM) [3], participants can form subchannels from adjacent subcarriers so that additional frequency separation is created. In particular, each subchannel consists of three adjacent subcarriers. To cast a vote on a subchannel, energy is injected on the middle subcarrier, using the adjacent subcarriers as guards. Although this approach limits the spectral efficiency of OFDM by essentially converting it to a FDD system, it still provides significant delay reduction for PHY-layer voting relative to message-based voting.

C. Decentralized PHYVOS

Similar to the centralized tallier scenario, the decentralized PHYVOS consists of four phases: the setup phase, the vote request phase, the vote casting phase, and the tallying phase.

Setup Phase: In the setup phase, the administrator initializes all M participants by preloading K_{perm} to each participant. Note that the pairwise keys $K_{vote,i}$ used for sharing a pairwise secret random sequence between each voter and the tallier are no longer used. The sequences were applied to each symbol vote to conceal the vote-to-subcarrier mapping from internal adversaries (Step 3 of the vote casting phase). When the tallier is replicated at every participant, all sequences R_i must be disclosed to participants, thus negating their security function.

Vote Request Phase: The vote request phase follows the same steps described in Section III-B.

Vote Casting Phase: In the vote casting phase, participants cast and receive votes simultaneously using FD-OFDM. Each vote v_i consists of a series of ℓ symbol votes. The vote casting steps are as follows:

Subcarrier assignment: The subcarrier assignment is performed in the same manner as in Section III-C.

Vote casting: Let voting casting be initiated at slot n_0. To cast a vote $v_i \in \{0, 1\}$, a participant u_i generates ℓ symbol votes $v_i(n_0) = v_i(n_0 + 1) = \ldots = v_i(n_0 + \ell - 1) = v_i$. Each $v_i(n)$ is represented by an OFDM symbol with the following values per subcarrier

$$x_k(n) = \begin{cases} \alpha_y, & f_{u_i}^{v_i(n)}(n) \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

where α_y is a randomly selected modulation symbol and $n_0 \leq n < n_0 + \ell$. Note that the placement of energy of either $f^0_{u_i}(n)$ or $f^1_{u_i}(n)$ is solely based on the value of $v_i(n)$.

Vote Tallying Phase: In the vote tallying phase, each participant u_i individually computes the votes of other participants by applying Steps 1-4 outlined in Section III-D. The only difference is in the application of Step 2 for extracting symbol votes. Eq. (4) is modified as follows to omit the XORing of the symbol votes with the pseudo-random binary sequence.

$$\hat{v}_i(n) = \begin{cases} 0, & \text{if } p^0_{u_i}(n) > \gamma_D, \ \ p^1_{u_i}(n) \leq \gamma_D \\ 1, & \text{if } p^0_{u_i}(n) \leq \gamma_D, \ \ p^1_{u_i}(n) > \gamma_D \\ e, & \text{otherwise.} \end{cases} \tag{16}$$

D. Security Analysis

In this section, we briefly sketch the robustness of PHYVOS with a decentralized tallier under an external and internal adversary model.

External adversary: An external adversary is unaware of the cryptographic key K_{perm} used to permute the subcarrier assignment per symbol vote. Therefore, his best strategy is to inject energy on randomly selected subcarriers. Let the adversary inject energy on J subcarriers, as in the case of centralized PHYVOS. Consider the tallying operation occurring at participant u_i. By injecting energy on J subcarriers, the adversary can potentially impact any vote but v_i, because v_i is known to u_i a priori. If v_i is in favor of the voting outcome $\mathcal{T}$, the adversary has to successfully nullify $\mu - \gamma$ votes excluding v_i in order to flip $\hat{\mathcal{T}}$. This probability is given by Proposition 1 by adjusting the number of in-favor votes that can be nullified to $n_1 = \frac{M+\mu}{2} - 1$. If v_i is against the voting outcome $\mathcal{T}$, the probability of flipping the voting outcome is given by Proposition 1, without adjusting n_1.

Internal adversary: An internal adversary is aware of cryptographic key K_{perm} used by each participant for generating its subcarrier assignment. This allows the adversary to identify the subcarriers used by specific participants to cast votes. Moreover, the subcarrier-to-vote mapping is known because it is no longer randomized by the pairwise secret sequences R_i. The application of these sequences is no longer effective because every participant must be aware of them to correctly tally votes. With full knowledge of the subcarrier assignment, flipping the voting outcome can be achieved by nullifying $\mu - \gamma$ in-favor votes by targeting exactly $\mu - \gamma$ subcarriers.

Although the tally modification cannot be prevented, it is easily detectable by legitimate participants. In-favor participants can determine that their votes are nullified by detecting energy on the opposite subcarrier from the active one. Moreover, the number of nullified votes received by each participant (tallier) is indicative of an ongoing tally modification. In this case, the voting results can be invalidated.

VI. Voting Overhead

In this section, we compare the voting delay of PHYVOS with the voting delay of message-based voting. Suppose a popular OFDM-based protocol such as 802.11g is used for message-based voting (MV). Each 802.11g packet consists of a 20 μsec preamble (5 OFDM symbols), a 30-byte MAC header and a 4-byte CRC code. Moreover, the vote integrity is protected by a message authentication code based on a secure hash function such as SHA-256 [30]. The message digest size for SHA-256 is 32 bytes. Assuming the highest possible transmission rate for 802.11g, each OFDM symbol

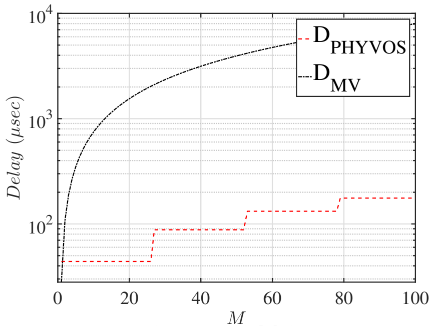


Fig. 6: Voting overhead as a function of M for message-based voting (MV) and PHYVOS.

can carry 6 bits per subcarrier, times 48 data subcarriers = 36 bytes. Therefore, one vote can be transmitted in 7 OFDM symbols. Ignoring any contention for capturing the wireless medium, participants must wait at least a DCF interframe space (DIFS) between transmitting messages. For 802.11g, DIFS = 13 OFDM symbols. The total delay required to cast M votes becomes

$$D_{MV} = 20M - 13 \text{ OFDM symbols.} \tag{17}$$

In PHYVOS, up to 26 participants can simultaneously cast their votes using ℓ OFDM symbols (for 52 subcarriers and no pilots). For $M > 26$, a second voting round is required. The value of ℓ is based on the analysis presented in Section IV. For our comparison, we set $\ell = 11$ symbols, which yields a robustness level of 10^{-3} (we note that this is an online attack, without any opportunity for repeated trials. Therefore, a robustness of 10^{-3} is acceptable). The total delay required to cast M votes becomes,

$$D_{PHYVOS} = \left\lceil \frac{M}{26} \right\rceil \ell \text{ OFDM symbols.} \tag{18}$$

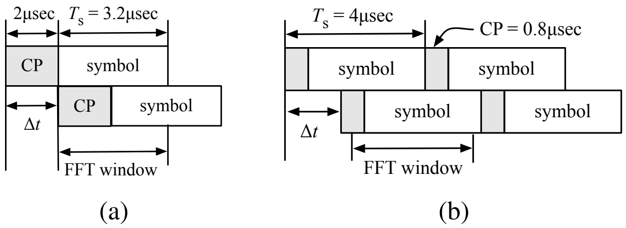
Figure 6 shows the voting delay as a function of the number of participants M, assuming a typical OFDM symbol duration of 4μsec. PHYVOS reduces delay by one order of magnitude for $M = 11$ and two orders of magnitude for $M = 50$. Note that for $M = 26$, the MV incurs a delay of at least 2 sec.

We note that in most modern OFDM systems the number of available subcarriers could be substantially higher than 52. For instance, the number of subcarriers in LTE exceeds 300 and can reach up to 1,200 when the allocated bandwidth is 20 MHz. Therefore, a much larger number of participants can be simultaneously supported, although we do not anticipate that this number will be large for one-hop scenarios. In the event that multiple rounds are needed to accommodate the number of voting participants, the individual delay until a each participants casts its vote does not affect the voting delay, which is defined as the delay until all votes are casted. If an application requires rectifying the unfairness in the individual voting delay, a round robin approach can be used to alternate between voting groups on every voting round.

VII. Practical Considerations and Implementation

A. Frequency Synchronization

Radio oscillators do not operate at the same nominal frequency due to manufacturing imperfections. This frequency misalignment is known as carrier frequency offset (CFO). OFDM systems are particularly sensitive to CFO due to the subcarrier orthogonality requirement. The CFO has two critical effects on demodulation. First, subcarriers are no longer orthogonal causing inter-carrier interference (ICI) and reducing the SNR. Second, symbols at each subcarrier appear arbitrary rotated in the constellation. Finally, a large CFO can cause a subcarrier shift at the receiver, whereby a symbol transmitted over subcarrier f_i is mapped to f_j. This shift occurs if the CFO is larger than the subcarrier spacing [25], [26].



(a) (b)

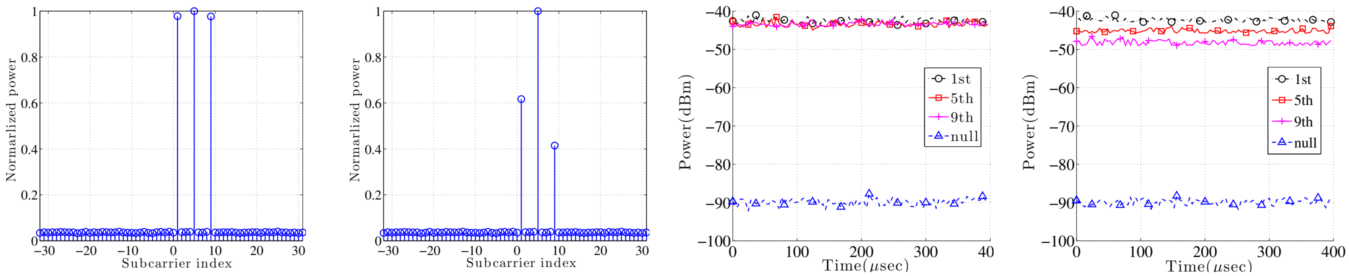
Fig. 7: (a) Increasing the CP, (b) casting a symbol vote in two symbol durations.

To mitigate the impact of CFO in practical systems, receivers estimate the CFO using the preamble transmitted with every packet. In PHYVOS, no preamble is present with the transmission of votes to save on messaging overhead. However, the lack of frequency synchronization does not impact the correct vote estimation, because *no demodulation is performed.* Any symbol rotation in the constellation map does not affect the energy estimation on a given subcarrier. After all, the symbol transmitted to realize a vote is selected at random and does not convey any information. Furthermore, for a CFO that does not cause a subcarrier bin shift, the strongest ICI component comes from adjacent subcarriers. To limit ICI, the subcarriers assigned to each participant can be spaced as far as the number of participants allows. For instance, for 10 voters and 64 subcarriers, every 3rd subcarrier is used to cast a vote.

B. Time Synchronization

Another practical problem for PHYVOS is that symbol votes do not reach the tallier perfectly synchronized. Differences in propagation delay and device clock drifts can cause a time misalignment between the symbol votes casted by each device. This misalignment will affect the set of samples that fall within the FFT window of the Fourier transform applied at the receiver for extracting the spectral components of the OFDM signal. This is similar to *symbol bleeding* caused in OFDM systems when delayed copies of OFDM symbols arrive at the receiver due to multipath effects. The solution applied in OFDM is to append a cyclic prefix (CP) to every symbol, which is in the order of 0.8 μsec.

For PHYVOS, the time misalignment Δt between symbols at the receiver can be greater than 0.8 μsec. For a typical WiFi range of 300m, the propagation delay difference between two devices can by up to 1μsec. Moreover, the typical clock error for modern clocks is well below 5ppm [18]. If clock synchronization is performed every 100msec (typical beacon transmission period for WiFi base stations), the expected clock error between two devices can be up to 1μsec, making the total time misalignment $\Delta t \leq 2\mu$sec.



(a) Topology A (b) Topology B (c) Topology A (d) Topology B

Fig. 8: (a), (b) Normalized average received power per subcarrier, (c), (d) received power per subcarrier as a function of time.

To cope with the symbol misalignment, we can extend the CP duration to 2μsec to account for the maximum expected Δt. The increase in CP comes at the expense of a higher overhead to cast a symbol vote (5.2 μsec vs. 4 μsec). Note that the increased CP duration is adopted only for vote casting and is not part of the normal OFDM operation for data transmissions. Alternatively, to maintain compatibility with the current OFDM specifications, we can extend the symbol vote duration to two OFDM symbols, without increasing the CP duration. This solution comes at the expense of doubling the overhead for casting a symbol vote. A similar solution was adopted in [9]. The two solutions are shown in Fig. 7.

C. PHYVOS Implementation

Testbed setup: We implemented PHYVOS on NI USRPs 2921 devices, operating in the 2.4 GHz band over a 39.6 MHz spectrum. A total of four radios were at our disposal. Under normal operation, three radios operated as voters, whereas one radio operated as the tallier. One radio was switched to an attacker role for adversarial scenarios. Voter radios were placed in a LoS configuration at varying distances from the tallier within an office environment. We divided the 39.6 MHz spectrum to 64 subcarriers. To cast a symbol vote, each radio used BPSK modulation to transmit a random symbol at the designated subcarrier. The CP value was set to 0.8 μsec, as the time synchronization error between the different radios was relatively small. We used a 64-point FFT to collect the symbol votes from each subcarrier. The transmission power of each radio was set to 20 dBm (0.1 W).

Selection of threshold γ_D: In the first experiment, we investigated the selection of the power threshold γ_D used in eq. (4) for detecting votes. We assigned the 1st, 5th, and 9th subcarrier to each of the three voter radios. Each voter casted 1,000 symbol votes at its designated subcarrier by transmitting 1,000 BPSK symbols. The rest of the subcarriers remained null. A time gap of100 msec was imposed between two consecutive votes. Fig. 8(a) shows the normalized magnitude of the FFT output at the tallier, averaged over the 1,000 transmitted symbols when the three voters are placed 5ft away from the tallier (topology A). Fig. 8(b) shows the same results when the three voters are at 5ft, 10ft, and 15ft away from the tallier (topology B).

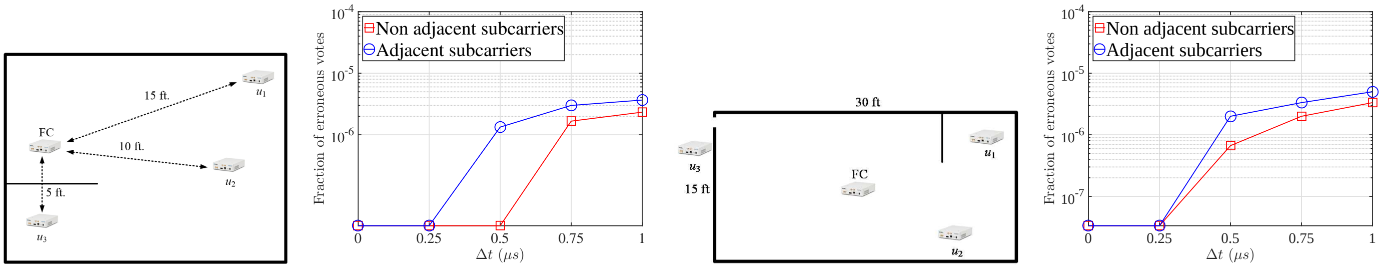
Fig. 8(c) and 8(d) show the received power as a function of time for 100 consecutive symbols. For topology A, the power of active subcarriers is approximately -42dBm, whereas the power of null subcarriers is -90dBm. The recorded -90dBm value for the null subcarriers is well above the noise floor due to the operation of nearby devices over the ISM band. For topology B, the received power from the farthest radio dropped to -49dBm. Based on the recorded values, we set the threshold γ_D for the detection of a symbol vote to -80dBm, which is well above the receiver sensitivity.

Time synchronization: In the second experiment, we studied the effect of time synchronization on the correct operation of PHYVOS. The experimental setup is shown in Fig. 9(a). We used one USRP as the FC, while three USRPs were setup as voting participants. We set the CP value to 2.0μsec, the FFT window to 1.2μsec, and varied the time synchronization error between the participants. This was achieved by adjusting the firing times of the USRP devices for symbol transmissions, while the USRPs were placed at different distances from the FC. The three participants u_1, u_2, u_3 were placed as follows: u_1 was placed at 15ft from the FC with a LoS channel, u_2 was placed at 10ft from the FC with a LoS channel, whereas u_3 was placed at 5ft from the FC, but with an obstruction on the LoS path. This created different profiles of synchronization offset for different users due to multipath and also clock errors. For each synchronization offset (Δt), we transmitted 10^6 votes.

Further, we performed the experiment for two subcarrier allocations. In the first allocation, the USRP devices were assigned non-adjacent subcarriers, (1,2), (9,10), and (15,16) for submitting yes/no votes. In the second allocation, USRPs were assigned adjacent subcarriers (1,2), (3,4), and (5,6). In Fig. 9(b), we show the fraction of erroneously received votes as a function of the maximum synchronization error Δt between any two devices. We note that as long as the CP duration is larger than Δt, votes are correctly inferred despite the symbol misalignment. The scenario with non-adjacent subcarriers achieves slightly better performance, as the sample misalignment does not impact adjacent votes.

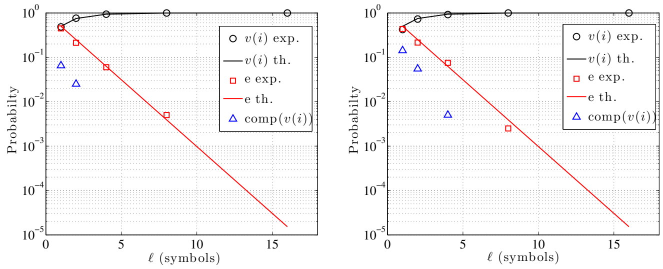
We repeated the above experiment for the topology of Fig. 9(c), where u_3 was placed on the outside of the room that housed the FC, thus obstructing the LoS path. In Fig 9(d), we show the fraction of lost votes as a function of Δt. We observed similar results to the performance under the topology of Fig. 9(a), indicating that the use of a longer CP alleviates the misynchronization phenomenon even for NLoS channels.

Voting in the presence of an internal adversary: In the third experiment, we implemented Strategy 1 for an internal



(a) (b) (c) (d)

Fig. 9: (a) The USRP topology used to evaluate the effect of time synchronization, (b) fraction of erroneously decoded votes at the receiver as a function of the synchronization error (Δt) between participants, c) the USRP topology used to evaluate the effect of NLoS paths, and (d) fraction of erroneously received votes as a function of the synchronization error (Δt) for the topology of Fig 10(c).



(a) (b)

Fig. 10: (a) Probability of tallying the correct vote $v(i)$, having an inconclusive vote e, or flipping the vote to comp($v(i)$), and (b) probability of tallying the correct vote $v(i)$, having an inconclusive vote e, or flipping the vote to comp($v(i)$).

adversarial. One of the three USRPs was assigned the role of an internal attacker that is aware of the subcarrier assignment to other voters. Voter #1 was assigned the 1st and 2nd subcarrier while voter #2 was assigned the 5th and 6th subcarrier. For each symbol vote, the attacker randomly selected one subcarrier per voter and injected a random symbol in order to nullify or flip the casted vote (Strategy 1). The experiment lasted for 10^6 symbol votes. Fig. 10(a) shows the probability of tallying the correct vote $v(i)$, having an inconclusive vote e, or flipping the vote to comp($v(i)$), as a function of the security parameter ℓ for topologies A and B. The theoretical values for tallying the correct vote $v(i)$, and having an inconclusive vote e are also shown (solid lines). The theoretical values are computed according to equation (12).

We observe that the experimental values are in close agreement with the theoretical ones. As expected, the probability of tallying the correct vote rapidly converges to one with the increase of ℓ, whereas the probability of an inconclusive vote becomes small (zero for $\ell > 8$). In our experiments, some votes were actually flipped indicating a drop in the received power on a designated subcarrier to a value smaller than γ_D for ℓ consecutive symbol votes. However, this occurred with very low probability and was not observed at all when $\ell > 2$. The results were similar for topology B (see Fig. 10(b)), with a slight increase in the probability of flipping a vote. This was primarily observed due to the near-far effect for the most distant voter (placed at 15ft from the tallier).

D. Simulated Experiments

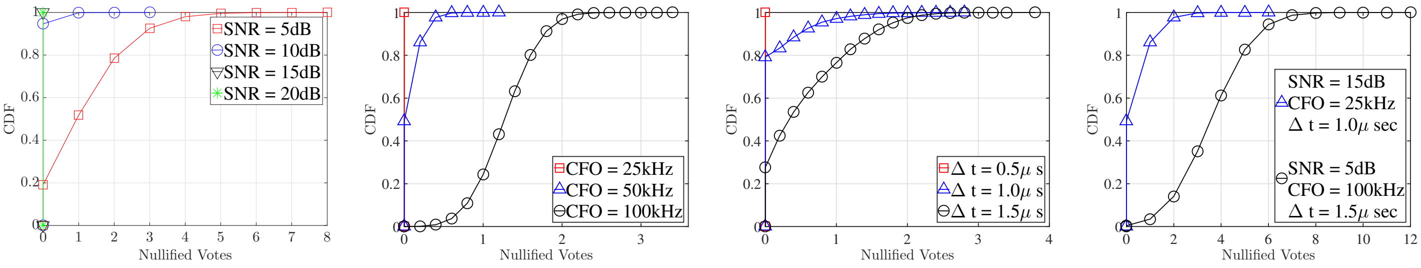
The USRP experiments involved a small number of devices and were primarily used to study the implementation nuances of simultaneous vote casting. In this section, we perform simulated voting experiments with a large number of participants.

Simulation setup: We simulated PHYVOS using MATLAB R2015B [22]. We initially considered 26 participants casting votes over 52 subcarriers to a FC. We repeated some experiments for 100 participants. The wireless channel between the tallier and each participant was simulated by a Rician fading model with maximum path delay 1.5×10^{-6} sec, a *K-factor* equal to two, and a LoS SNR equal to 15dB. The Rician channel was selected because it is representative in many one-hop topologies. To cast a symbol vote, participants randomly selected a QPSK symbol. The symbol vote detection threshold γ_D was set to -80 dBm. A plurality vote criterion ($\gamma = 0$) was applied to compute the voting outcome.

Vote nullification due to channel imperfections: In the first set of experiments, we measured the probability of unintentional vote nullification due to wireless channel imperfections. In the absence of an adversary, we varied the SNR of the participant-tallier channel and measured the number of nullified votes at the tallier. Each vote consisted of three symbol votes. Fig. 11(a) shows the CDF of the nullified votes for different SNRs. We observe that even at low SNR values ($\leq$ 10 dB), less than four out of the 26 votes are nullified due to fading, with probability over 95%.

We also measured the number of unintentionally nullified votes due to CFO and time offsets. These effects are discussed in Section VII-A and VII-B. Each participant was randomly assigned a CFO of either 0 KHz or CFO$_{\max}$. We opted to combine participants with and without CFO to allow the maximum frequency misalignment between certain subcarriers at the tallier. Fig. 11(b) shows the CDF of the nullified votes for various CFO$_{\max}$. For typical CFO values, less than two out of 26 votes are nullified in 95% of the observed runs. Vote nullification occurs when a sufficient amount of energy is leaked to adjacent subcarriers due to the CFO. The effect of the CFO can be mitigated if the tallier compensates for it before vote tallying. The tallier can record the CFO of each participant using the preambles of prior packet transmissions. Note that CFO estimation must be repeated infrequently, as it varies very slowly with time.

Furthermore, we simulated the impact of time synchronization errors caused by the misalignment of symbol votes due to time offsets (Δt). Each participant was assigned a time offset of either 0 μsec or $\Delta t_{\max}$ μsec at random.



(a) (b) (c) (d)

Fig. 11: (a) CDF plot of number of nullified votes due to wireless channel noise for varying channel SNR, (b) CDF plot of number of nullified votes received due to carrier frequency offset error for varying CFO, (c) CDF plot of number of nullified votes received due to synchronization error for varying time offset, and (d) CDF of the nullified votes when the fading, CFO, and time misalignment phenomena are all combined in the same experiment.

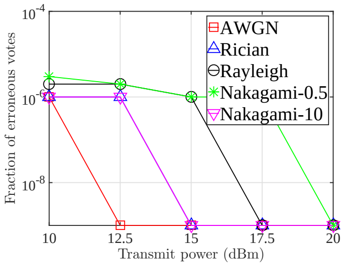
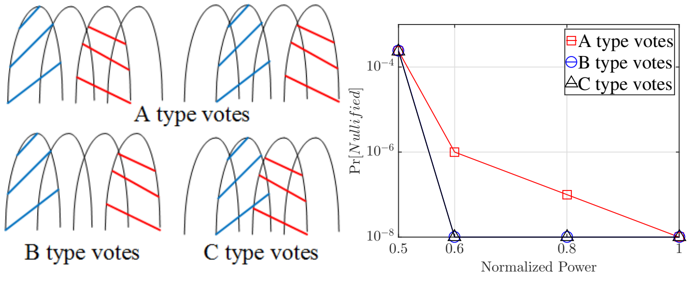


Fig. 12: Fraction of incorrectly received votes as a function of the transmission power for various wireless channel models.



(a) (b)

Fig. 13: (a) Energy assignment to subcarriers for all voting combinations, and (b) probability of votes received incorrectly plotted against normalized power of votes received for all possible voting combination.

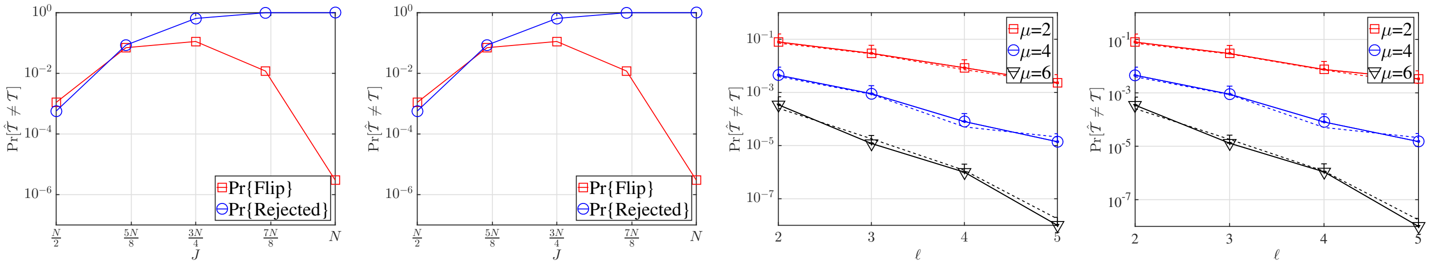
The tallier used the two symbol vote estimation technique outlined in Section VII-B to compensate for the symbol time misalignment. Fig. 11(c) shows the CDF of the nullified votes for varying $\Delta t_{\max}$. We observe that two symbols for the symbol vote estimation eliminates the impact of misalignment.

In Fig. 11(d), we show the CDF of the nullified votes when fading, CFO, and time misalignment are all present in the same experiment. We observe that under typical values (SNR = 15dB, CFO = 25kHz and Δt = 1 μsec), less than one votes are nullified, on average, with probability over 95%. In worse conditions (SNR = 5dB, CFO = 100kHz and Δt = 1.5 μsec), less than six votes are nullified with probability over 95%. This CDF shift is primarily due to the low SNR. We use Fig. 11(d) to set the threshold γ_{null} to six votes. Recall that γ_{null} is used to detect the presence of an adversary if an unusual number of votes are nullified at the tallier.

Finally, we performed a simulated experiment to evaluate the effects of various channel models on vote correctness. We measured the number of erroneously received votes at the FC for an AWGN channel, a Rayleigh channel with a maximum path delay of $1.5\mu sec$, a Rician channel with a K factor of 2, and a maximum path delay of $1.5\mu sec$ and a Nakagami-m channel with fading factors 0.5 and 10. A total of 10^6 votes per participant were transmitted. In Fig. 12, we show the fraction of erroneous votes received at the FC as a function of the transmit power in dBm. It can be observed that the vote error remains below 10^{-5} for all transmit powers and it drops with the power increase. The Nakagami-m channel with a fading factor of 0.5 yields the worst performance, but the error is still quite low and does not significantly affect the energy-based vote detection.

Effect of varying participant distances: In this scenario, we placed two participants at different distances from the tallier in order to vary the received power ratio between subcarriers at the tallier. We considered the subcarrier assignments shown in Fig. 13(a). Votes of type A represent cases where two participants inject energy on subcarriers separated by a single subcarrier, votes of type B represent cases where the subcarrier separation is equal to two, whereas votes of type C represent cases where the two participants inject energy on adjacent subcarriers. Fig. 13(b) shows the probability of vote nullification for any of the two participants as a function of the power of the more distant participant, normalized over the power of the closest participant. The probability of vote nullification remains low even when the power of the distant participant is half of the power of the closest one. Votes of type A have the highest probability of being nullified, because energy from two active subcarriers "bleeds" into a common adjacent empty subcarrier. On the other hand, votes of type B and C exhibit the same probability of vote nullification, because only one active subcarrier "bleeds" into an inactive one. As the transmission powers between the participants become equal, the probability of vote flipping attains very low values.

External adversary: In the third set of experiments, we evaluated the robustness of PHYVOS against an external adversary. The adversary attempted to flip the voting outcome by injected energy to J randomly selected subcarriers. The tallier used the threshold γ_{null} to detect an ongoing attack, if a large number of votes are nullified. We also fixed the number of symbol votes to $\ell = 3$ and the voting margin to $\mu = 3$. Fig. 14(a) shows the tradeoff between probability of flipping $\mathcal{T}$ and rejecting the voting round as a function of J. As J increases, the probability of flipping the voting outcome improves for the adversary until J equals 3/4 of the available subcarriers. Any further increase of J has a negative effect. This is because the adversary nullifies votes that oppose the



(a) (b) (c) (d)

Fig. 14: (a) Probability of flipping voting outcome and rejecting a voting round as a function of the number of attacked subcarriers (J) in presence of an external adversary, and for 26 participants, (b) probability of flipping voting outcome and rejecting a voting round as a function of the number of attacked subcarriers (J) in presence of an external adversary, and for 100 participants, (c) probability of flipping the voting outcome as a function of the number of symbol votes (ℓ) in presence of an internal adversary, and for 26 participants, and (d) probability of flipping the voting outcome as a function of the number of symbol votes (ℓ) in presence of an internal adversary, and for 100 participants.

voting outcome. On the other hand, the probability of rejecting the voting round strictly increases with J. For the value of J that maximizes the probability of flipping the voting outcome ($\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] = 0.1$, for $J = \frac{3N}{4}$), the voting round is rejected with probability 94.2%.

To verify that PHYVOS is scalable, we repeated the simulated experiments for 100 participants who casted votes over 52 subcarriers. As there are only 52 subcarriers available, the participants were divided to three groups of size 26 and one group of size 22. Participants of the same group casted their votes simultaneously using ℓ symbol votes, requiring a total of 4ℓ symbols to complete a voting round. Fig. 14(b) shows the tradeoff between probability of flipping $\mathcal{T}$ and rejecting the voting round as a function of J. We observe that the increased number of participants does not qualitatively affect the robustness of PHYVOS.

Internal adversary: In the fourth set of experiments, we evaluated the robustness of PHYVOS against an internal adversary when applying Strategy 1. Using his knowledge of the subcarrier assignment, the adversary injected energy at one of the two subcarriers assigned per participant. In Fig. 14(c), we show the probability of flipping the voting outcome as a function of number of symbol votes ℓ, and for $\mu = \{2, 4, 6\}$. Solid lines indicate the values obtained via simulation, whereas dotted lines show the theoretical values calculated using (13). For the simulation results, we also plot the upper bound of the 95% confidence intervals (the lower bounds are omitted due to the log scale on the Y axis). The simulation results verify the theoretical analysis for the probability of flipping the voting outcome. Using larger values of ℓ allows the tallier to substantially reduce this probability. We repeated our experiments for 100 participants who casted their votes in groups. Fig. 14(d) shows similar results to Fig. 14(c). This is expected, as only 26 participants vote at every slot.

VIII. Related Work

The use of voting for improving reliability has been studied since the 1950s [31], with a long literature on various reliability and efficiency aspects (e.g., [4], [5], [14], [16], [17], [32]). Levitin proposed a weighted mechanism for binary voting where each vote is weighted based on the participant's identity [16]. The author showed that for participants with different decision times, a tradeoff exists between reliability and delay. He proposed an algorithm to maximize reliability under a time constraint. Barbara and Molina studied the reliability of voting mechanisms, when participants are divided into groups and are assigned a number of votes [4]. The group with the voting majority is prioritized to perform critical system operations. They proposed several vote assignment heuristics to improve the overall system reliability. Kwiat *et al.* examined three binary voting rules for fault tolerance and evaluated the resulting reliability and security [14]. They proposed a random selection algorithm for computing the voting outcome from a set of votes that contain malicious ones. We emphasize that PHYVOS implements a PHY layer vote casting mechanism that guarantees vote integrity. The voting rules (majority, random selection, number of votes per participant, vote weights, etc.), which is the subject of most previous studies in reliability and fault-tolerance, is complementary to our method.

In the context of wireless networks, voting finds wide application to data fusion, intrusion detection and secure localization in WSNs [2], [13], [15], [34], real-time coordination in multi-agent systems [8], and fault-tolerant protocols [20], [23] The de facto voting mechanism adopted in these works is message-based voting, in which votes are casted through messaging. Message-based voting also facilities the integration of security measures for preventing the manipulation of the voting outcome. Voters can be authenticated, and vote integrity can be verified using standard cryptographic primitives such as digital signatures, message authentication codes, and digital certificates [30]. Compared to message-based voting, PHYVOS requires significantly less communication overhead, without sacrificing robustness to vote manipulation.

From an implementation standpoint, the most relevant works to ours are presented in [9], [27]. In [9], Dutta *et al.* proposed SMACK, an acknowledgment scheme for implementing a reliable broadcast service. Similar to PHYVOS, SMACK exploits the subcarrier orthogonality of OFDM to allow the simultaneous submission of acknowledgements in response to a broadcast message transmitted by a single source. In [27], Rahul *et al.* proposed SourceSync, a distributed wireless architecture that explores sender diversity in OFDM. SourceSync enables the reception and *demodulation* of OFDM symbols composed of symbol transmissions over individual subcarriers by a diverse set of senders. Contrary to SMACK and PHYVOS, SourceSync can demodulate the combined

OFDM symbol and retrieve the individual data streams of each sender. This capability comes at the expense of complex symbol-level synchronization and channel estimation at the senders, performed through the transmission of preambles.

Recently, the infeasibility of erasing energy from a wireless channel was challenged. Pöpper *et al.* showed that under stable and predictable channel conditions (e.g., LOS), an attacker utilizing a pair of directional antennas for relaying the inverse of the received signal could cancel a signal at a targeted receiver [24]. Such powerful signal cancellation attacks are hard to launch in practice against PHYVOS due to the multiple wireless channels used by the participants for the simultaneous communication with the tallier. Moreover, channel estimation of any of those channel within the channel coherence time becomes difficult without the transmission of preambles.

IX. Conclusions

We presented PHYVOS, a secure and fast PHY-layer voting scheme for wireless networks. In PHYVOS, no explicit messaging is necessary. Participants cast their votes simultaneously by exploiting the subcarrier orthogonality in OFDM. PHYVOS is aimed at reducing the delay overhead for wireless applications where secure voting is time-critical. We analyzed the robustness of PHYVOS against both external and internal adversaries who aim at altering the voting outcome at the tallier. We showed that PHYVOS maintains the integrity of the voting outcome with high probability, without using cryptographic primitives. We extended PHYVOS to a decentralized operation scenario, in which participants can determine the voting outcome without the presence of a centralized tallier. We implemented PHYVOS on the USRP platform and verified the robustness properties via experimentation and simulations.

References

[1] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan. Cooperative spectrum sensing in cognitive radio networks: A survey. *Physical Comm.*, 4(1):40–62, 2011.

[2] N. Al-Nakhala, R. Riley, and T. Elfouly. Distributed algorithms in wireless sensor networks: an approach for applying binary consensus in a real testbed. *Comp. Nets.*, 2015.

[3] J. G. Andrews, A. Ghosh, and R. Muhamed. *Fundamentals of WiMAX: understanding broadband wireless networking*. Pearson Education, 2007.

[4] D. Barbara and H. Garcia-Molina. The reliability of voting mechanisms. *IEEE Trans. Computers*, 36(10):1197–1208, 1987.

[5] M. Barborak, A. Dahbura, and M. Malek. The consensus problem in fault-tolerant computing. *ACM Comp. Surveys*, 25(2):171–220, 1993.

[6] D. Bharadia, E. McMilin, and S. Katti. Full duplex radios. In *Proc. of the SIGCOMM Computer Communication Review*, pages 375–386. ACM, 2013.

[7] S. Capkun, M. Cagalj, R. Rengaswamy, I. Tsigkogiannis, J.-P. Hubaux, and M. Srivastava. Integrity codes: Message integrity protection and authentication over insecure channels. *Dependable and Secure Computing, IEEE Transactions on*, 5(4):208–223, 2008.

[8] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson. Distributed event-triggered control for multi-agent systems. *IEEE Trans. on Aut. Cntrl.*, 57(5):1291–1297, 2012.

[9] A. Dutta, D. Saha, D. Grunwald, and D. Sicker. SMACK: a SMart ACKnowledgment scheme for broadcast messages in wireless networks. *ACM SIGCOMM Comp. Comm. Rev.*, 39(4):15–26, 2009.

[10] S. Gollakota, N. Ahmed, N. Zeldovich, and D. Katabi. Secure in-band wireless pairing. In *USENIX security symposium*, pages 1–16. San Francisco, CA, USA, 2011.

[11] I. . W. Group. IEEE 802.22 WRAN standards. http://www.ieee802.org/22/, 2011.

[12] T. Jin, G. Noubir, and B. Thapa. Zero pre-shared secret key establishment in the presence of jammers. In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, pages 219–228. ACM, 2009.

[13] W. Kim, K. Mechitov, J. Choi, and S. Ham. On target tracking with binary proximity sensors. In *Proc. of the IPSN*, pages 301–308, 2005.

[14] K. Kwiat, A. Taylor, W. Zwicker, D. Hill, S. Wetzonis, and S. Ren. Analysis of binary voting algorithms for use in fault-tolerant and secure computing. In *Computer Engineering and Systems (ICCES), 2010 International Conference on*, pages 269–273. IEEE, 2010.

[15] L. Lazos and R. Poovendran. SeRLoc: robust localization for wireless sensor networks. *ACM Trans. on Sens. Nets.*, 1(1):73–100, 2005.

[16] G. Levitin. Weighted voting systems: reliability versus rapidity. *Reliability Engineering & System Safety*, 89(2):177–184, 2005.

[17] G. Levitin and A. Lisnianski. Reliability optimization for weighted voting system. *Reliability engineering & system safety*, 71(2):131–138, 2001.

[18] LitePoint. Practical manufacturing testing of 802.11 OFDM wireless devices. http://www.litepoint.com/whitepaper/Testing\%20802.11\%20OFDM\%20Wireless\%20Devices_WhitePaper.pdf, 2012.

[19] Y. Liu, P. Ning, H. Dai, and A. Liu. Randomized differential dsss: Jamming-resistant wireless broadcast communication. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[20] X. Luo, M. Dong, and Y. Huang. On distributed fault-tolerant detection in wireless sensor networks. *IEEE Trans. on Comp.*, 55(1):58–70, 2006.

[21] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In *Proc. of the MOBICOM Conf.*, pages 128–139. ACM, 2008.

[22] MATLAB. *version 8.6.0 (R2015b)*. The MathWorks Inc., Natick, Massachusetts, 2015.

[23] E. Ould-Ahmed-Vall, B. H. Ferri, and G. F. Riley. Distributed fault-tolerance for event detection using heterogeneous wireless sensor networks. *IEEE Trans. on Mob. Comp.*, 11(12):1994–2007, 2012.

[24] C. Pöpper, N. O. Tippenhauer, B. Danev, and S. Capkun. Investigation of signal and message manipulations on the wireless channel. In *Proceedings of the 16th European Conference on Research in Computer Security*, pages 40–59, 2011.

[25] H. Rahbari, M. Krunz, and L. Lazos. Security vulnerability and countermeasures of frequency offset correction in 802.11 a systems. In *INFOCOM, 2014 Proceedings IEEE*, pages 1015–1023. IEEE, 2014.

[26] H. Rahbari, M. Krunz, and L. Lazos. Swift jamming attack on frequency offset estimation: The achilles? heel of OFDM systems. *IEEE Transactions on Mobile Computing*, 15(5):1264–1278, 2016.

[27] H. Rahul, H. Hassanieh, and D. Katabi. SourceSync: a distributed wireless architecture for exploiting sender diversity. *ACM SIGCOMM Comp. Comm. Rev.*, 41(4):171–182, 2011.

[28] A. Sahai, G. Patel, and A. Sabharwal. Pushing the limits of full-duplex: Design and real-time implementation. *arXiv preprint arXiv:1107.0607*, 2011.

[29] K. Sampigethaya and R. Poovendran. A framework and taxonomy for comparison of electronic voting schemes. *Computers & Security*, 25(2):137–153, 2006.

[30] D. R. Stinson. *Cryptography: theory and practice*. CRC press, 2005.

[31] J. Von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Aut. studies*, 34:43–98, 1956.

[32] L. Wang, Z. Li, S. Ren, and K. Kwiaty. Optimal voting strategy against rational attackers. In *Risk and Security of Internet and Systems (CRiSIS), 2011 6th International Conference on*, pages 1–8. IEEE, 2011.

[33] Z. Zhan, G. Villemaud, and J.-M. Gorce. Design and evaluation of a wideband full-duplex ofdm system based on aasic. In *IEEE Personal Indoor and Mobile Radio Communications (PIMRC), 2013*, pages 68–72, 2013.

[34] M. Zhu, S. Ding, Q. Wu, R. R. Brooks, N. S. V. Rao, and S. S. Iyengar. Fusion of threshold rules for target detection in wireless sensor networks. *ACM Trans. on Sens. Nets.*, 6(2):181–187, 2010.