PHYVOS: Physical Layer Voting for Secure and Fast Cooperation

Bocan Hu, Yan Zhang, and Loukas Lazos
Dept. of Electrical and Computer Engineering, University of Arizona
Email: {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 exchanging messages with a fusion center or just between the participants. However, the communication and delay overheads of message-based voting can be prohibitive when voting is frequent. Additional overheads are incurred if voter authentication and vote integrity verification are required.**

In this paper, we propose a fast PHY-layer voting scheme called PHYVOS, which significantly reduces the voting overhead. In PHYVOS, wireless devices transmit their votes to a fusion center simultaneously, by exploiting the subcarrier orthogonality in OFDM and without explicit messaging. We show that PHYVOS is secure against attackers 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 further discuss practical implementation challenges of PHYVOS related to multi-device frequency and time synchronization. Finally, we present a prototype implementation of PHYVOS on the USRP platform.

I. Introduction

Distributed wireless networks fundamentally rely on the principle of cooperation. Network 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 cognitive radio networks (CRNs) [1]. The so called cooperative gain comes from exploiting the spatial diversity of the RF sensing operation, when sensing observations are fused. 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 consensus algorithms increase fault-tolerance at relative low cooperation overhead. In binary consensus, 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 parameter value. This rule is based on some form of majority voting, plurality or threshold, to achieve the desired level of reliability. Binary votes are casted using a messaging scheme, in which 1-bit votes are carried by messages. For wireless networks, 1-bit votes require the transmission of a preamble, a PHY header and a MAC layer header. This communication and delay overheads can be prohibitive for applications where voting is applied frequently. As an example, in CRNs, channel state observations are fused every 2 seconds [6]. For control applications in networked multi-agent systems, the consensus time requirement could be even more stringent [4].

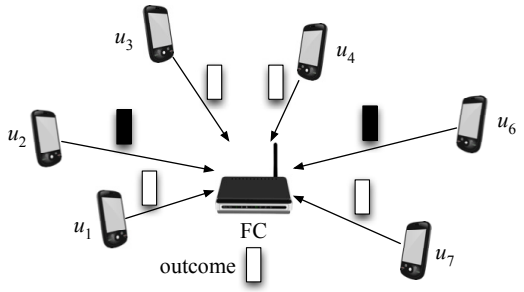


Fig. 1: The PHYVOS voting scheme. Wireless devices simultaneously cast their votes to a fusion center using orthogonal subcarriers. The fusion center tallies all votes and computes the voting outcome.

The voting delay is amplified by the application of medium access control on the wireless medium. Moreover, for applications where secure voting is required, additional overheads are incurred. To prevent an attacker from altering the voting outcome by inserting fake votes or modifying legitimate ones, cryptographic methods for user authentication and message integrity are applied. Verifying the voter authenticity and protecting the integrity of binary votes using digital signatures and message authentication codes could further increase the cooperation cost by several orders of magnitude.

To address the limitations of message-based voting, we present a secure 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 a fusion center (FC) within just a few symbols. The FC computes the voting outcome without demodulating the received signals. 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 at a single receiver (fusion center) requires intricate transmitter and receiver designs [5], [15]. 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 [15].

In this paper, we make the following contributions.

Our Contributions: We design PHYVOS, a PHY-layer majority voting scheme that reduces the cooperation cost by several orders of magnitude compared to message-based voting. In PHYVOS, participants simultaneously cast their votes by exploiting the subcarrier orthogonality of OFDM. To overcome the challenges related to decoding simultaneous transmissions from multiple senders, binary votes are casted by adding energy to designated subcarriers. Therefore, 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.

An attacker could still attempt to modify votes by inserting his own energy into various subcarriers. We reduce the probability of voting outcome manipulation by executing multiple voting rounds. Since a voting round only lasts for a few OFDM symbols, executing multiple rounds is still far more efficient than applying message-based voting. We analytically evaluate the voting robustness as a function of the number of voting rounds. We further discuss practical implementation challenges of PHYVOS related to frequency and time synchronization. Finally, we present a prototype implementation of PHYVOS on the NI USRP platform.

Paper Organization: In Section II, we briefly describe OFDM and present the system and adversary models. Section III describes PHYVOS. In Section IV, we analyze the security of PHYVOS. Practical considerations and experimental verification of PHYVOS are presented in Section VI. In Section VII, we discuss related work and conclude in Section VIII.

II. Preliminaries

OFDM System: Orthogonal Frequency Division Multiplexing (OFDM) is a multicarrier modulation method adopted by many contemporary wireless technologies (e.g., 802.11a/g/n/ac/ad, LTE, WiMax, DVB-T) due to its high spectral efficiency. The main idea of OFDM is to divide the data stream to substreams, which are independently modulated in closely separated, orthogonal subcarriers. A basic block diagram of an OFDM system is shown in Fig. 2. The data steam is fed to a serial-to-parallel (s2p) converter to generate N bit streams, where N is the number of available subcarriers for data transmission. The N streams are modulated (using BPSK, QPSK, QAM, etc.) and an N-point inverse Fourier transform (IFFT) is applied on the complex symbols. The IFFT output is fed to a parallel-to-serial (p2s) converter and further processed by a D/A converter to

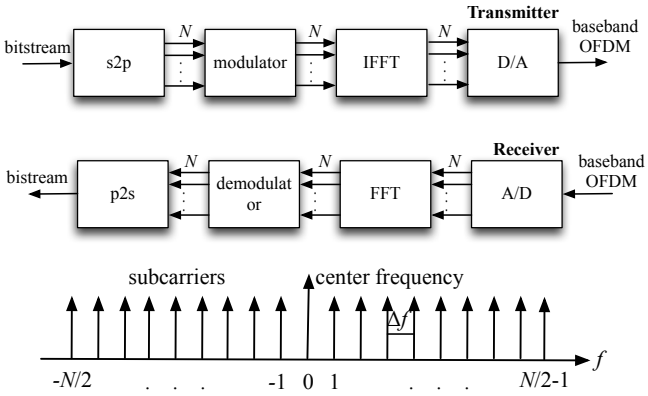


Fig. 2: Block diagram of OFDM.

compose the baseband OFDM signal. At the receiver, after the downconversion to the baseband frequency, the analog signal is digitized by the A/D converter. The Fourier transform is applied to recover the complex constellation symbols and the N substreams are combined by a p2s converter to form the original data stream. The discrete time domain representation of the baseband OFDM signal $x(n)$ is given by [12]:

$$x(n) = \sum_{k=0}^{N-1} x_k(n) * e^{\frac{j2\pi nk}{N}}, \tag{1}$$

where $x_k(n) \in \{\alpha_1, \alpha_2, \ldots, \alpha_q\}$ is the complex modulated symbol at each of the N subcarriers transmitted at time n, and $\alpha_1, \alpha_2, \ldots, \alpha_q$ are the possible modulation symbol values (q denotes the modulation order). By selecting $x_k(n)$, we can control the energy that is injected at each of the N subcarriers. This energy is detected at an OFDM receiver by passing the time domain signal through an FFT. The energy detection at each subcarrier is the basic PHY-layer function exploited by PHYVOS for implementing the voting process.

System model: A set of M participants $u_1, u_2, \ldots, u_M$ casts M binary votes $v_1, v_2, \ldots, v_M$ with $v_i \in \{0, 1\}$ to a FC (see Fig. 1). The FC tallies all votes and computes the voting outcome according to a threshold-based rule:

$$\mathcal{T} = \begin{cases} 1, & \text{if } \sum_{i=1}^{M} (-1)^{v_i} < \gamma \\ 0, & \text{if } \sum_{i=1}^{M} (-1)^{v_i} \geq \gamma. \end{cases} \tag{2}$$

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 consensus. We emphasize that we are only interested in the robustness property of voting. Well-known electronic voting requirements such as voter privacy, receipt-freeness, universal and individual verifiability, coercion-resistance [16] are beyond the context of this work. The voting robustness is defined as follows.

Definition 1 (Robustness): A cooperative voting scheme is said to be robust against active attacks and faults if the voting outcome $\mathcal{T}$ reflects the true outcome when the votes of all honest participants are tallied.

Note that the voting robustness differs from voting accuracy, in which all votes must be tallied correctly. For the former, it is sufficient to reach the correct voting outcome, even if some

votes are incorrectly tallied. The participants cast their votes to the FC using an OFDM system with N orthogonal subcarriers, denoted by $f_1, f_2, \ldots f_N$. All participants can directly reach the FC (within one hop), but could be located at varying distances from the FC. Moreover, participants and the FC are loosely synchronized to a time-slotted system with a maximum syntonization error of Δt. The FC shares a pairwise secret seed s_i with each participant u_i. The secret seed can be used to extract pairwise secret sequences between the FC and each u_i and authenticate the user (without providing non-repudiation).

Adversary model: We consider an adversary who launches active attacks on the voting scheme by injecting OFDM signals of his own choosing during the voting process. The adversary aims at modifying the voting outcome $\mathcal{T}$ at the FC. Two models are considered. In the first model, the adversary is unaware of the vote intent of each participant (value of each v_i), before votes are casted. This model applies to general voting procedures in which the vote intent cannot be inferred ahead of time by observing some physical phenomena. In the second model, the vote intent is known to the adversary. This model is relevant when the vote intent is correlated to some observable phenomenon. As an example, if participants vote on the state of a channel (idle or busy), the adversary can predict the votes of the participants by sensing the channel ahead of time. Finally, the adversary is not interested in launching denial-of-service attacks, in which a voting outcome is not reached.

III. PHYVOS: Physical Layer Voting

The key principle of PHYVOS is to simultaneously cast the votes by injecting energy on designated subcarriers. Energy detection is robust to active attacks and unintentional interference compared to vote decoding. An adversary attempting to modify a vote on subcarrier f_i, would have to "erase" the signal received by the FC 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. This information needs to be collected and synchronized for all voters. The PHYVOS scheme consists of three phases; *the vote request phase, the vote casting phase, and the tallying phase*.

A. Vote Request Phase

In the vote request phase, the FC signals to the participants for a vote. This phase is necessary to ensure that overhead gains are achieved by the simultaneous vote casting. Two mechanisms can be employed for requesting a vote; periodic voting and an on-demand voting. In periodic voting, participants exploit their synchronization to a common time-slotted system to cast their votes at fixed intervals without an explicit request from the FC. This operation mode is suitable for periodic network operations. In on-demand voting, the FC could broadcast a vote request message to all participants to initiate the voting process.

B. Vote Casting Phase

During the vote casting phase, participants simultaneously cast their votes to the FC. Each participant u_i is assigned two subcarriers f_1^i and f_2^i for casting his vote v_i. One subcarrier is used to cast a "yes" vote, while the other is used to cast a "no" 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 designated subcarrier. Therefore, we adopt a two-subcarrier solution.

Moreover, subcarriers f_1^i and f_2^i are not statically mapped to vote values. We use a secret cryptographically-secure pseudorandom binary sequence $R_i(s_i)$, shared between u_i and the FC, to randomize the mapping of vote values to subcarriers. This prevents the adversary from guessing the subcarrier where energy has to be injected to spoof a desired vote. Finally, we randomly select the transmitted symbol on the selected subcarrier to harden the nullification of the transmitted signal at the FC. Formally, vote casting involves the following steps.

1. Each u_i is assigned two subcarriers f_1^i and f_2^i.

2. Each u_i and the FC use a pseudorandom bit generator (PRBG) to individually generate a pairwise secret binary sequence, using s_i as a seed.

$$R_i(s_i) = \{r_i(n) = PRBG(n, s_i),\ n = 1, 2, \ldots\} \quad (3)$$

In (3), the time n is quantized to the OFDM symbol duration.

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

$$x_k(n) = \begin{cases} \alpha_y, & f_k = f^i_{1+v_i \oplus r_i(n)} \\ 0, & \text{otherwise,} \end{cases}, \quad n_0 \le n < n_0 + \ell. \quad (4)$$

where α_y is a randomly selected modulation symbol.

The vote casting phase for a set of M participants is depicted in Fig. 3. Participant u_1 is assigned subcarriers f_1 and f_2, participant u_2 is assigned subcarriers f_3 and f_4, etc. Participants transmit $\ell = 4$ symbol votes to cast a vote. The votes for the individual participants are $v_1 = 1$, $v_2 = 0, \ldots$, $v_M = 1$. Each participant u_i XORs his vote with the pseudorandom bit sequence $R_i(s_i)$ to determine the subcarrier index where a symbol vote will be transmitted at each time n. Participant u_1 transmits symbol votes in f_1, f_2, f_2, f_1, participant u_2 transmits symbols votes in f_4, f_4, f_3, f_4, etc. The symbol votes arrive at the FC such that OFDM symbols are formed. The vote casting phase is followed by the vote tallying phase.

C. Vote Tallying Phase

In the vote tallying phase, the FC computes the voting outcome $\mathcal{T}$ according to (2). To infer the votes of each participant, the FC computes the FFT of the digitized baseband OFDM signal to separate the spectral components to each of the subcarriers. The FC then uses an energy detector at each output of the FFT block to detect the transmitted symbol votes. Note

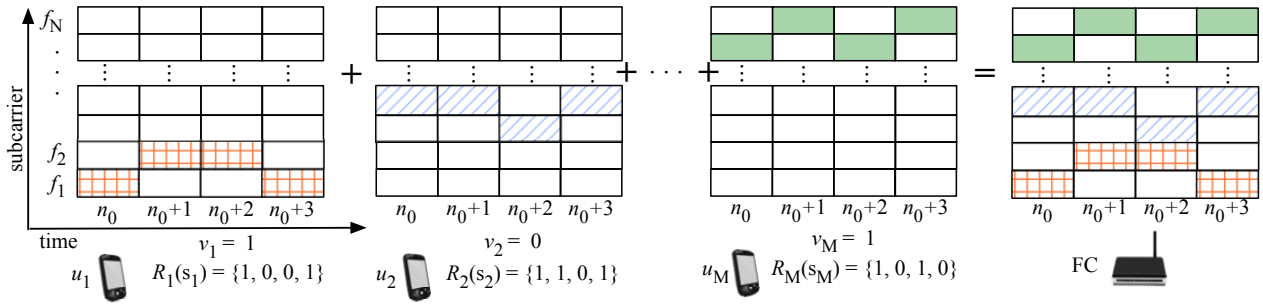


Fig. 3: The vote casting phase for M participants voting over N subcarriers (here $N = 2M$).

here that *no symbol demodulation is necessary to determine the presence of energy*. This process lasts for the duration of ℓ symbols that are used to submit the ℓ symbol votes. 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. If both subcarriers have an average power higher than γ_D (due to intentional or unintentional interference), or both have low power, the symbol vote is recorded in error. Formally, for a participant u_i, the recovery of v_i at the FC is performed as follows.

1. Sample the FFT output of subcarriers f_1^i and f_2^i assigned to u_i and compute the average received power over K samples:

$$p_j(n) = \frac{1}{K}\sum_{i=1}^{K}|y_j(i)|^2, \quad p_{j+1}(n) = \frac{1}{K}\sum_{i=1}^{K}|y_{j+1}(i)|^2, \quad (5)$$

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

2. The symbol votes $v_i(n)$ are computed as:

$$v_i(n) = \begin{cases} 0 \oplus r_i(n), & \text{if } p_j(n) > \gamma_D, \quad p_{j+1}(n) \leq \gamma_D \\ 1 \oplus r_i(n), & \text{if } p_j(n) \leq \gamma_D, \quad p_{j+1}(n) > \gamma_D \\ e, & \text{otherwise.} \end{cases} \quad (6)$$

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

3. The final vote v_i is computed as:

$$v_i = \begin{cases} 0, & \text{if } \sum_{n=n_0, v_i(n)\neq e}^{n_0+\ell-1}(-1)^{v_i(n)} > 0 \\ 1, & \text{if } \sum_{n=n_0, v_i(n)\neq e}^{n_0+\ell-1}(-1)^{v_i(n)} < 0 \\ e, & \text{otherwise.} \end{cases} \quad (7)$$

In (6), we XOR the output with the pseudorandom sequence $R_i(s_i)$ shared between u_i and the FC to correctly map the subcarrier index to the vote value. Moreover, in (7) we discard all inconclusive symbol votes with value $v_i(n) = e$. Such votes could be the result of unintentional interference from systems operating over the same spectrum, or an active attack.

The tallying operation at the FC is shown in the example of Fig. 3. For participant u_1, the FC detects an average power over γ_D on subcarriers f_1, f_2, f_2, and f_1. By XORing the output $\{0, 1, 1, 0\}$ with the random sequence $R_1(s_1) = \{1, 0, 0, 1\}$, it obtains the symbol votes $v_1(n_0) = 1$, $v_1(n_0 + 1) = 1$, $v_1(n_0 + 2) = 1$, and $v_1(n_0 + 3) = 1$, indicating a final vote $v_1 = 1$. Similarly, participant u_2 uses random sequence $R_2(u_2) = \{1, 1, 0, 1\}$ to compute $v_2 = 0$. The vote computation proceeds in parallel for all participants.

IV. Security Analysis

In this section, we evaluate the robustness of PHYVOS to an active adversary that attempts to flip the voting outcome $\mathcal{T}$. We first analyze the adversary's ability to modify a single vote v_i. We then extend our analysis to modifying $\mathcal{T}$.

A. Modifying a Single Vote

To modify a vote v_i casted by u_i, the adversary can attempt to modify the ℓ symbol votes used in the computation of v_i. Let u_i select subcarrier f_1^i for transmitting $v_i(n)$, based on v_i and $r_i(n)$. Without knowledge of $r_i(n)$, determining the subcarrier used by u_i to cast $v_i(n)$ before $v_i(n)$ is transmitted, is equivalent to a random guess. The probability of a successful guess is equal to 0.5. Even if the adversary correctly guesses f_1^i, he cannot "erase" energy from f_1^i, in order to flip the value of $v_i(n)$. 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 FC as well as the adversary and the FC, and precise synchronization between the voter and the adversary. We note that u_i randomly selects a_y for each symbol vote. Moreover, the channel between u_i and FC rapidly decorrelates with the distance from u_i. Unless the adversary is within a very short distance from u_i (within half a wavelength), the channel between u_i and FC becomes unpredictable [11].

The adversary can inject energy to f_2^i to nullify $v_i(n)$ (i.e., change the value of $v_i(n)$ from $v_i(n) = v_i$ to $v_i(n) = e$). According to (7), to nullify $v(i)$, all symbol votes $v_i(n_0), v_i(n_0 + 1), \ldots, v_i(n + 0 + \ell - 1)$ 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 randomly (based on $R_i(s_i)$) and independently, the probability of nullifying v_i becomes:

$$\begin{aligned} \Pr[v(i) = e] &= \Pr[v_i(n_0) = e, \ldots, v_i(n_0 + \ell - 1) = e] \\ &= 0.5^{\ell}. \end{aligned} \quad (8)$$

Note that (8) is true even if the value of v_i is known a priori, because v_i is XORed with $R_i(s_i)$ (see eq. (4)). From (8), we can select ℓ to drive $\Pr[v(i) = e]$ to any desired level.

B. Modifying the Voting Outcome

Let the vote tally used to compute the voting outcome $\mathcal{T}$ be equal to $\sum_i^M (-1)^{v_i} = \gamma + \mu$. Here, μ denotes the margin by which the tally exceeds the decision threshold γ (the case of $\sum_i^M = (-1)^{v_i} = \gamma - \mu$ is treated similarly). We analyze the

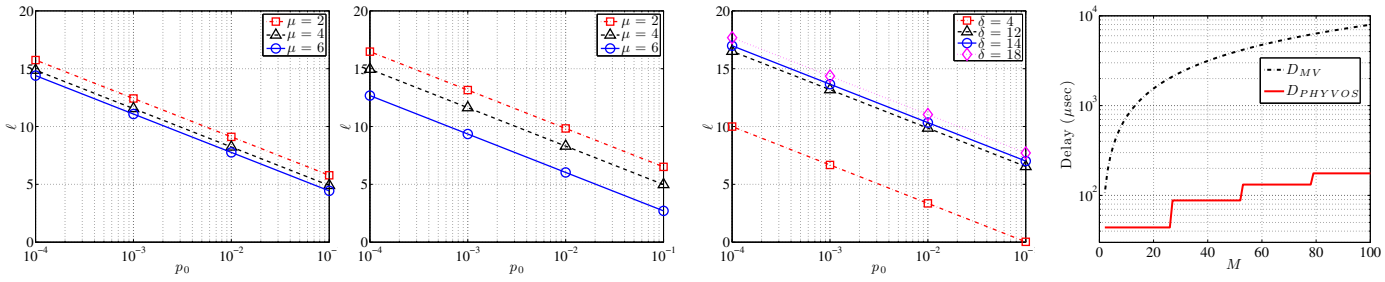


Fig. 4: (a) Minimum number of symbol votes ℓ to guarantee robustness p_0 when votes are unknown, (b) minimum number of symbol votes ℓ to guarantee robustness p_0 when votes are known, (c) minimum number of symbol votes ℓ to guarantee robustness p_0 for $\mu = 4$ and various δ, when votes are unknown, (d) voting overhead as a function of M for message-based voting (MV) and PHYVOS.

probability of successfully modifying the voting outcome for the two adversary models defined in Section II.

Proposition 1: An adversary unaware of the vote intent of each participant can modify the voting outcome $\mathcal{T}$ for a decision threshold γ and a margin μ with probability

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

where $n_1 = \frac{M+\gamma+\mu}{2}$ denotes the number of votes in favor of $\mathcal{T}_0$, HG denotes the pmf of the hypergeometric distribution

$$HG(K, N, k, n) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}, \tag{9}$$

B denotes the pmf of the binomial distribution

$$B(j, i, p) = \binom{i}{j} p^j (1-p)^{i-j}, \tag{10}$$

δ denotes the number of votes that the adversary attempts to nullify and $p = \Pr[v(i) = e]$ denotes the probability of nullifying a singe vote, which is given by (8).

Proof: The proof is provided in Appendix A. ∎

Proposition 2: An adversary with a priori knowledge of v_i, $\forall i$ can modify the voting outcome $\mathcal{T}$ for a decision threshold γ and a margin μ with probability

$$\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] = \sum_{i=\mu}^{n_1} B(i, n_1, p), \quad n_1 = \frac{M+\gamma+\mu}{2}. \tag{11}$$

Proof: The proof is provided in Appendix B. ∎

C. Selecting the Security Parameter ℓ

Propositions 1 and 2 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 according to the following corollaries.

Corollary 1: For an adversary unaware of the vote intent of each of participant, $\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] \leq p_0$ if

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

Proof: The proof is provided in Appendix C. ∎

Corollary 2: For an adversary with a priori knowledge of v_i $\forall i$, the probability $\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] \leq p_0$ if

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

Proof: The proof is provided in Appendix D. ∎

Form corollaries 1 and 2, 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 Figs. 4(a), 4(b), and 4(c), 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). Figure 4(a) considers an adversary with a priori knowledge of the intended votes. A total of 20 participants are considered and the voting threshold γ is set to zero (plurality rule). Finally, δ is set to the number of positive votes. Exact values of ℓ can be computed using numerical methods. We observe that the margin μ has a small impact on the required ℓ. This is because the adversary only attacks participants that intend to cast "yes" votes.

Figure 4, shows ℓ as a function of p_0 for an adversary without a priori knowledge of the intended votes. As μ increases, fewer symbol votes are necessary to provide the same robustness compared to the model in Fig. 4(a). This is because the adversary corrupts both "yes" and "no" votes, thus making it harder to close the margin. In Fig. 4(c), we plot ℓ as a function of p_0 for different δ and for $\mu = 4$. If few votes are attacked (small δ), the achieved robustness is high for relatively small ℓ. Moreover, the adversary gains diminish with the increase of δ beyond a certain threshold. This threshold is located at the number of "yes" votes (for $\mu = 4$ and $\gamma = 0$, $n_1 = 14$).

V. Voting Overhead

In this section, we compare the delay overhead of PHYVOS with the overhead 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 20μsec 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 [18]. The message digest size for SHA-256 is

32 Bytes. Assuming the highest possible transmission rate for 802.11g, each OFDM symbol 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

$$D_{MV} = 20M - 13 \quad \text{OFDM symbols.} \tag{12}$$

In PHYVOS, up to 26 participants can simultaneously cast their votes using ℓ OFDM symbols (2 subcarriers assigned per participant, no pilots necessary). 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}. The total delay required to cast M votes becomes,

$$D_{PHYVOS} = \left\lceil \frac{M}{26} \right\rceil \quad \text{OFDM symbols.} \tag{13}$$

Figure 4(d) 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 2sec (this delay increases if contention is taken into account), which makes it unsuitable for time-critical applications (e.g., spectrum sensing in CRNs [6]).

VI. Practical Considerations and Implementation

A. Frequency Synchronization

Radio oscillators do not operate at the same nominal frequency due to manufacturing imperfections. This frequency misalignment, known as carrier frequency offset (CFO), is amplified in mobile scenarios by the Doppler shift phenomenon. OFDM systems are particularly sensitive to CFO, due to the subcarrier orthogonality requirement [12]. The CFO has two critical effects on the demodulation process. First, subcarriers are no longer orthogonal causing inter-carrier interference (ICI) and reducing the SNR. Second, symbols at each subcarrier appear with arbitrary rotation in the constellation map. Finally, in the extreme case, a large CFO can cause a shift of the subcarrier bins at the receiver, whereby a symbol transmitted over subcarrier f_i is mapped to subcarrier f_j. This shift occurs if the CFO is larger than the subcarrier spacing [12].

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.

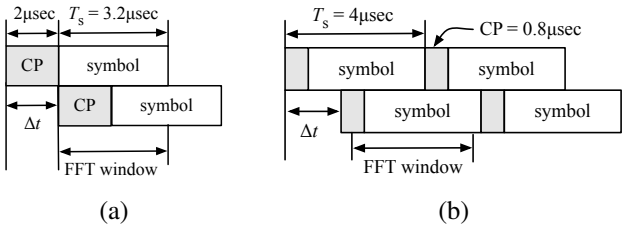


Fig. 5: (a) Increasing the CP, (b) casting a symbol vote in two symbol durations.

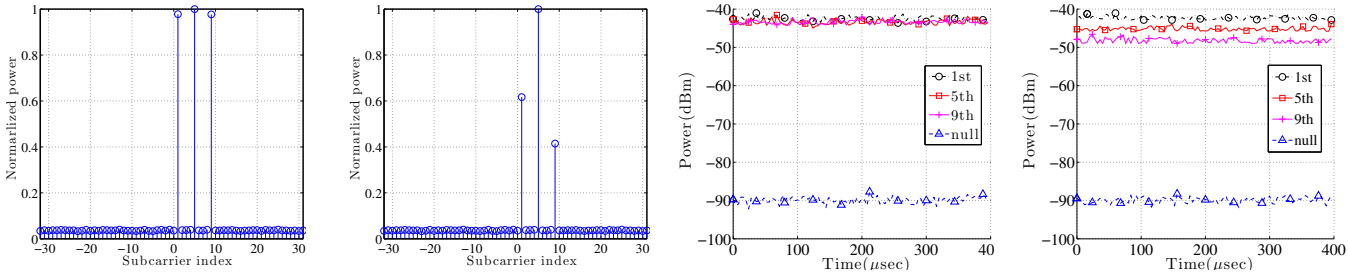
Finally, in typical OFDM systems, the subcarrier spacing is much larger than the expected CFO. As an example, the subcarrier spacing in 802.11g is 312.5KHz, while the expected device frequency misalignment is in the order of 10s of KHz [9]. In addition, the Doppler shift due to mobility is in the order of 10s of Hz (at 2.4GHz, 80Hz of Doppler shift equals a moving speed of 36Km/hr). Thus the compounded CFO due to oscillator imperfections and the Doppler effect are not sufficient to cause a shift of the subcarrier bins. This is also observed in the experimental implementation of PHYVOS, where energy was detected primarily in the designated subcarriers.

B. Time Synchronization

Another practical problem for PHYVOS is that symbol votes do not reach the FC 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 0.8μsec.

For PHYVOS, the time misalignment Δt between symbols at the receiver can be greater than 0.8μsec. 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 [9]. 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.

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 (5.2μsec vs. 4μsec). 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



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

Fig. 6: (a), (b) Normalized average received power per subcarrier, (c), (d) received power per subcarrier as a function of time.

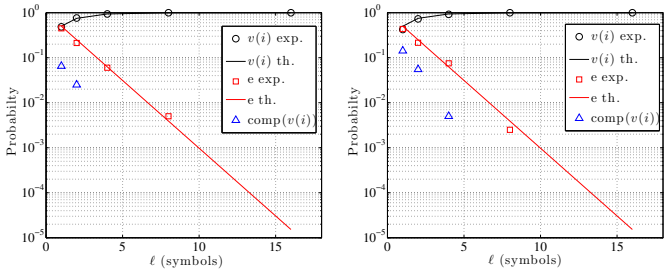
specifications, we can extend the symbol vote duration to two OFDM symbols, without increasing the CP duration. This solution comes at the expense of doubling the overhead for casting a symbol vote. A similar solution was adopted in [5]. The two solutions are shown in Fig. 5.

C. PHYVOS Implementation

Testbed setup: We implemented PHYVOS on NI USRPs 2921 devices, operating in the 2.4GHz band over a 39.6MHz spectrum. A total of four radios were at our disposal. Under normal operation, three radios operated as voters, while one radio operated as the FC. One radio was switched to an attacker role for adversarial scenarios. Voter radios were placed in a LoS configuration at varying distances from the FC within an office environment. We divided the 39.6MHz 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 0.8μsec, 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 20dBm (0.1W).

Selection of threshold γ_D: In the first experiment, we assigned the 1st, 5th, and 9th subcarrier to each of the three voter radios. Each voter radio 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 of 100msec was imposed between two consecutive votes. Figure 6(a) shows the normalized magnitude of the FFT output at the FC, averaged over the 1,000 transmitted symbols when the three voters are placed 5ft away from the FC (topology A). Figure 6(b) shows the same results when the three voters are at 5ft, 10ft, and 15ft away from the FC (topology B). We observe that in both cases the magnitude of the FFT output is significantly higher for the active subcarriers.

Figures 6(c) and 6(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, while 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.



(a) Topology A (b) Topology B

Fig. 7: Probability of tallying the correct vote $v(i)$, having an inconclusive vote e, or flipping the vote to comp($v(i)$).

Voting in the presence of an adversary: In the second experiment, we implemented an adversarial scenario. One of the three voter USRPs was assigned the role of an attacker. 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. The experiment lasted for 10^6 symbol votes (no time gap). Figure 7 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 (8).

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 1 with the increase of ℓ, while 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, with a slight increase in the probability of comp($v(i)$). This was primarily observed due to the near-far effect for the most distant voter (placed at 15ft from the FC).

VII. Related Work

The use of voting for improving reliability has been studied since the 1950s [19], with a long literature on various reliability

and efficiency aspects [3]. In the context of wireless networks, voting finds wide application to data fusion, intrusion detection and secure localization in WSNs [2], [7], [8], [20], real-time coordination in multi-agent systems [4], fault-tolerant protocols [10], [13], and distributed spectrum sensing in CRNs [1]. 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 [18]. Compared to message-based voting, PHYVOS requires significantly less communication overhead, without sacrificing robustness to vote manipulation.

The messaging overhead for implementing majority voting has primarily been a concern in distributed consensus protocols. A wealth of prior works on *gossip algorithms* have been devoted to fine-tuning tradeoffs between the voting outcome accuracy, the messaging overhead, and the time until a consensus is reached [17]. The majority of gossip algorithms were not designed with security in mind and are susceptible to manipulation by internal and external entities. Our setup differs from that of gossip algorithms in that we consider a centralized, one-hop topology. This topology can arise in infrastructure-based networks following a client-base station model, and in distributed networks where majority voting is used for taking local decisions or performing aggregation [14].

Form an implementation standpoint, the most relevant works to ours are presented in [5], [15]. In [5], 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. The authors outline system implementation details related to the concurrent symbol transmission over different subcarriers and the reception of a combined OFDM symbol. To combat the problems of frequency and time synchronization, the detection of individual ACKs is based on energy and not demodulation, similar to our scheme. However, verification of the ACK integrity is beyond the scope of the SMACK design. A single attacker could emulate ACK responses for all broadcast receivers, by transmitting an OFDM symbol with energy on all subcarriers.

In [15], Rahul et al. propose 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. This additional communication overhead for maintaining tight synchronization and continuously estimating the channel makes the SourceSync solution inadequate for our purposes.

VIII. Conclusion and Future Work

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 enabling fast and communication-efficient voting for wireless applications where secure voting is time-critical. We showed that PHYVOS maintains the integrity of the voting outcome with high probability, without using cryptographic primitives. The proposed scheme was demonstrated for binary voting. As future work, we intend to extend PHYVOS to accommodate x-ary voting. We will further explore methods for accommodating a larger number of votes over the same spectrum by allowing two or more participants to simultaneously cast votes over the same subcarriers. Finally, we will investigate the application of PHY-layer voting to fully distributed consensus algorithms.

Acknowledgments

This research was supported in part by the NSF under grant CNS-1409172 and ARO grant W911NF-13-1-0302. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the NSF.

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

[4] 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.

[5] 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.

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

[7] 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.

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

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

[10] 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.

[11] 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.

[12] R. v. Nee and R. Prasad. *OFDM for wireless multimedia communications*. Artech House, Inc., 2000.

[13] 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.

[14] S. Ozdemir and Y. Xiao. Secure data aggregation in wsns: A comprehensive overview. *Comp. Nets*, 53(12):2022–2037, 2009.

[15] 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.
[16] K. Sampigethaya and R. Poovendran. A framework and taxonomy for comparison of electronic voting schemes. *Computers & Security*, 25(2):137–153, 2006.
[17] D. Shah. *Gossip algorithms*. Now Publishers Inc, 2009.
[18] D. R. Stinson. *Cryptography: theory and practice*. CRC press, 2005.
[19] J. Von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Aut. studies*, 34:43–98, 1956.
[20] 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.

APPENDIX A - PROOF OF PROPOSITION 1

For a voting outcome $\mathcal{T}$ selected with a margin μ, there are $n_1 = \frac{M+\gamma+\mu}{2}$ votes in favor of $\mathcal{T}$ and $n_2 = \frac{M-\gamma-\mu}{2}$ votes against $\mathcal{T}$. Let the in favor votes be "yes" votes. To flip $\mathcal{T}$, the adversary must nullify at least μ more "yes" votes than "no" votes and drop the vote tally at or below γ. For an adversary that attempts to nullify a total of δ votes, the probability that i of them are "yes" votes is given by a hypergeometric distribution.

$$\Pr[I = i] = HG(n_1, M, i, \delta), \tag{14}$$

where I is an RV denoting the number of attacked "yes" votes when a total of δ votes are attacked. Each vote is successfully nullified with probability $p = \Pr[v_i = e] = 0.5^\ell$. Let X be an RV denoting the number of votes successfully nullified, when i are attacked. Because the nullification of each vote is an independent Bernoulli trial, X follows the binomial distribution

$$\Pr[X = x] = B(x, i, p), \quad p = 0.5^\ell. \tag{15}$$

Similarly let Y be an RV denoting the number of "no" votes that are successfully nullified. For Y,

$$\Pr[Y = y] = B(y, \delta - i, p), \quad p = 0.5^\ell. \tag{16}$$

The probability that the number of successfully nullified "yes" votes exceeds the number of nullified "no" votes by exactly z votes is given by RV $Z = X - Y$. The pmf of Z can be computed using the convolution formula.

$$\begin{aligned}
\Pr[Z = z] &= \sum_x \Pr[X = x, Y = x - z] \\
&= \sum_x \Pr[X = x] \Pr[Y = x - z] \\
&= \sum_x B(x, i, p) B(x - z, \delta - i, p) \\
&= \sum_{x=z}^{\min\{i, \frac{\delta+z}{2}\}} \frac{\binom{i}{x}\binom{\delta-i}{x-z}}{\binom{\delta}{2x-z}} B(2x - z, \delta, p). \qquad (17)
\end{aligned}$$

Summing over all $z \geq \mu$ yields,

$$Pr[Z \geq \mu] = \sum_{z=\mu}^{i} \sum_{x=z}^{\min\{i, \frac{\delta+z}{2}\}} \frac{\binom{i}{x}\binom{\delta-i}{x-z}}{\binom{\delta}{2x-z}} B(2x - z, \delta, p). \tag{18}$$

Using (14) and (18), we compute

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

APPENDIX B - PROOF OF PROPOSITION 2

For a voting outcome $\mathcal{T}$ selected with a margin μ, there are $n_1 = \frac{M+\gamma+\mu}{2}$ votes in favor of $\mathcal{T}$ and $n_2 = \frac{M-\gamma-\mu}{2}$ votes against $\mathcal{T}$. Let the in favor votes be "yes" votes. When the adversary is aware of the vote intent of each participant, he can target only "yes" votes. The voting outcome $\mathcal{T}$ is flipped if at least μ "yes" votes are nullified. For an adversary that attempts to nullify a total of δ "yes" votes, the number of successfully nullified votes follows the binomial distribution.

$$\Pr[X = x] = B(x, \delta, p), \quad p = 0.5^\ell. \tag{19}$$

Summing over all values of $x \geq \mu$ yields,

$$\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] = \sum_{i=\mu}^{\delta} B(i, \delta, p), \quad p = 0.5^\ell. \tag{20}$$

The δ is equal or smaller to the number of "no" votes n_1.

APPENDIX C - PROOF OF COROLLARY 1

We wish to determine the value of ℓ for which $\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] \leq p_0$. From (17), it follows that

$$\begin{aligned}
\Pr[Z = z] &= \sum_x \frac{\binom{i}{x}\binom{\delta-i}{x-z}}{\binom{\delta}{2x-z}} B(2x - z, \delta, p) \qquad (21a) \\
&< \sum_x B(2x - z, \delta, p) \qquad (21b) \\
&< \frac{\delta p}{z} \qquad (21c)
\end{aligned}$$

In (21b), we used the fact that $\binom{N}{n}\binom{M}{m} < \binom{N+M}{n+m}$. In (21c), we used the Chernoff bound to limit the tail sum of the Binomial distribution. Substituting to (9) yields,

$$\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] < \sum_{i=\mu}^{n_1} HG(n_1, M, i, \delta) \sum_{z=\mu}^{i} \frac{\delta p}{z}. \tag{22}$$

Limiting the RHS of (22) by p_0 and solving for p results in

$$p < \frac{p_0}{\delta \sum_{i=\mu}^{n_1} HG(n_1, M, i, \delta) \sum_{z=\mu}^{i} \frac{1}{z}}. \tag{23}$$

Substituting $p = 0.5^\ell$ and solving for ℓ completes the proof.

APPENDIX D - PROOF OF COROLLARY 2

The proof follows by using the Chernoff bound to limit the tail probability of the binomial distribution in (20).