Trusted Computing for Fault-Prone Wireless Networks

Seth Gilbert[1] and Dariusz R. Kowalski[2]

[1] National University of Singapore, Singapore
gilbert@comp.nus.edu.sg

[2] University of Liverpool, United Kingdom
D.Kowalski@liverpool.ac.uk

Abstract. We consider a fault-prone wireless network in which communication may be subject to wireless interference. There are many possible causes for such interference: other applications may be sharing the same bandwidth; malfunctioning devices may be creating spurious noise; or malicious devices may be actively jamming communication. In all such cases, communication may be rendered impossible.

In other areas of networking, the paradigm of "trusted computing" has proved an effective tool for reducing the power of unexpected attacks. In this paper, we ask the question: can some form of trusted computing enable devices to communicate reliably? In answering this question, we propose a simple "wireless trusted platform module" that limits the manner in which a process can access the airwaves by enabling and disabling the radio according to a pre-determined schedule. Unlike prior attempts to limit disruption via scheduling, the proposed "wireless trusted platform module" is general-purpose: it is independent of the application being executed and the topology of the network.

In the context of such a "wireless trusted platform module," we develop a communication protocol that will allow any subset of devices in a region to communicate, despite the presence of other disruptive (possibly malicious) devices: up to k processes can exchange information in the presence of t malicious attackers in $O(\max(t^3, k^2)\log^2 n)$ time. We also show a lower bound: when $t < k$, any such protocol requires $\Omega(\min(k^2, n)\log_k n)$ rounds; in general, at least $\Omega(\min(t^3, n^2))$ rounds are needed, when $k \geq 2$.

1 Introduction

Wireless networks are everywhere, enabling devices to communicate and exchange information without the need for physical infrastructure. Wireless networks rely on the open airwaves for communication, and the open airwaves are

[⋆] This publication was prepared while the first author was at EPFL, Switzerland. The work of the second author was supported by the Engineering and Physical Sciences Research Council [grant numbers EP/G023018/1, EP/H018816/1].

publicly accessible by anyone and everyone. This openness has advantages, allowing universal participation and creating a lower barrier to entry; it also has disadvantages: any user can join the network and cause disruption. Disruption may be caused intentionally, by malicious parties, or unintentionally, by other applications sharing the same bandwidth.

Trusted Computing. Recently, the paradigm of *Trusted Computing* has come to be seen as a powerful technique for reducing vulnerability to attack. (See, e.g., [29, 33].) The basic idea underlying *trusted computing* is that each computer (or networked device) will contain a tamper-proof component (often a special-purpose chip) known as a *trusted platform module* (TPM) that provides certain reliable guarantees. For example, the TPM may contain cryptographic authentication keys that securely identify the computer. The TPM may also contain a mechanism that protects data stored on a computer, or that may provide certain guarantees as to the software running on that computer. Elements of the trusted computing architecture are implemented today in Windows Vista, for example, in BitLocker Drive Encryption.

In this paper, we examine the application of trusted computing techniques to wireless networking, in particular to the problem of interference and disruption (either benign or malicious) on the wireless airwaves. Imagine that every wireless device has a “wireless” TPM (wTPM) that controls access to the radio[3]. (See Figure 1 for a simplified schematic representation.) When the radio is enabled by the wTPM, the software running on the wireless device can send and receive messages; conversely, when the radio is disabled by the wTPM, the software running on the wireless device cannot access the radio. Thus, even when a malicious attacker hacks or takes control of a wireless devices, (s)he can only create interference when the radio is enabled[4].

An important design criterion for a wTPM is that it be simple, and that it be as computation-agnostic as possible. The wTPM should not be aware of the computation running on the wireless device, nor should it monitor the communication sent and received over the radio. Ideally, it should simply connect and disconnect the radio, irregardless of what the device is doing or whether the device is sending or receiving a message. The fundamental open question is whether it is possible to design such a simple, computation-agnostic wTPM that will still allow wireless devices to perform the communication and computation that they desire, without sacrificing efficiency.

Overview of Results. In this paper, we make some progress toward answering these questions. We focus on the basic problem of reliably exchanging information: there are at most k wireless devices—from some larger universe of n

[3] Note that use of the radio frequencies is already heavily regulated in most countries, and hence it might be feasible to require all legal devices to be equipped with such a wTPM; of course the “trusted computing” approach will be ineffective for a concerted attacker with access to illegal hardware.

[4] While “trusted computing” is sometimes criticized for its privacy implications, these problems are less severe in the wireless case where the wTPM only affects when the radio is enabled, while revealing no personal information.

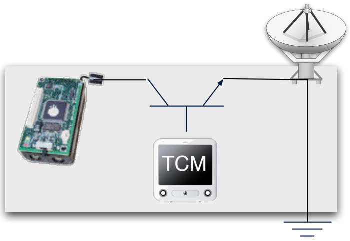

Fig. 1. Simple schematic of sensor device with a wireless "trusted computing module"(e.g., a wTPM) controlling power to the antenna.

devices—that want to exchange information with each other. (We assume that k is fixed; however, we discuss in Sections 5.3 how to adapt to varying numbers of participants.) At the same time, there are at most t malicious devices that disrupt communication. These devices may broadcast corrupt messages, or they may "jam" the airwaves—when their radios are enabled by the wTPM—preventing any information form being exchanged.

Each device contains two pieces of software: (1) the wTPM, which is tamper-proof, and (2) the communication protocol, which may be corrupt on malicious devices. *(1) wTPM*: The wTPM consists of a fixed binary sequence indicating whether the radio is enabled or disabled at any given time. When the radio is enabled, the device can communicate; when the radio is disabled, it can neither send nor receive. The behavior or the wTPM is fixed in advance, and is not affected by anything that occurs during an execution. *(2) Communication protocol*: The communication protocol determines whether the device broadcasts or receives in any given round, if the radio is enabled. We focus on *oblivious* protocols where the broadcast/receive schedule is also fixed in advance.

Our main result consists of the wTPM design, along with an efficient communication protocol that is compatible with the wTPM. Our protocol runs in $\Theta(\max(t^3, k^2)\log^2 n)$ rounds; surprisingly, this is almost as efficient as the best (oblivious) protocols for exchanging information, even when all the devices are honest: every such protocol requires $\Omega(\min(k^2, n)\log_k n)$ rounds [6,11]. We provide a complementary lower bound, showing that exchanging information requires at least $\Omega(\min(t^3, n^2))$ rounds, for $k \geq 2$. (Note: there is a trivial $O(n^2)$ solution that selectively enables each pair of processes.) Together, these lower bounds indicate that our proposed protocol is near optimal in most cases.

Both the wTPM and the protocol are generated by a random process which is designed to activate only (approximately) k/t radios in each round. Even though communication among the honest devices is (effectively) limited to one-to-one communication, we still exchange information nearly as efficiently as protocols that rely on one-to-many communication, e.g., have each sender transmit his information while all the other processes listen. Of note, the resulting protocols are relatively simple to implement when provided with a good source of (pseudo)-random bits (or data structures for generating such bits, e.g., extractors).

Our approach to security also has a secondary benefit: it can significantly reduce the power usage of wireless protocols. Powering the radio is one of the most energy-consuming operations for a small wireless device. If the wTPM disables the radio sufficiently often, then it forces every protocol running on the device to be much more energy efficient than might otherwise be the case. When $t = \Theta(k)$, the wTPM enforces a high-level of energy efficiency, enabling only $O(1)$ devices in each round. By contrast, prior protocols for information exchange activate $\Theta(k)$ processes per round.

Finally, in order to enable more general applications, we also briefly consider the *continuous* version of information exchange, where there are an unknown number of processes that occasionally have information to distribute. A natural generalization of our protocol ensures that every message injected in some round r will be delivered by round $r + O(\max(t^3, \ell^2) \log^3 n)$, as long as there are at most $\ell \leq n$ active messages during that interval.

Other Approaches to Tolerating Malicious Devices in a Radio Network. The idea of enforcing a fixed broadcast schedule for a wireless radio, in order to avoid malicious interference, has been previously proposed on several occasions. Koo [20], in one of the first papers studying the problem of reliable broadcast in a wireless network subject to Byzantine failures, suggested that devices be forced to follow a "round-robin" schedule, preventing malicious devices from broadcasting out of turn. Later papers (e.g., [4,5]) followed this approach as well. In general, such an approach either assumes that only one device is enabled at a time—leading to $\Omega(n)$ or $\Omega(n^2)$ running times—or it relies on some geographic property to determine whether a device can broadcast, for example, enabling devices in specific regions to broadcast in a given round. By contrast, in this paper, we attempt to develop a *generic* wTPM that is computation-agnostic, geographically ignorant, and yet still achieves efficient performance.

An alternate approach for dealing with malicious disruption is to posit some limit on the *amount* of disruption (see, e.g., [1, 16, 21]), or on the *rate* at which the devices can cause disruption [2]. Such limits might arise from practical considerations, e.g., the size of the battery on a malicious device, or from hardware constraints, e.g., a device might not be allowed to broadcast at above some specified rate. The latter approach, in particular, has significant promise for a wTPM solution, as a wTPM might enforce a bounded rate of broadcast.

A third approach for coping with malicious disruption is to leverage the availability of more than one communication channel: while malicious devices may disrupt some subset of the available channels, reliable communication can proceed on the other channels. This has proved a popular approach, as it requires minimal limitations on the power or scope of the malicious devices. (See, for example, [12–15,17,26,28,31,32].) One open question is whether such multichannel solutions could be even more efficient if a wTPM were available.

Other Related Work. The problem of resolving contention among a set of (honest, fault-free) devices on a multiple-access channel has been extensively studied (see, e.g., [3, 19, 22, 34], among many others). Wireless networks with crash failures (but not Byzantine failures) have also been studied extensively (e.g., [8–10, 24]). In essence, the challenge in this paper is to solve the problem of contention among honest processes, while simultaneously preventing the malicious processes from jamming.

Recently, there has been much interest in other models of interference, such as the SINR model [18, 27], and the *dual-graph model* [25]. These models capture interference in a somewhat more sophisticated manner, and hence it is an interesting open question how to cope with malicious interference in such models.

2 A Model for Wireless Trusted Computing

Model. Let Π be a set of n processes. Each process knows n and the set Π. Each process is either *active* or *passive*. An active process can send/receive messages and perform computations; a passive process cannot act in any way. At most k honest processes are activated. At most t dishonest (or *Byzantine*) processes may also be activated; such processes may act in an arbitrarily malicious fashion.

Processes communicate with a radio over a collision-prone wireless channel. In each round, each process (whether honest or dishonest) can either broadcast or listen. When exactly one process broadcasts, every other process receives the message; when more than one process broadcasts, no process receives anything.

Trusted Computing. Each device is equipped with a tamper-proof *wireless trusted-platform module (wTPM)*. The wTPM at each process is initialized with a binary string that indicates, for each round, whether the radio is enabled or disabled. When the radio is disabled, the process can neither send nor receive. The Byzantine devices cannot corrupt the wTPM, meaning that they cannot broadcast when the radio is disabled by the wTPM.

We define an *algorithm* $\langle T, B \rangle$ to be two binary (n, m)-matrices. We refer to matrix T as the *radio-enable* matrix and matrix B as the *broadcast-listen* matrix. Rows of each matrix correspond to processes, and columns corresponds to rounds. That is, row p of matrix T is the initialization string for the wTPM at process p: the radio at process p is enabled in round r if and only if $T[p, r] = 1$. Similarly, row p of matrix B indicates whether process p broadcasts or listens in each round: process p broadcasts in round r if $B[p, r] = 1$; otherwise it listens. We assume, for simplicity, that whenever a process is enabled and scheduled to broadcast, it transmits all available information. (There is no required relation between T and B.)

By definition, algorithms are *oblivious*: the behavior of each process is fixed; they do not adapt to adversarial behavior. Oblivious protocols have several advantages: they are often more robust, as they do not depend on accurately observing ongoing events. In the case of a wTPM, an oblivious wTPM would appear more plausible, as it can be constructed in a generic protocol-independent manner (as compared to attempting to adapt to circumstances).

Exchanging Information. We consider the problem of (k,t)-*information exchange.* Define $P \subseteq \Pi$ to be the set of at most k active honest processes. (Note that activations are local; a process knows only whether it is in set P or not.) Each process in P is initialized with a rumor. At the end of the execution, every active, honest process should transmit its rumor to every other active, honest process, as long as there are at most t active dishonest processes.[5]

The primary metric is time complexity, i.e., the number of rounds that the protocol executes. In Section 5.3 we consider a continuous variant where we count from a rumor's injection until the rumor is received by all other honest processes. Another complexity measure of interest is energy consumption, defined as the sum, over all rounds, of the number of radio-enabled processes.

3 Lower bound

Theorem 1. *For the problem of (k,t)-information exchange:(i) if $k \geq t$, then $\Omega(\min(k^2,n)\log_k n)$ rounds are required; (ii) if $k \geq 2$, then $\Omega(\min(t^3,n^2))$ rounds are required.*

Proof. When $k \geq t$, the lower bound of $\Omega(\min(k^2,n)\log_k n)$ follows from bounds on superimposed codes [6,11], which holds even when all processes are honest.

Now assume $k \geq 2$. We show that there are two honest processes that fail to exchange rumors in the first $t^2(t-2)/32$ rounds. It is sufficient to consider the case when $t^2(t-2)/32 < n(n-1)/4$.

For $0 \leq i \leq n$, define R_i to be the set of rounds such that $A_r = \{p : T[p,r] = 1\}$ is of size i. We omit rounds in set $R_0 \cup R_1$ from the analysis, as at most one radio-enabled process cannot send a message to any other process. We focus on sets R_2 and $R_{\geq 3} = \bigcup_{i \geq 3} R_i$. Let S_2 be the set of all pairs that are radio-enabled in rounds in R_2, i.e., $S_2 = \{\{p,q\} : \exists_{r \in R_2}\ T[p,r] = T[q,r] = 1\}$. We have $|S_2| \leq |R_2| \leq t^2(t-2)/32 < n(n-1)/4$.

Let F_2 be a set of $t/2$ processes such that the number of pairs of elements from F_2 that are in S_2 is smaller than $\binom{t/2}{2}/2$. Such a set exists by a probabilistic argument: the expected number of pairs from S_2 included in the random set of $t/2$ processes is smaller than:

$$|S_2| \cdot \frac{\binom{n-2}{t/2-2}}{\binom{n}{t/2}} = |S_2| \cdot \frac{(t/2-1)(t/2)}{(n-1)n} < n(n-1)/4 \cdot \frac{(t/2-1)(t/2)}{(n-1)n} = \frac{1}{2} \cdot \binom{t/2}{2} ,$$

and therefore a set with this property exists.

Consider processes in F_2 and rounds in $R_{\geq 3}$. Let R be the subset of $R_{\geq 3}$ containing rounds $r \in R_{\geq 3}$ such that $|\{p \in F_2 : T[p,r] = 1\}| = 2$; let S be the set $\{\{p,q\} : p,q \in F_2\} \setminus S_2$. Since each round in R is associated with at most one pair in S, and there are at least $\binom{t/2}{2} - \binom{t/2}{2}/2 = \binom{t/2}{2}/2$ pairs in S, there is a pair $\{p^*,q^*\}$ in S associated with at most $\frac{|R|}{|S|} \leq \frac{t^2(t-2)/32}{\frac{1}{2}\cdot\binom{t/2}{2}} \leq t/2$ rounds in $R_{\geq 3}$.

[5] Malicious processes may create their own rumors, which cannot be distinguished from honest rumors; it is unavoidable that processes may deliver such rumors.

Let $R^*_{\geq 3} = \{r \in R_{\geq 3} : T[p^*, r] = T[q^*, r] = 1\}$. By the choice of p^*, q^*, we have $|R^*_{\geq 3}| \leq t/2$. Since $R^*_{\geq 3} \subseteq R_{\geq 3}$, for every $r \in R^*_{\geq 3}$ there is a process $p(r)$ different from p^*, q^* such that $T[p(r), r] = 1$. Let F_3 be the set of processes $\{p(r) : r \in R^*_{\geq 3}\}$. An estimate $|F_3| \leq |R^*_{\geq 3}| \leq t/2$ holds. We define F as $(F_2 \setminus \{p^*, q^*\}) \cup F_3$. It follows that $|F| \leq |F_2| - 2 + |F_3| < t$.

Let $\{p^*, q^*\}$ be the set of honest processes, and F be the set of Byzantine processes. The adversary's strategy is as follows: whenever a Byzantine process is radio-enabled according to T, it transmits. It is easy to check that in each round where processes p^*, q^* are radio-enabled by T, there is also another process in F which is radio-enabled by T, and thus it interrupts any attempted transmission between p^* and q^*. Indeed, they cannot both be active in round $r \in R_2$, since $\{p^*, q^*\}$ in set S, and set S does not contain—by definition—any pair in set S_2, i.e., any pair that is active alone in some round in R_2. Therefore no communication can occur between p^* and q^* during rounds in R_2. (Recall that this is also impossible in rounds in R_0 and R_1, by definition.) Consider a round $r \in R_{\geq 3}$. If p, q are both active in round r, then, by definition of $R_{\geq 3}$, there must be at least one more process active in this round. Hence, round r satisfies the condition in the definition of set F_3, which means that at least one process $p \in F_3$ is such that $T[p, r] = T[p^*, r] = T[q^*, r] = 1$, and thus p jams the communication between p^*, q^* in round r. Therefore, rumors between p^*, q^* are not exchanged. Finally, note that set F of Byzantine processes is of size $|F_2| - 2 + |F_3| < t$ and there are only two honest processes p^*, q^*.

4 Implementing Information Exchange

We now present an algorithm that performs (k, t)-*information exchange* in time $\Theta(\max(t^3, k^2) \log^2 n)$. Each process knows that there are at most k honest processes active, and at most t Byzantine processes active. We define $\langle T, B \rangle$ (i.e., the algorithm) using a random process, and argue that the resulting algorithm achieves the desired results with high probability. This both shows that there exists an efficient deterministic solution to the problem of (k, t)-*information-exchange*, via the probabilistic method, and shows how to find it efficiently.

We construct the algorithm out of sub-pieces. As we do not know how many honest processes are active, we define algorithm $A(\ell, \ell/2)$ which assumes that there are more than $\ell/2$ but at most ℓ honest processes active, and which runs in time $\Theta(\max(t^3/\ell, \ell^2) \log^2 n)$ rounds. The final protocol consists of concatenating the algorithms $A(\cdot, \cdot)$ for exponentially decreasing ranges, i.e., $A_k = A(k, k/2)$ & $A(k/2, k/4)$ & $\ldots$ & $A(2, 1)$, where & represents concatenation. Summing the costs as ℓ decreases, the final running time is $\Theta(\max(t^3, k^2) \log^2 n)$.

4.1 Defining Algorithm $A(\ell, \ell/2)$

We now define $A(\ell, \ell/2) = \langle T, B \rangle$, for any $2 \leq \ell \leq k$, where ℓ is a power of 2. We divide $A(\ell, \ell/2)$ into three "sub-algorithms", $\langle T_1, B_1 \rangle$, $\langle T_2, B_2 \rangle$, $\langle T_3, B_3 \rangle$, which when concatenated, form $A(\ell, \ell/2)$. Let $m = c \cdot \max(t^3/\ell, \ell^2) \log n$, for a sufficiently large constant c, to be derived in the analysis. (The function of each of these stages is described in more detail in Section 4.2.)

- *Stage 1:* $\langle T_1, B_1\rangle$ We define T_1 to be a binary $(n \times m)$-matrix where, for every p, r, each bit $T_1[p, r] = 1$ with probability $\min(1/t, 1/\ell)$; otherwise $T_1[p, r] = 0$. We define B_1 to be a binary $(n \times m)$-matrix where, for every p, r, each bit $B_1[p, r] = 1$ with probability $1/2$; otherwise $B_1[p, r] = 0$.
- *Stage 2:* $\langle T_2, B_2\rangle$ Define the $(n \times 2m)$-matrix T' as follows, for all p : For each odd column $r = 1, 3, 5, \ldots$, define $T'[p, r] = 1$ with probability $\min(1/t, 1/\ell)$; otherwise $T'[p, r] = 0$. For each even column $r = 2, 4, 6, \ldots$, define $T'[p, r]$ to be identical to the preceding column, i.e., $T'[p, r] = T'[p, r - 1]$. Define T_2 as $2 \log n$ repetitions of T'; T_2 is a $(n \times (4m \log n))$-matrix.
  Define B' as follows, for all p: For each odd column $r = 1, 3, 5, \ldots$, we define $B'[p, r] = 1$ with probability $1/2$; otherwise $B'[p, r] = 0$. For each even column $r = 2, 4, 6, \ldots$, we define $B'[p, r]$ to be the inverse of the preceding column, i.e., we define $B'[p, r] = (1 - B'[p, r - 1])$. Define B_2 as $2 \log n$ repetitions of B'. Note that B_2 is a $(n \times (4m \log n))$-matrix.
- *Stage 3:* $\langle T_3, B_3\rangle$ We define T_3 to be identical to T_1, i.e., $T_3 = T_1$. We define B_3 to be the inverse of B_1, i.e., for all p, r: $B_3[p, r] = (1 - B_1[p, r])$.

Thus, $A(\ell, \ell/2)$ is defined by the matrices T_1 & T_2 & T_3 and B_1 & B_2 & B_3. It follows that the length of algorithm $A(\ell, \ell/2)$ is $O(\max(t^3/\ell, \ell^2) \log^2 n)$.

4.2 Overview of the Analysis

We now analyze the protocol and show that it is correct and efficient. We consider $A(\ell, \ell/2)$, where $2 \leq \ell \leq k$. As we have already bounded the running time, we focus on showing that every honest process succeeds in transmitting its rumor to every other honest process. Fix ℓ such that there are more than $\ell/2$ and at most ℓ honest, active processes. Recall that P is the set of honest, active processes. We examine each of the three "sub-algorithms" separately.

- *Stage 1:* guarantees each rumor is delivered to $> (|P| - \ell/8)$ honest processes.

When this stage completes, each rumor is known to a large number of honest processes. However, there may be no one honest process that knows all the rumors. While it is relatively cheap to distribute each rumor to a large fraction of the participants, it is more expensive to deliver each rumor to *every* other active participant. In the first stage, rumors are delivered directly, in a pairwise fashion: each participant directly sends its rumor to a large fraction of the other participants. In order to deliver every rumor directly in a pairwise fashion to *every* process would require approximately $\Theta(k^2 t)$ rounds. The second stage avoids this by exchanging rumors indirectly.

- *Stage 2:* guarantees that there is a subset $P^* \subseteq P$ of size $\ell/8$ where every process in P^* has received all the rumors.

The second stage relies on a more careful examination of the communication graph defined by the protocol. Unlike in Stage 1, we do not rely on direct edges between pairs of processes, but instead expect rumors to be passed indirectly over multiple "hops" in the induced communication graph. We show that the communication graph, when appropriately defined, has good *expansion* (see Definition 1), which immediately implies that the communication graph has a large component with small diameter (see Corollary 1). We can then conclude that every process in the large component learns every rumor.

– *Stage 3:* guarantees that each honest process receives at least one message from a process in P^*.

Processes in P^* cooperate to ensure that every process in P is notified of all the rumors. Notably, it turns out that Stage 3 is the symmetric opposite of Stage 1: whereas Stage 1 involved disseminating rumors, Stage 3 involves collecting them. We now proceed to analyze the three parts in more detail.

4.3 Stage 1: Spreading

The goal of the first stage, intuitively, is to distribute each rumor to more than $|P| - \ell/8$ honest participants. We show that the sub-protocol $\langle T_1, B_1 \rangle$ guarantees the following property: For every $P^* \subseteq P$, where $|P^*| = \ell/8$, and for every rumor ρ, there exists some process $q \in P^*$ such that q receives ρ during Part 1 of the protocol. This implies that we can choose any subset of P of size $\ell/8$ and be sure that every rumor is known by at least one member of that subset.

For the purpose of the next lemma, fix some set P of size bigger than $\ell/2$ and at most ℓ, some subset $P^* \subseteq P$ of size $\ell/8$, and some process $p \in P \setminus P^*$. (When $p \in P^*$, the property follows trivially.) Also, fix some set F of at most t Byzantine processes. We calculate the probability that the rumor from process p reaches some process in P^* without being disrupted by a process in F:

Lemma 1. *For given sets P, P^*, F and process $p \in P \setminus P^*$: there is some round r and some process $q \in P^*$ such that p successfully transmits its rumor to q in round r (i.e., p is the only process radio-enabled that transmits and q is the only process radio-enabled that listens in round r) with probability at least $1 - e^{-(c/128) \cdot \max(t, \ell) \log n}$.*

Proof (sketch). For any given round r, the probability that p is radio-enabled and set to broadcast, while exactly one process in P^* is radio-enabled and set to listen, while every other process in P and F is radio-disabled is at least $\min(\ell/(16t), 1/16)$. Thus, the probability that p fails to broadcast to q in all $c \cdot \max(t^3/\ell, \ell^2) \log n$ rounds is as desired, with high probability.

By counting the number of possible configurations of subsets, we conclude, by a union bound, that the desired property is achieved by the end of the first stage:

Lemma 2. *The following event holds w.h.p., for sufficiently large constant c: For every set P of active processes where $\ell/2 < |P| \leq \ell$, for every subset $P^* \subseteq P$ of size $\ell/8$, for every set F of at most t Byzantine processes, every rumor in P is received by some process in P^* by the end of sub-algorithm $\langle T_1, B_1 \rangle$.*

4.4 Stage 2: Exchanging

We now show that there is some subset of honest processes of size $\ell/8$ where every process in the set has received every rumor by the end of the second stage.

Recall that $\langle T_2, B_2 \rangle$ consists of $2\log n$ repetitions of two matrices T' and B', respectively. Given a set of honest processes P and a set of Byzantine processes F, we define an undirected graph $G(P, F, T', B')$ based on T' and B'. Each vertex in G represents a process, i.e., there are n vertices. For each odd column $r = 1, 3, 5, \ldots$ we add an edge (p, q) to graph G if the following hold: (1) For every process $p' \in F$, $T'[p', r] = 0$, i.e., every Byzantine process is radio-disabled. (2) For every process $q' \in P \setminus \{p, q\}$, $T'[q', r] = 0$, i.e., every other process is radio-disabled. (3) For processes p and q, $T'[p, r] = T'[q, r] = 1$, i.e., processes p and q are radio-enabled. (4) For process p, $B'[p, r] = 1$; for process q, $B'[q, r] = 0$.

This implies that process p succeeds in sending a message to process q in the round based on column r. Since column $r+1$ is defined in terms of column r, we conclude that q succeeds in sending a message to process p in the round based on column $r+1$. Thus, we consider the graph G to be undirected.

We argue that for all sets P and F, the graph $G(P, F, T', B')$ has a large subgraph with small diameter. We show this by examining the *expansion* of G. Following the definition from [30], we say that a graph G is an α-expander if it follows the following property:

Definition 1. *A graph $G = (V, E)$ is an α-expander if for every pair of subsets $W_1 \subseteq V$ and $W_2 \subseteq V$, where $|W_1| \geq \alpha$ and $|W_2| \geq \alpha$, there is some $p \in W_1$ and some $q \in W_2$ such that $(p, q) \in E$.*

We will argue that, with high probability, for every set P and set F, graph $G(P, F, T', B')$ is an $\ell/8$-expander:

Lemma 3. *With high probability, for sufficiently large c, for every P and F, graph $G(P, F, T', B')$ is an $\ell/8$-expander.*

Proof. Fix a set P of size $\ell/2 < |P| \leq \ell$ and a set F of size at most t. (We may assume, without loss of generality, that F is of size exactly t, as otherwise the adversary could add "silent" Byzantine processes without otherwise changing the execution.) We calculate the probability that $G(P, F, T', B')$ is a $\ell/8$-expander (after which we take a union bound over all possible sets P and F).

Fix arbitrary sets W_1 and W_2 of size at least $\ell/8$. We calculate the probability that there is some edge between W_1 and W_2 in $G(P, F, T', B')$. (We then take a union bound over all possible sets W_1 and W_2.) Specifically, for a given column of T' and B': (1) Every process in F is radio-disabled: with probability $\geq (1-\min(1/t, 1/\ell))^{|F|} \geq (1-1/t)^t \geq 1/4$. (2) Exactly one process in W_1 is radio-enabled: with probability at least $(\ell/8) \cdot \min(1/t, 1/\ell) \cdot (1-\min(1/t, 1/\ell))^{\ell/8-1} \geq (1/32) \cdot \min(\ell/t, 1)$. (3) Exactly one process in W_2 is radio-enabled: with probability at least $(\ell/8) \cdot \min(1/t, 1/\ell) \cdot (1-\min(1/t, 1/\ell))^{\ell/8-1} \geq (1/32) \cdot \min(\ell/t, 1)$. (4) The conditional event, under the assumption that one radio-enabled element in W_1 is chosen and one radio-enabled element in W_2 is chosen, that either the radio-enabled process in W_1 is set to broadcast and the radio-enabled process in

W_2 is set to receive, or the radio-enabled process in W_1 is set to receive and the radio-enabled process in W_2 is set to broadcast: with probability at least $1/2$.

Thus, for a given column, there is an edge between W_1 and W_2 with probability at least $\frac{1}{8} \cdot \left(\frac{\min(\ell/t,1)}{32}\right)^2$. Thus over $c \cdot \max(t^3/\ell, \ell^2) \log n$ odd columns (and their even counterparts corresponding to the edge in the reverse direction), the probability that there is no edge between W_1 and W_2 is bounded by:

$$\left(1 - \frac{1}{8} \cdot \left(\frac{\min(\ell/t,1)}{32}\right)^2\right)^{c \cdot \max(t^3/\ell, \ell^2) \log n} \leq \left(\frac{1}{e}\right)^{\frac{\min(\ell^2/t^2,1)}{8 \cdot 32^2} \cdot c \cdot \max(t^3/\ell, \ell^2) \log n}$$
$$= e^{-\frac{c}{8 \cdot 32^2} \max(t\ell, \ell^2) \log n} .$$

We now count the total number of sets W_1 and W_2, and also the total number of sets P and F. There are at most n^ℓ sets P with more than $\ell/2$ and at most ℓ elements. There are at most n^t sets F with (at most) t elements. There are at most 2^ℓ sets W_1, and similarly at most 2^ℓ sets W_2. In total, we can bound the number of sets P, F, W_1, and W_2 by: $n^\ell \cdot n^t \cdot 2^\ell \cdot 2^\ell = 2^{2\ell + (\ell+t) \log n} \leq 2^{3 \max(t,\ell) \log n}$. By a union bound over all possible sets, the probability that there exists any sets P and F such that $G(P, F, T', B')$ is not an $\ell/8$-expander is no greater than: $2^{3 \max(t,\ell) \log n} \cdot e^{-\frac{c}{8 \cdot 32^2} \max(t\ell, \ell^2) \log n} \leq e^{-\left(\frac{c}{8 \cdot 32^2} - 3\right) \cdot \max(t\ell, \ell^2) \log n}$. Thus, for sufficiently large c, w.h.p., graph $G(P, F, T', B')$ is a $(\ell/8)$-expander for every P and F.

We now apply the following, proven in [7], to conclude that there is some subset of P with small diameter:

Theorem 2. *Let G be an α-expander. For every set Q of at least 4α nodes, there is a subset $Q^* \subseteq Q$ of at least α nodes such that the subgraph of G induced by set Q^* has diameter of at most $2 \log n$.*

Corollary 1. *For every set P containing more than $\ell/2$ and at most ℓ processes, for every set F of size at most t, there is a subset $P^* \subseteq P$ containing $\ell/8$ processes such that P^* has diameter at most $2 \log n$ in $G(P, F, T', B')$*

We thus conclude that after executing $\langle T_2, B_2 \rangle$, there is some subset P^* of size $\ell/8$ such that every process in P^* knows every rumor:

Lemma 4. *The following event holds w.h.p., for sufficiently large constant c: For every set P with more than $\ell/2$ and at most ℓ processes, for every set F of at most t processes: after executing $\langle T_1, B_1 \rangle$ & $\langle T_2, B_2 \rangle$, there is some subset $P^* \subseteq P$ containing $\ell/8$ honest processes such that every rumor has been received by every process in P^*.*

Proof. Define P^* as per Corollary 1. Recall that P^* has diameter at most $2 \log n$. At the end of $\langle T_1, B_1 \rangle$, i.e., at the end of Stage 1, every rumor is known to some process in P^*, by Lemma 2. In every iteration of $\langle T', B' \rangle$ during Stage 2, rumors are propagated one hop through graph $G(P, F, T', B')$. Thus, during Stage 2, over $2 \log n$ iterations of $\langle T', B' \rangle$, every rumor stored in P^* at the end of Stage 1 is propagated to every other process in P^*.

4.5 Stage 3: Dissemination

In the third stage, the identified subset P^* distributes the rumors gathered during Stage 2 to the remaining processes in P. We have already shown that in Stage 1, each process in $P \setminus P^*$ successfully sends a message to some process in P^*. As $T_3 = T_1$ and B_3 is the entry-by-entry binary inverse of B_1, each successful sender in Stage 1 becomes a successful receiver in Stage 3 and *vice versa*, in every round. (Note: in the analysis of Stage 1, we considered only events/rounds in which there was only one sender and one receiver.) Thus, each process in $P \setminus P^*$ receives a message from some process in P^*. Thus we conclude:

Lemma 5. *The following event holds w.h.p., for sufficiently large constant c: For every set P with more than $\ell/2$ and at most ℓ of honest processes, and for every set F of at most t processes, after executing $\langle T_1, B_1\rangle$ & $\langle T_2, B_2\rangle$ & $\langle T_3, B_3\rangle$, each process in P has received all rumors of other processes in P.*

Combining the $\log k$ instances for exponentially decreasing ℓ, and applying the probabilistic argument to Lemma 5 for each instance $A(\ell, \ell/2)$, we conclude:

Theorem 3. *There exists a (k,t)-information exchange algorithm with running time $O(\max(t^3, k^2)\log^2 n)$.*

5 Extensions

5.1 Energy Usage

An advantage of the wTPM is that it enforces energy efficiency: in each around of $A(\ell, \ell/2)$, only a $\min(1/t, 1/\ell)$ fraction of honest processes are radio-enabled; the remainder cannot access their radios, saving power. Thus, we can show:

Lemma 6. *There exists a (k,t)-information exchange protocol with running time $O(\max(t^3, k^2)\log^2 n)$, where there are an average of $O(\lceil k/t\rceil)$ processes radio-enabled in each round.*

Proof. In protocol $A(\ell, \ell/2)$, in expectation, there are $\leq \min((k+t)/t, (k+t)/\ell) \leq 2\lceil k/t\rceil$ processes radio-enabled in every round. Thus, w.h.p., there are $O(\max(t^3, k^2)\log^2 n \cdot \lceil k/t\rceil)$ processes radio-enabled throughout the execution. Combining this with time complexity result of Lemma 5, which holds with high probability, and using the probabilistic argument, we obtain the claimed result.

For $t = \Theta(k)$ the per round energy usage is $O(1)$, on average, which is optimal.

5.2 Self-Verifying Rumors

We can somewhat improve the previous results when rumors are *self-verifying*, that is, when a process can distinguish a rumor that was initiated at an honest process from a rumor initiated at a malicious process (for example, via public keys or MACs). If a process can *stop early*, i.e., can cease executing the protocol when it believes it has received all available rumors, then we can obtain the following result:

Lemma 7. *There exists a (k, t)-information exchange protocol with running time $O(\max(t^3/k', k^2)\log^2 n)$ where an average of $O(1)$ honest processes are radio-enabled in each round, and k' is the actual number of active honest processes.*

Proof. Consider the protocol as before: $A(k, k/2)$ & $\cdots$ & $A(2, 1)$. A process terminates when it completes protocol $A(\ell, \ell/2)$, having already received at least $\ell/2$ rumors: there are clearly at least $\ell/2$ honest processes, and there is no need to continue executing the protocol for smaller ℓ. Since the running time for each $A(\ell, \ell/2)$ is $O(\min(t^3/\ell, k^2)\log^2 n)$, the claimed running time follows.

For energy: assume that $k' \leq k$ honest process are activated. On average, there are $\min((k'+t)/t, (k'+t)/\ell)$ processes enabled in each round. Since the protocol terminates no later than where $\ell > k'/2$, we conclude that on average there are no more than $O(1)$ processes radio-enabled in each round, by using the similar argument as in the proof of Lemma 6.

We conjecture that by carefully ordering the $A(\cdot, \cdot)$ instances, and by detecting when to stop, it may be possible to adapt to $|P|$, independent from k and n.

5.3 Continuous Communication

To this point, we have assumed that honest processes are all enabled in the same round, and that they each have exactly one rumor to distribute. In some situations, processes may be activated—and rumors injected—in any round. Consider, then, the following straightforward strategy: instead of executing each instance of $A(\ell, \ell/2)$ sequentially, interleave the executions. That is, divide time into blocks of $\log n$ rounds, and in a round r where $r \mod \log n = k$, execute one round of $A(2^{k+1}, 2^k)$. When a rumor is injected at a process p, it begins participating for a given $A(\ell, \ell/2)$ each time a new instance is started. (Here, a global clock or additional synchronization mechanism must be used). From this we conclude that there exists a *continuous* information exchange protocol where if there are $k \leq n$ rumors active in some round r, for an unknown value k, then all such rumors will be delivered no later than time $r + O(\max(t^3, k^2)\log^3 n)$.

6 Conclusions

We have shown that it is possible to design a wTPM that, by selectively enabling and disabling the radio, facilitates reliable communication. Surprisingly, as long as $t < k^{2/3}$, the resulting protocol is nearly as efficient, in time complexity, as optimal *oblivious information exchange* protocols for networks with no malicious devices. We have also shown a new lower bound indicating that when $k \leq \sqrt{n}$ or when $t \geq k^{2/3}$, the resulting bound is near optimal. The new protocol also provides improved energy efficiency, as existing oblivious solutions (in the model without malicious devices) need $O(k \cdot \min(k^2, n)\log_k n)$ energy [6,11].

An interesting open question is the performance of protocols such as the one in this paper in multi-hop networks. When there are no malicious devices, the time complexity of all-to-all communication is $\Theta(n \min(D, \sqrt{n}))$ [23]. When there are Byzantine processes, the situation is more complex, as we need to guarantee that honest processes form a connected component. Another question is whether, by relaxing the restrictions on the wTPM, allowing randomization or some adaptivity, we may be able to achieve even better performance.

References

1. Alistarh, D., Gilbert, S., Guerraoui, R., Milosevic, Z., Newport, C.: Securing your every bit: Reliable broadcast in byzantine wireless networks. In: Proceedings of the Symp. on Parallel Algorithms and Architectures (SPAA). pp. 50–59 (2010)
2. Awerbuch, B., Richa, A.W., Scheideler, C.: A jamming-resistant mac protocol for single-hop wireless networks. In: Proceedings of the Symp. on Principles of Distributed Computing (PODC). pp. 45–54 (2008)
3. Bar-Yehuda, R., Goldreich, O., Itai, A.: On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. J. of Computer and System Sciences 45(1), 104–126 (1992)
4. Bhandari, V., Vaidya, N.H.: On reliable broadcast in a radio network. In: Proceedings of the Symp. on Principles of Distributed Computing (PODC). pp. 138–147 (2005)
5. Bhandari, V., Vaidya, N.H.: On reliable broadcast in a radio network: A simplified characterization. Tech. rep., U. of Illinois at Urbana-Champaign (2005)
6. Bonis, A.D., Gasieniec, L., Vaccaro, U.: Optimal two-stage algorithms for group testing problems. SIAM J. on Computing 34(5), 1253–1270 (2005)
7. Chlebus, B., Kowalski, D.R., Shvartsman, A.A.: Collective asynchronous reading with polylogarithmic worst-case overhead. In: Proceedings of the Symp. on Theory of Computing (STOC). pp. 321–330 (2004)
8. Chlebus, B.S., Kowalski, D.R., Lingas, A.: The do-all problem in broadcast networks. In: Proceedings of the Symp. on Principles of Distributed Computing (PODC). pp. 117–127 (2001)
9. Clementi, A., Monti, A., Silvestri, R.: Optimal f-reliable protocols for the do-all problem on single-hop wireless networks. In: Proceedings of the International Symp. on Algorithms and Computation (ISAAC). pp. 320–331 (2002)
10. Clementi, A., Monti, A., Silvestri, R.: Round robin is optimal for fault-tolerant broadcasting on wireless networks. JPDC 64(1), 89–96 (2004)
11. Clementi, A.E.F., Monti, A., Silvestri, R.: Selective families, superimposed codes, and broadcasting on unknown radio networks. In: Proceedings of the twelfth annual ACM-SIAM Symp. on Discrete algorithms. pp. 709–718 (2001)
12. Dolev, S., Gilbert, S., Guerraoui, R., Newport, C.: Gossiping in a multi-channel radio network: An oblivious approach to coping with malicious interference. In: Proceedings of the Symp. on Distributed Computing (DISC). pp. 208–222 (2007)
13. Dolev, S., Gilbert, S., Guerraoui, R., Newport, C.: Secure communication over radio channels. In: Proceedings of the Symp. on Principles of Distributed Computing (PODC). pp. 105–114 (2008)
14. Dolev, S., Gilbert, S., Guerraoui, R., Kowalski, D.R., Newport, C., Kuhn, F., Lynch, N.: Reliable Distributed Computing on Unreliable Radio Channels. In: MobiHoc S^3 Workshop (2009)
15. Dolev, S., Gilbert, S., Guerraoui, R., Kuhn, F., Newport, C.: The Wireless Synchronization Problem. In: Proceedings of the Symp. on Principles of Distributed Computing (PODC). pp. 190–199 (2009)

16. Gilbert, S., Guerraoui, R., Newport, C.: Of malicious motes and suspicious sensors: On the efficiency of malicious interference in wireless networks. In: Proceedings of the Conference on Principles of Distributed Systems (OPODIS). pp. 215–229 (2006)
17. Gilbert, S., Guerraoui, R., Kowalski, D., Newport, C.: Interference-Resilient Information Exchange. In: INFOCOM. pp. 2249–2257 (2009)
18. Goussevskaia, O., Moscibroda, T., Wattenhofer, R.: Local broadcasting in the physical interference model. In: DIALM-POMC. pp. 35–44 (2008)
19. Komlos, J., Greenberg, A.: An asymptotically fast non-adaptive algorithm for conflict resolution in multiple access channels. IEEE Trans. on Information Theory pp. 302–306 (1985)
20. Koo, C.Y.: Broadcast in radio networks tolerating byzantine adversarial behavior. In: Proceedings of the Symp. on Principles of Distributed Computing (PODC). pp. 275–282 (2004)
21. Koo, C.Y., Bhandari, V., Katz, J., Vaidya, N.H.: Reliable broadcast in radio networks: The bounded collision case. In: Proceedings of the Symp. on Principles of Distributed Computing (PODC). pp. 258–264 (2006)
22. Kowalski, D.R.: On selection problem in radio networks. In: Proceedings of the Symp. on Principles of Distributed Computing (PODC). pp. 158–166 (2005)
23. Kowalski, D.R., Pelc, A.: Time complexity of radio broadcasting: adaptiveness vs. obliviousness and randomization vs. determinism. Theoretical Computer Science 333(3), 355–371 (2005)
24. Kranakis, E., Krizanc, D., Pelc, A.: Fault-tolerant broadcasting in radio networks. J. of Algorithms 39(1), 47–67 (2001)
25. Kuhn, F., Lynch, N., Newport, C., Oshman, R., Richa, A.: Broadcasting in radio networks with unreliable communication. In: Proceedings of the Symp. on Principles of Distributed Computing (PODC) (2010)
26. Meier, D., Pignolet, Y.A., Schmid, S., Wattenhofer, R.: Speed Dating Despite Jammers. In: Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS). pp. 1–14 (2009)
27. Moscibroda, T., Wattenhofer, R.: The complexity of connectivity in wireless networks. In: INFOCOM (2006)
28. Newport, C.: Distributed Computation on Unreliable Radio Channels. Ph.D. thesis, MIT (2009)
29. Pearson, S., Balacheff, B.: Trusted computing platforms: TCPA technology in context. Prentice Hall (2002)
30. Pippenger, N.: Sorting and selecting in rounds. SIAM J. of Computing 16, 1032–1038 (1987)
31. Strasser, M., Pöpper, C., Capkun, S.: Efficient Uncoordinated FHSS Anti-jamming Communication. In: Proceedings International Symp. on Mobile Ad Hoc Networking and Computing (MOBIHOC). pp. 207–218 (2009)
32. Strasser, M., Pöpper, C., Capkun, S., Cagalj, M.: Jamming-resistant Key Establishment using Uncoordinated Frequency Hopping. In: Proceedings of the Symp. on Security and Privacy. pp. 64–78 (2008)
33. Trusted Computing Group: Trusted platform module (tpm) specifications, `http://www.trustedcomputinggroup.org/resources/tpm_main_specification`
34. Willard, D.E.: Log-logarithmic selection resolution protocols in a multiple access channel. SIAM J. of Computing 15(2), 468–477 (1986)