Chorus: Scalable In-band Trust Establishment for Multiple Constrained Devices over the Insecure Wireless Channel

Yantian Hou, Ming Li
Department of Computer Science
Utah State University
Logan, UT 84322
houyantian@gmail.com,
ming.li@usu.edu

Joshua D. Guttman
Department of Computer Science
Worcester Polytechnic Institute
Worcester, MA 01609
guttman@cs.wpi.edu

ABSTRACT

Secure initial trust establishment for multiple resource constrained devices is a fundamental issue underlying wireless networks. A number of protocols have been proposed for secure key deployment among nodes without prior shared secrets (ad hoc), however so far most of them rely on secure out-of-band (OOB) channels (e.g., audio, visual) which either only work with a small number of devices or require auxiliary hardware. In this paper, for the first time, we design a solution that enables secure initialization of a group of wireless devices, which works merely within the wireless band. Our proposed solution is based on a novel physical-layer primitive for authenticated string comparison over the insecure wireless channel, called *Chorus*, which simultaneously compares the equality of fixed-length authentication strings held by multiple wireless devices within constant time. The Chorus achieves a key authentication property, which prevents an adversary from tricking each device to believe that all strings are equal when they are not, which is enabled by exploiting the infeasibility of signal cancellation and unidirectional error detection codes. Chorus can be employed as a foundation to provide in-band group message authentication (GMA) and group authenticated key agreement (GAKA), that does not require any prior shared secret. Specifically, we design two GAKA protocols based on Chorus and formally prove their security. The most appealing features of our proposed protocols include: minimal hardware requirement (a common radio interface and a button), minimal user effort (pressing a button on each device on average), nearly constant running time, thus they are scalable to a large group of constrained wireless devices. Through extensive analysis and experimental evaluation, we show the security and robustness of Chorus under a realistic attack model, and demonstrate the high scalability of our GAKA protocols.

WiSec'13, April 17-19, 2013, Budapest, Hungary.
Copyright 2013 ACM 978-1-4503-1998-0/13/04 ...$15.00.

Categories and Subject Descriptors

C.2.0 [**General**]: Security and Protection; C.2.1 [**Network Architecture and Design**]: Wireless Communication

Keywords

Wireless Network, Trust Establishment, Message Authentication, Key Agreement, Security Protocols, Physical-layer

1. INTRODUCTION

Wireless networks are increasingly adopted by the emerging cyber-physical systems or "Internet-of-Things" (IoT) [1]. These networks typically consist of a large number of interoperable smart wireless devices that are constrained in resources (power, hardware, and user interfaces), such like wireless sensors. Their applications range from e-healthcare systems, smart home/building, to boarder monitoring in homeland security. Data transmitted by such wireless networks usually contain privacy-sensitive or safety-critical information, which are subjected to eavesdropping and malicious manipulations. Thus a fundamental problem is to securely initialize multiple wireless devices by establishing secret keys to protect the communication among them from the scratch.

Previously, a number of key pre-distribution based mechanisms have been proposed for establishing initial trust in Wireless Sensor Networks (WSNs) [13, 8, 11]. However, they all assume that nodes are loaded with some form of shared key materials before initial use. This may be a reasonable assumption in some scenarios but certainly not for all, especially for *ad hoc* formed wireless networks in user-centric applications. For example, a patient who purchases tens of wearable medical sensors and wants to deploy a body area network on her body, or a property manager who wants to setup a building monitoring system with hundreds of sensor nodes. The main reasons are three-fold: 1) Constrained wireless devices usually lack necessary user interfaces (e.g, USB ports) to configure keys manually. Even if they do, manual key deployment is not scalable to a large group of devices. 2) Commodity sensor devices are not sold with pre-loaded secrets, while the manufacturers are not always trusted by the users. 3) A global public key infrastructure (PKI) is not likely to exist as wireless devices can be produced by various manufacturers. Therefore, an important research task is to design *secure ad hoc trust initialization solutions that do not presume shared secrets*, and satisfy the following three properties: highly usable, scalable, and compatible with constrained resources.

In order to achieve secure ad hoc trust initialization, the main technical challenge is message authentication over the (insecure) wireless channel. It is well-known that the simple Diffie-Hellman key exchange over the wireless channel suffers from the Man-in-the-Middle (MitM) attack, as the unprotected wireless signal is subjected to malicious modifications (such like bit flipping and message overshadowing [7]). Thus, in the past decade, various researchers have proposed secure channel based approaches to work around this problem, which is usually called "secure device pairing". It relies on the security (authentication) properties of some auxiliary out-of-band (OOB) channel in one way or another. For example, well-known OOB channels include USB connection [38], infrared [2], visual [5, 32, 33, 27, 29, 9, 23, 22], audio [15], faraday cage [18], etc. However, all these schemes require non-trivial human support, and the devices to be paired should possess common additional hardware such like USB ports, screen, keypads, LEDs, accelerometers, etc. This assumption is often strong and impractical, because all these schemes are often obtrusive to use and not scalable, and are against the global trend for device miniaturization. Moreover, it is commonly believed that human implemented OOB channels can only tolerate up to 10 devices [9, 23, 30]. The human-implemented OOB channel and requirement for advanced hardware have been major obstacles against the practical adoption of those protocols.

Thus, it is very desirable to find alternative solutions that avoid the use of OOB channels, and merely operate over the wireless (in-band) while do not rely on any additional hardware. Ideally, it should work compatibly with any constrained device with a common wireless radio interface and require minimum user participation. Next we review recent advances in wireless physical-layer based secure communication initialization (including authentication and secrecy).

1.1 Related Works

Physical-layer trust establishment. The idea of this category of approaches is to derive trust using some physical layer characteristics unique to each link that cannot be easily eavesdropped/forged by others. Existing schemes have been mostly tackling the two issues of *key generation* and *device authentication* separately. On the former, Mathur et. al. [26] and Jana et. al. [17] first proposed to utilize the randomness in received signal strength (RSS) to extract a secret key between two devices. On the latter, related methods include ensuring close device proximity [6, 25, 35, 37], location distinction [36] and device identification, etc. Unfortunately, almost all of these techniques require costly advanced hardware, such like multiple-antennas [6] and wideband transceivers [35, 25]. This limits their applicability on constrained devices. In addition, the security notions of device proximity and location distinction are quite different from "entity and message authentication". They cannot uniquely bind a message to its originating entity. Furthermore, it is non-trivial to combine key generation with device authentication techniques.

Message Authentication and Integrity Protection. The closest works to ours are integrity code (I-code) proposed by Čapkun et. al. [7], and Tamper-Evident Pairing (TEP) proposed by Gollakota et. al [14]. The I-code primitive protects the integrity of every message sent over the insecure wireless channel. It assumes the infeasibility of signal cancellation, and exploits unidirectional error detection codes to provide message tamper-evidence. It can be applied to key establishment, satellite signal authentication, etc. On the other hand, TEP is an in-band device pairing protocol for 802.11 devices, which uses a tamper-evident announcement (TEA) that protects the message integrity by embedding cryptographic authentication information (e.g., a hash) into the physical signals, such that any tampering with it will be caught by the receiver.

Though the concept of the above is appealing, there are two limitations. First, their security are both based on the infeasibility of energy cancellation. But they only achieve a weak security guarantee, since recently Pöpper et. al [34] proposed a stronger yet practical correlated signal cancellation attack using a pair of directional antennas. Second, it is difficult to apply them to securely initialize multiple constrained devices such like medical sensors due to the scalability issue. I-code and TEA are both one-to-one message authentication primitives suitable for pairwise communication. If implemented on a sensor platform with 250kbps transmission rate, an I-coded message requires 0.5s to transmit 50 bits on a ZigBee sensor platform, given a slot length of 5ms [7]. While in TEA, each synchronization packet must be at least 19ms long [14]. In addition, the number of "ON_OFF" slots is large (roughly equals a hash length). This yields a total of more than 750ms for each TEA. Thus, direct usage or simple extension of I-code or TEP is not scalable to a large group of constrained devices, whereas the *delay* can be critical in many real-world applications [24].

1.2 Our Contributions

In this paper, we aim at making ad hoc trust initialization work strictly in-band and scalable to a group of devices, by firstly introducing a novel physical-layer primitive called "Chorus" which achieves authenticated message comparison over the insecure wireless channel, and use it to construct secure group authenticated key agreement (GAKA) protocols. The Chorus is partially inspired from I-code and TEP in that we also exploit the infeasibility of signal cancellation and unidirectional error detection codes; however, we combine a similar idea to I-code with the concept of empirical OOB channels used in message authentication protocols, to achieve key authentication and confirmation. We observe that in most of the group message authentication protocols (MAPs), the role of OOB channel is to achieve secure comparison: an authentication string (AS) s_i is typically derived by each device from the protocol transcript (messages to be authenticated); when all nodes' ASes are equal to each other all devices should output accept, and whenever any nodes' ASes are not equal all devices should output reject.

Thus, the key idea of Chorus is to let N devices compare the equality of their fixed-length strings by simultaneously emitting specially encoded signals, such that any differences among the strings will be detected by all the devices. It only outputs 1 bit of information (accept - all strings are equal, or reject - some strings are different). Due to the unidirectional property of the wireless channel (attacker can only flip a "0" to "1" but not vice versa), changing the comparison result from reject to accept is impossible except negligible probability. This makes Chorus an ideal replacement for traditional OOB channels. Based on Chorus, we design secure in-band GAKA protocols, where all the messages to be authenticated are exchanged using the normal high-bandwidth wireless transmission, with only one run of Chorus in the end of the protocols. Therefore our protocols achieve greater scalability than previous solutions and are suitable for constrained devices.

Specifically we make the following contributions:

(1) We introduce "Chorus", a primitive for authenticated equality comparison of strings from multiple devices over the wireless channel in constant time. We make Chorus resilient to the strong correlated signal cancellation attack using uncoordinated frequency hopping. Through extensive analysis, we show that the proposed design satisfies authenticated comparison with high probabilistic guarantees for real-world constrained devices, under a relatively strong attacker model. Our defense against the correlated signal cancellation attack is also of independent interest.

(2) Using Chorus, we construct two group message authentication protocols (MAPs) based on AS comparisons, which naturally yields two GAKA protocols. Because Chorus neither require human interaction nor is limited in the length of AS to be compared, we show that the Chorus greatly simplifies the trust initialization protocol design, by achieving an optimal number of rounds (two) and minimal amount of user interaction. Our GAKA protocols both run in nearly constant time, regardless of the number of devices.

(3) We provide thorough security proofs for our proposed group MAPs (and GAKAs), implement and evaluate the proposed protocols on 24 real-world wireless sensor devices. Experimental results demonstrate that our protocols are scalable and usable.

2. PROBLEM STATEMENT

We consider an ad hoc group $\mathcal{G}$ of N constrained wireless devices/nodes that share a common radio interface (e.g., ZigBee or WiFi), which is chosen by a user and will be deployed to form a wireless network. The devices do not share any secret key materials a priori. We assume the user either knows or can count the group size N correctly. The goal for trust initialization is to establish authenticated shared secret keys among them in the setup phase to support secure communication afterwards, which may include group or pairwise keys.

2.1 Design Requirements

Here we first give informal definitions for security requirements in ad hoc trust initialization. (1) Key authentication: the derived secret key is authentic and the same among all the devices in the intended group $\mathcal{G}$. This essentially requires both entity and message authentication, which means each message sent by a legitimate device in $\mathcal{G}$ should be identical to what is received by its intended recipient(s), and an attacker should not be able to impersonate any legitimate device. (2) Key secrecy: the derived secret key is not known by an attacker. (3) Key confirmation: every device in $\mathcal{G}$ should confirm the successful derivation of the same key if the above two properties are satisfied.

For practicality, the scheme should satisfy the following: (1) High scalability and efficiency. It should support a large number of devices up to the order of hundreds or even thousands. Ideally, the running time of the protocol shall be nearly constant regardless of the group size. In addition, the per-device communication, computation and storage overhead must be small. (2) High usability. The solution should involve as little human effort as possible, and be intuitive to use by non-expert users. (3) Low hardware requirement. The solution shall be compatible with commercial-off-the-shelf (COTS) constrained devices with few interfaces (e.g., wireless sensors), and no advanced hardware such like multi-antennas or wide-band transceivers.

2.2 Attack Model and Assumptions

Our attack model is similar to the *Dolev-Yao model* [10], in that the adversary can take full control of the *normal wireless channel*, for example, it can eavesdrop, modify, remove, replay or inject messages (packets) transmitted over the wireless channel, and it can forge its identity (e.g., MAC address). However, the attacker cannot trivially disable the channel and block the transmission (e.g., using a Faraday cage). The signal cancellation attack is indeed possible for normal wireless channel as indicated in [34]. However, we will discuss ways to prevent this using specific mechanisms in more details in Sec. 3. The attacker can also jam the transmission so as to prevent the correct transmission of the information contained in a message. Further, we assume that the attacker is *computationally bounded*. We do not specifically address malicious denial-of-service/jamming attacks, which is an orthogonal problem; yet we do consider non-malicious interference from other nearby wireless devices operating within the same spectrum.

The attacker may possess powerful hardware such like software-defined radios and directional antennas. In addition, the attacker may have precise knowledge about the targeting environment and devices. For example, the exact location of each device, the channel status between each pair of wireless devices, and those between itself and the devices.

We assume that all legitimate devices are within direct communication range of each other. Furthermore, for key agreement protocols, we assume all the devices in $\mathcal{G}$ are benign (i.e., the manufacturer will not sneak spying devices when selling them). Otherwise if any device is compromised, no solution can achieve secrecy as it can send the key to an attacker. But if the protocol is merely for message authentication, this assumption is not necessary. Note that, our adversary model is relatively strong. Similar models have also been adopted by [7] and [14].

3. AUTHENTICATED COMPARISON OVER THE INSECURE WIRELESS CHANNEL

In this section, we first present the basic idea of authenticated equality comparison (AEC) over wireless channel (Chorus). Then we describe and analyze an enhancement which defends against known energy cancellation attacks.

3.1 The Basic Idea of Chorus

The Chorus does not directly authenticate a message that is sent and received over the wireless interface. Instead, it authenticates the equality comparison results for N bit strings over the wireless (derived from every node's messages as we will see in Sec. 4), as the truthful comparison of the equality of strings is the key to achieve authentication in group MAPs. We first define AEC.

DEFINITION 3.1. (Authenticated Equality Comparison) *Let there be $N \geq 2$ nodes that are within the communication range of each other, each holding a binary string $s_i, i \in (1, ..., N)$. AEC requires the following: 1) Non-spoofing: Whenever $\exists i, j, s_i \neq s_j$, then $\forall i \in (1, ..., N)$ outputs reject with high probability. 2) Correctness: If $\forall i, j \in (1, ..., N), s_i =$*

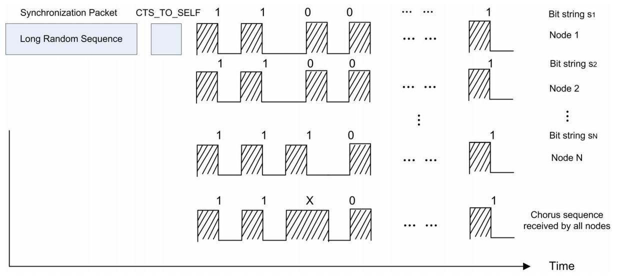


Figure 1: An example execution of basic Chorus using Manchester coding. Node N's string is different from others', which can be detected by all the N nodes. Shaded slots are random packets.

s_j, every node outputs accept. 3) Non-blocking: the AEC can neither be blocked from happening nor delayed, and the existence of it cannot be hidden.

That is, whenever a node outputs accept, it is assured (w.h.p.) that $\forall i, j, s_i = s_j$. Since only "accept" leads to successful message authentication in MAP protocols, the non-spoofing property is essential for security. AEC's properties are different from tamper-evidence in I-code [7] or TEA [14].

Examples of traditional OOB channels that satisfy AEC include the simultaneous LED blinking [33, 22]. However, we want to realize AEC over wireless. A straightforward idea is to let each of the N nodes broadcast its own s_i one-by-one using I-code; but this wastes bandwidth. Instead, we allow every node to broadcast their s_i simultaneously (Chorus), by encoding their strings using unidirectional error detection code and converting the encoded bits into ON-OFF keying. Detailed steps of the basic Chorus are as follows (instantiated using Manchester coding):

(1) It starts with a synchronization packet sent by one node (called coordinator), which contains random content and is longer than an usual packet. All other nodes detect the existence of this packet via threshold energy detection (i.e., the average RSSI is larger than a threshold T^2).

(2) After a short period when the sync packet ends, the coordinator broadcasts a short CTS_TO_SELF packet of length T_{cts}, which reserves the channel for the time period until Chorus concludes, by suppressing unwanted interference from other co-existing devices.

(3) Comparison phase: Each node i encodes its bit string s_i (of length l) using Manchester coding [40] to obtain an $2l$ bit string ($0 \rightarrow 01$ and $1 \rightarrow 10$), and map each encoded bit (1/0) into an ON/OFF slot respectively (of the same duration T_s). During each time slot $1 \leq j \leq 2l$, if it is an ON slot for a node, a short packet with random content is transmitted, simultaneously with everyone else (**"chorus"**); but if j is an OFF slot for a node, it remains silent and listens the channel. If $\forall 1 \leq j \leq 2l$, a node i does not detect energy in any of its own OFF slots, it outputs accept, otherwise outputs reject.

A sample timing diagram of a Chorus run is depicted in Fig. 1, where node N's string is "$1110\cdots$" which differs from others' strings ("$1100\cdots$") by one bit. The encoded strings are "$10101001\cdots$" and "$10100101\cdots$", respectively. This can be detected by all nodes (including N itself), because N will detect the aggregated signal of all other nodes during its 6th (OFF) slot, while all other nodes detect energy during their 5th (OFF) slot.

3.2 Security of the Basic Chorus

Different from I-code, in Chorus when each node sends its own signal, it cannot receive others' signals (we do NOT assume full-duplex transceivers). It seems that half of the information is lost. Thus the question is whether non-spoofing property can still be achieved. Next, we show that it is indeed the case as long as an adversary can only flip "0" to a "1" bit but not vice versa.

Claim 1: If signal cancellation is infeasible, the basic Chorus satisfies authenticated equality comparison.

First, for any two nodes' strings s_i, s_j that differ only in one bit, their respective Manchester encodings of that bit are either "01,10" or "10,01". Then, both nodes will detect a "1" during its OFF slot and output reject. Interestingly, each node can also decode its own ON slot as "1", and will obtain "11" which is not a correct codeword in Manchester code. Second, in general $\mathcal{G}$ can be divided into several subgroups $\mathcal{G}_1, ..., \mathcal{G}_k$ where strings in the same subgroup are equal but are pairwise different between different subgroups. For any node $i \in \mathcal{G}_{k'}$, its string s_i will differ from every one other subgroup's string by at least one bit. Thus it can be reduced to the two-node case.

Next, we consider an attacker that can only inject a signal generated by itself (type-I signal cancellation attacker).

Lemma 3.1. *The realization of basic Chorus is secure against the type-I signal cancellation attacker.*

Proof. The correctness is obvious.

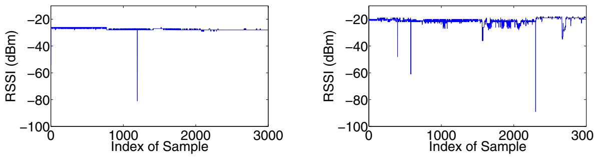
According to Proposition 7.1 in [14], if the transmitted signal is unpredictable and the sender and receiver are within communication range, a type-I attacker cannot cancel the signal energy at the receiver even if she knows the channel function $h(t)$ between the sender and receiver and is perfectly synchronized with the sender. This is because the attacker needs to generate a signal with exactly the same content but the inverse phase in advance, which is infeasible.

Similarly, in our Chorus realization, the aggregated energy of packets sent during a "ON" slot cannot be canceled at any receiver by the adversary even if she knows the exact channel status. Because, during the jth slot, each node i in the chorus set $\mathcal{C}_j$ (an ON slot for them) sends a random packet denoted as signal $s_i(t)$, and the aggregated signal received by another node whose encoded bit $s'_j = 0$ is: $\sum_{i \in \mathcal{C}_j} (s_i(t) \star h_i(t)) + n(t)$ (where $n(t)$ is Gaussian noise), which is still a random signal. Thus, the soundness follows.

In addition, an adversary *cannot block or hide the existence* of a Chorus because it cannot cancel the energy of the sync packet by generating the same signal with an inverted phase by itself. Thus the non-blocking property follows. Note that, we do not consider denial-of-service attack as it does not affect authentication, i.e., flipping "0" slot to "1" only causes all the nodes to abort. □

Remarks. Note that, the above reasoning assumes that the aggregated chorus signal of ON slots of the same subgroup does not cancel out itself at receiving node i. But one may wonder whether this is the case in reality. Next, we show that the self-cancellation only happens with very small probability.

Intuitively, the more nodes transmit simultaneously, the higher the average total received power. This is similar to the phenomenon that more people speaking simultaneously



(a)

(b)

Figure 2: The self-cancellation effect. (a): 2 senders scenario. (b): 8 senders scenario.

in a room will more likely induce a louder sound. Specifically, we can model the signal received from each node as $\cos(\omega + \theta_i)$, where θ_i is a random phase delay. The superposition signal is $x_c(t) = \sum_{i=1}^{N} \cos(\omega t + \theta_i) = B_n \cos(\omega t + \tau)$, where B_N is the amplitude:

$$B_N^2 = N + \sum_{i=1}^{N} \sum_{j=1}^{N} \cos(\theta_i - \theta_j) \tag{1}$$

From Eq. (1), we can easily derive the expectation $\mathbb{E}[B_N^2)] = N$, which verifies our intuition of average power. To obtain the actual probability of self-cancellation (P_{sc}), we carry out two sets of experiments, in which $N(2$ or $8)$ sensor nodes transmit their Chorus signals simultaneously, and the receiver samples 3 RSSIs during each slot.

From Fig. 2, it can be seen that: (1). The average power of the aggregated received signal increases as N increases. (2). The signal self-cancellation phenomenon does exist; however, the occurrence of severe attenuation (-80dB) is very rare. We can derive an empirical value of P_{sc} from the experiment results (1 out of 1000 slots). In evaluation section, we will show that the self-cancellation does NOT affect the security of Chorus as P_{sc} is small.

3.3 Defending Against Powerful Signal Cancellation Attacks

A correlated signal cancellation attack is recently shown by Pöpper et. al. [34] to be practical, where the attacker does not generate its own signal. It is based on signal relaying, i.e., the attacker (Lucifer) is located at a distance away from both the sender (Alice) and receiver (Bob), and utilizes a pair of directional antennas to relay the sender's signal to the receiver. If he creates a phase delay for the carrier signal on the relay channel that is multiple of π and with the same signal amplitude, the received signal strength can be completely attenuated (see Fig. 4(a)). This attack doesn't depend on the packet content and modulation, while it mainly works under stable and predictable channel environments (e.g., static indoor scenarios). So it is important to consider this type of attack (we refer as Type-II) in the Chorus's design.

To defeat this type of powerful attack, we observe that the key factor for Lucifer to succeed is to create a phase difference of $\Delta\phi = (2k-1)\pi, k = 1, 2, \ldots$. Assuming the processing delay at Lucifer is negligible, we have:

$$\Delta\phi = \frac{2\pi f \Delta d}{c}, \tag{2}$$

where $\Delta d = d(A, L) + d(B, L) - d(A, B)$ is the distance difference between the relay channel and the direct channel of Alice and Bob, f is carrier frequency, and c is speed of light. Making $\Delta\phi \neq (2k-1)\pi$ will prevent the signal from being completely cancelled; but since one cannot predict the attacker's location, the only parameter Alice and Bob can control is f.

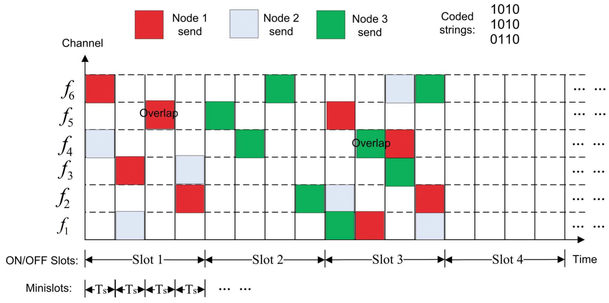


Figure 3: An example execution of the comparison phase of FH-Chorus.

3.3.1 *The Enhanced Chorus Scheme with Frequency Hopping (FH-Chorus)*

We propose to make novel use of uncoordinated frequency hopping (UFH) [39] to protect Chorus from the Type-II attack. The basic idea is to make the probability of cancellation arbitrarily small by hopping over multiple frequencies.

Suppose the available spectrum for the radios of all the devices consists of n consecutive channels $f_1, \ldots, f_n$ with the range being Δf. In FH-Chorus, each ON/OFF slot is extended to m minislots of the same duration T_s. For each local ON slot of node i, i randomly hops among the set of available channels for m minislots, and sends a random packet during each minislot. For each local OFF slot of node i, it also randomly hops a channel for each minislot (at the same hopping rate), and listens on each channel. For each node, in each OFF slot, as long as it detects energy during at least one of the minislots, it will output reject. Otherwise, if it does not detect energy in any OFF slot, it outputs accept. Note that, in the comparison phase the packets do not contain meaningful content.

Synchronization is a little more complicated due to the need of frequency rendezvous. But again we can use UFH. Suppose k is the coordinator which randomly hops the channel with slot length T_s for a total of m slots, in each it sends a sync packet containing a *counter* which increases from one to m, along with some random padding bits. All other nodes randomly hop the channel with a longer slot length T_s' (not synchronized initially). If a node receives a sync packet on any channel, it decodes the counter and starts chorus after $(m - counter) * T_s + T_{cts}$ (seconds). In this way, if m is large enough such that every node receives at least one packet with high probability considering the energy cancellation attack, all nodes will be synchronized. The UFH not only prevents the sync packet being cancelled, but also naturally provides some degree of resistance to jamming/interference.

The CTS_TO_SELF packet does not need to be protected as it does not affect security.

A toy example of the comparison phase of FH-Chorus is depicted in Fig. 3, where there are three nodes 1, 2, 3 with $s_1 = s_2 = 11, s_3 = 01$, $m = 4$, and number of channels is 6. As long as node 3 hops to one of node 1 or 2's channels in slot 1, and nodes 1 and 2 hop to one of node 3's channels in slot 2, the bit difference will be detected by all nodes.

3.3.2 *Analyzing the Attack Resilience of FH-Chorus*

We analyze the successful signal detection probability P_d

P_d	successful signal detection probability at Bob
P_{nc}	probability of signal not being cancelled within one minislot by attacker across $\triangle f$
$\triangle f$	total hopping frequency range
$\triangle d_0$	minimum distance difference of attacker
h	total number of FH channels
b	number of channels cancelled by the attacker
m	number of FH minislots in one slot
B	amplitude of the signal from A received by B
T	receiver's signal detection threshold (amplitude)
η	cancellation margin: B^2/T^2(in dB)

Table 1: Main notations.

at a node i if its string is different from some other nodes' strings. We look at the worst case where only one bit is different; in general, when multiple bits are different, P_d only becomes larger which benefits the receiver. So we constrain our analysis to a j-th bit. The nodes in $\mathcal{G}$ can be classified into two subgroups: those with $s_j = 0$ (denoted by $\mathcal{G}_0$) or $s_j = 1$ ($\mathcal{G}_1$). P_d is affected by the size of the group (e.g., $\mathcal{G}_0$) that the node is not in. Again, we consider the worst case where there is only one other node with a different string. This is because, due to random FH, i is more probable to detect energy in at least one minislot when there are many senders than only one sender.

Thus, we first focus on two nodes (Alice and Bob), given that their strings differ by one bit. Then we show that its result can be regarded as a lower bound to the detection probability when there are multiple nodes. The attacker's successful cancellation probability is $P_a = 1-P_d$. Note that, to spoof all nodes in $\mathcal{G}_1$, the attacker needs to cancel out the energy of all $|\mathcal{G}_0|\cdot|\mathcal{G}_1|$ transmission links from nodes in $\mathcal{G}_0$ to $\mathcal{G}_1$, which requires at least $N-1$ pairs of directional antennas that increases with group size. Thus, it is reasonable to assume only one attacker for each link.

Detailed Model and Assumptions. We assume that the attacker can always choose an optimal location and antenna gain to achieve the maximum cancellation probability (minimize the RSSI of the signal at Bob), given the UFH strategy adopted by Alice and Bob. Lucifer also knows the channel status between (A, L) and (L, B). However, there are several practical restrictions for the attacker: (1) Lucifer cannot be located very close to the device group[1]. Its distance difference to Alice and Bob is: $\Delta d = d(A, L) + d(B, L) - d(A, B) \geq \Delta d_0$, which is the outer space of an ellipse (an *unsafe region*). (2) Lucifer cannot change his location in a short FH minislot (e.g., 5ms). (3) Lucifer is not capable of doing any real-time computation. In a word, he will choose a location to stay, and relay the signal sent by the legitimate transmitter.

Main Analytical Results. Our main result is, P_d can be made arbitrarily close to 1, with an increasing number of minislots m. To satisfy a given P_d, the required m can be derived as a function of Δf and Δd_0. In reality, the feasible hopping range is often fixed, so we focus on the relationship of m and Δd_0.

(1). The Two-Node Case.

Let the signal received by Bob directly from Alice be $B\cos(2\pi f)$. Lucifer (at a fixed location) relays the signal such that the received relay signal by Bob is $B\cos(2\pi f_1 - \Delta\phi)$. The amplitude of the superposition signal is

$$x(f, \Delta d) = \sqrt{2B^2 + 2B^2\cos(\Delta\phi)} \tag{3}$$

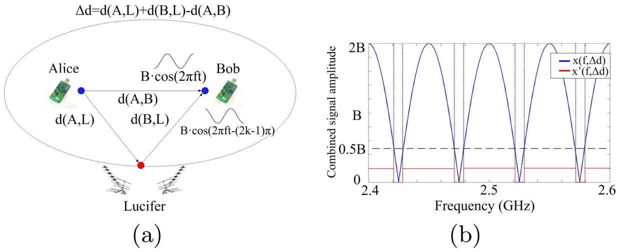


Figure 4: (a): illustration of correlated signal cancellation attack. (b): illustration of the superposition signal amplitude $x(f, \Delta d)$, and the corresponding wave function $x'(f, \Delta d)$ when choosing $T = 0.5B$.

in which $\Delta\phi = 2\pi f \Delta d/c$. The relation of $x(f, \Delta d)$ with f is illustrated in Fig.4(b). We can see that the attacker cannot cancel the signal at every frequency. By setting the detection threshold as T, the non-cancellation probability P_{nc} can be derived as $P_{nc} = \int_{f_a}^{f_b} x'(f, \Delta d)\, \mathrm{d}f/(f_b - f_a)$, in which $x'(f, \Delta d) = 1$ if $x(f, \Delta d) > T$, and $x'(f, \Delta d) = 0$ otherwise. For a fixed Δf, the P_{nc} is the ratio of the total length of non-cancelled frequency segments (L_1) to Δf. Combined with Eq. (3), we can also see the period of $x(f, \Delta d)$ monotonously decreases as Δd increases (intuitively, the larger the distance difference Δd, the more sensitive the phase difference $\Delta\phi$). Thus, as $\Delta d \to \infty$, this ratio will converge to some non-zero value.

As is shown by the simulation results in Fig. 5, P_{nc} first fluctuates and then converges as Δd increases. This implies that given any Δd_0, we can find a lower bound of P_{nc}: $P_{nc,min}$ for all $\Delta d \geq \Delta d_0$. Next we prove the existence of this lower bound in Theorem. 3.1.

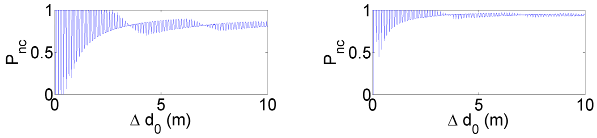
Theorem 3.1. (Lower Bound of P_{nc}) *Given a hopping range Δf, for any Δd_0, there is a lower bound $P_{nc,min} = (\lfloor x \rfloor - 1)L_1/(\lfloor x \rfloor L_0 + (\lfloor x \rfloor - 1)L_1)$, such that $\forall \Delta d \geq \Delta d_0$, we can guarantee $P_{nc} \geq P_{nc,min}$, where $L_1 = (\arccos((T^2 - 2B^2)/2B^2)c)/\pi\Delta d_0$, $L_0 = (-\arccos((T^2 - 2B^2)/2B^2)c + \pi c)/\pi\Delta d_0$, and $\lfloor x \rfloor$ is the maximum integer s.t. $\lfloor x \rfloor L_0 + (\lfloor x \rfloor - 1)L_1 \leq \Delta f$.*

The proof is in our technical report [16]. The above is a loose bound. In fact we can obtain a better actual lower bound $P_{nc,min}$ using numerical simulation. From Fig. 5, given any Δd_0, we can search all Δd within $[\Delta d_0, \Delta d_0 + W]$, where W is a large enough range. The minimum P_{nc} within this range will be taken as $P_{nc,min}$ for all $\Delta d \geq \Delta d_0$. We show both the theoretical and actual lower bounds of P_{nc} in Fig. 6. Because $\Delta d \geq \Delta d_0$ is an ellipse, for any Δd_0, we can guarantee a minimum P_{nc} by making sure that the attacker is out of this ellipse.

Next, we will derive the minimum number of minislots required to guarantee any P_d for FH-Chorus, based on the $P_{nc,min}$ derived above. Given a P_{nc}, we can obtain the cancellation probability on each FH channel. If the FH range includes h channels which span from f_a to f_b, then the maximum number of channels that can be cancelled by an attacker is $b = \lceil h(1 - P_{nc,min}) \rceil$. After deriving b, we can obtain the minimum number of minislots m to guarantee a given P_d:

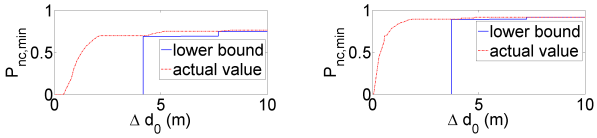
Theorem 3.2. (Minimum Number of FH Minislots) *Given a hopping range $\triangle f$ and Δd_0, an arbitrary detection probability threshold $P_d \to 1$ can be achieved by using a mini-*

[1] This is reasonable, as in practice an attacker with directional antennas can be easily spotted by the user.



(a) $\eta = 5dB$ (b) $\eta = 15dB$

Figure 5: Non-cancellation probability w.r.t. $\triangle d_0$



(a) $\eta = 5dB$ (b) $\eta = 15dB$

Figure 6: Lower bound of P_{nc}, given a $\triangle d_0$; $\Delta f = 80MHz$

mum number of FH minislots $m \geq \log_{(h^2-h+b)/h^2}(1-P_d)$, *in which* h *is the number of FH channels, and* $b = \lceil h(1 - P_{nc,min})\rceil$.

The proof is in [16]. From this we also found we can choose $h^* = \min\{2b, h\}$ to minimize m, therefore enhancing efficiency. Fig. 7 shows an example of the minimum number of minislots needed given a P_d and hopping range $\Delta f = 80MHz$. It can be seen that, if the cancellation margin η is larger, the required m is smaller to satisfy the same P_d.

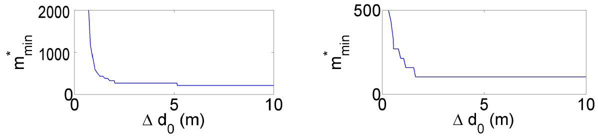
(2). The General-Group Case. In the group case, the superposition signal sent simultaneously by multiple nodes is actually easier to be detected, even with self-cancellation. This is because, with random frequency hopping, a larger group of senders increases the probability of any of their signals being detected by a receiver. We will prove this intuition in the following.

Theorem 3.3. (Minimum Number of Hopping Slots in the Group Case) *Assume in the two-node case, given a threshold* P_d, Δd_0 *and* Δf, *the required minimum number of minislots is* m^*. *Then in the group case, using the same* m^*, *and* Δf, *the probability of successful signal detection is lower bounded by* P_d, *if the attacker is located outside of the ellipses defined by every pair of nodes* ($\forall i, j, \Delta d_{ij} \geq \Delta d_0$).

The proof is in [16]. To sum up, given a feasible FH range Δf, we can satisfy any signal detection probability threshold P_d close to 1, by guaranteeing $\triangle d > \triangle d_0$ and its corresponding minimum number of minislots $m \geq \log_{(h^2-h+b)/h^2}(1 - P_d)$. As an example, when $\Delta d_0 = 2m$, $\Delta f = 80MHz$, $\eta = 15dB$, we obtain $m = 105$.

3.3.3 Robustness of Chorus

Chorus can be easily made robust against non-malicious interference. For example, the interference from 2.4G WiFi AP/laptop can generate a signal large enough to cause a false alarm in Chorus. Our first approach is to set a large enough RSSI detection threshold (e.g., −65dBm). This can filter out most of the ambient RF noise and cross-technology interference. In extreme cases (when the WiFi station is quite close) the interference could still induce a higher RSSI value than the threshold. However, through experiment (done in a campus building) we actually find that this type of strong interference is rare. We can only detect a small number



(a) $\eta = 5dB$ (b) $\eta = 15dB$

Figure 7: Minimum number of FH minislots required assuming different $\triangle d_0$; $\Delta f = 80MHz$

of false "ON" minislots in most cases (smaller than 3 when $m = 105$). Meanwhile, we observe that the expected number of true "ON" minislots is much larger (e.g., around 25 when $m = 105$) when any two nodes' comparison bit strings are different.

Therefore, our second approach to filter interference in FH-Chorus, is to impose a "ON" slot number threshold T_n (e.g 3) on the number of minislots where energy is detected, and ignore the "ON" alarm if that number is below T_n. In the following we analyze the robustness of Chorus given a chosen T_n. The tradeoff is that, the larger T_n, the higher the robustness, but the lower the detection probability P_d.

Theorem 3.4. (Chorus's Robustness) *Given a "ON" slot number threshold* T_n, *the new successful signal detection probability* P'_d *is:* $P'_d = P_d - \sum_{i=1}^{T_n} \binom{m}{i} \cdot ((1-P_r)^{m-i}) \cdot ((P_r)^i)$, *in which* $P_r = (1/h) \cdot (1 - b/h)$.

The proof is in [16]. For $T_n = 3$, the probability P'_d reduces to 0.9999 from 0.999999, when $\Delta d_0 = 2m$, $\Delta f = 80MHz$, $\eta = 15dB$.

3.3.4 Practical Issues

Two main practical issues in Chorus are: synchronization, and the message expansion (encoding overhead). For the former, the analysis of successful sync packet reception probability is similar to Sec. 3.3.2, and the same probability as P_d can be achieved using parameters in the previous example. For the latter, an optimal unidirectional code – Berger code can be used instead of Manchester code, where the check value size is $\lceil log_2(l+1)\rceil$ bits for a message of l bits. We omit the details here due to lack of space.

3.4 Comparison with Empirical Channels

We model the properties of Chorus and compare with two major state-of-the-art types of authenticated empirical channels in Table. 2. The Dolev-Yao channel is taken as a reference. Examples of weak empirical channel are non-face-to-face human communication such like voice mail. Examples of weak empirical channel include face-to-face conversation. However, existing unspoofable empirical channels require non-trivial human support [29].

Chorus achieves comparable security properties as strong empirical channel, but uses in-band wireless communication without user involvement. We already showed that Chorus achieves unspoofability and non-blocking (with high probability), since energy cancellation can be made infeasible. Delaying also won't work as the nodes are assumed to be in each other's communication range. The attacker's ability to "create" and "replay" is more subtle. In a wireless channel, nothing prevents an attacker from injecting/replaying a sync signal and initiating a Chorus process among the legitimate nodes (nodes cannot distinguish where the signal comes from). However, when we integrate Chorus into GAKA protocols, such a problem can be easily avoided since

Attacker capabilities	Create	Modify	Delay	Block	Replay	Overhear	In-band
Dolev-Yao [10]	√	√	√	√	√	√	√
Weak empirical [41, 31, 29]	×	×	√	√	√	√	×
Strong empirical [41, 31, 29]	×	×	×	×	×	√	×
Chorus	×*	×	×	×	×*	√	√

Table 2: Comparison of Chorus with existing authenticated channels. *: explained in text.

a legitimate coordinator node will always send out its sync signal during each protocol run, so that all nodes will abort if they hear it more than once (similar to the synchronization mechanism in [14]).

4. IN-BAND TRUST INITIALIZATION PROTOCOLS FOR GROUPS OF DEVICES

Chorus enables the design of truly scalable in-band trust initialization protocols. Next we present two example protocols representing two extremes in the design space that use short and long authentication strings, respectively. As any GAKA protocol can be reduced to a message authentication protocol (MAP) [29], we will illustrate how to design efficient MAPs based on Chorus and focus on the group setting.

4.1 In-band MAP with Short Authentication String (SAS)

When the AS to be compared is a short string (e.g., 16 bits), there exists several traditional SAS-comparison based protocols both in the two-party setting [41, 19, 5] and the group setting [29, 20, 33]. The basic protocol structure follows the principle of "joint commitment before knowledge" (JCBK) [29], which consists of three rounds: 1) Commitment; 2) Decommitment; 3) Computing SAS and compare it over an OOB channel. We show that in-band group MAPs with short SAS (GMS) can be designed by using Chorus as a primitive to replace OOB channel-based SAS comparison.

Our GMS protocol is based on the SHCBK protocol proposed by Nguyen and Roscoe [28, 29]. The reason to choose their protocol is, unlike most of the group protocols [29, 20, 33], it does not need a heavy-weight non-malleable commitment scheme thus is much more computationally efficient. It uses a cryptographic hash ($H(x)$), and a *Digest* function:

Definition 4.1 (Digest function). *$Digest(r, m)$: $\{0,1\}^L \times \{0,1\}^n \rightarrow \{0,1\}^\ell$ is a mapping where m is the message to be digested and r is a key. It shall have two properties:*

(1) (ϵ_u key-based uniformity) for any fixed m and y, $Pr_{r \in_R \{0,1\}^L}[Digest(r, m) = y] = \epsilon_u$.

(2) (ϵ_r no uniform compensation) for any fixed θ and $m \neq m'$, $Pr_{r \in_R \{0,1\}^L}[Digest(r, m) = Digest(r \oplus \theta, m')] = \epsilon_r$.

A concrete construction is given in [28] based on matrix product, where the ideal properties are achieved: $\epsilon_u = \epsilon_r = \frac{1}{2^\ell}$. Usually the output of a digest function is a short string, e.g., $\ell = 16$ bits. It is similar to a universal hash function, but the latter does not concern collision resistance under different keys.

Our GMS protocol is outlined in Fig. 8. Since the group is formed in an ad hoc way, the devices do not know the group $\mathcal{G}$ in advance. So in step 0, the user should count and enter the group size into a designated coordinator node. Steps 1 and 2 are the same with the SHCBK protocol. In round 3, each node computes an SAS and compare it via Chorus. $\mathcal{G}_i$ is the set of group IDs received by node i. Output confirmation is done by the coordinator sending another sync signal to all nodes (which cannot be removed by the attacker). This is because only the coordinator knows the correct group size. There is no need for the user to press buttons again.

```
Input: message $INFO_i = \{i, m_i\}$ , $i \in (1, ..., N)$
0. User picks coordinator $k$ and enters group size $N$.
1. $\forall i \rightarrow_N \forall i'$: $INFO_i, H(i, r_i)$
   $r_i$ is a random number;
2. $\forall i \rightarrow_N \forall i'$: $r_i$
3. $k$ initiates Chorus by sending sync packet;
   $\forall i \Leftrightarrow_{Chorus} \forall i'$: $Digest(\oplus_{j \in \mathcal{G}_i} r_j, \{INFO_j\}_{j \in \mathcal{G}_i})$
   For $k$, if all SASes match and $|\mathcal{G}_k| = N$, accept.
   Otherwise output fail, send sync signal again.
   $\forall i \neq k$, if detected sync more than once, abort.
   Otherwise output accept.
Note: "$\rightarrow_N$": normal wireless channel;
"$\Leftrightarrow_{Chorus}$": Chorus channel.
```

Figure 8: In-band Group MAP with SAS (GMS)

Protocol Synchronization. Clearly, the GMS strictly follows the JCBK principle, where after round 1 all nodes are committed to their final SAS values. Synchronization is important to ensure JCBK. This can be done via several ways, for example, a strict message order can be imposed ([33]) where nodes with smaller ID send first, and the coordinator is the last one. In addition, all the nodes can set a timer to ensure the reception of sync packet in Chorus.

Security. We stress that the attacker takes full control of the normal wireless channel but not Chorus. The original SHCBK protocol was proven secure in [28], which uses a strong empirical OOB channel. We prove the security of GMS in Sec. 5. The intuition is that, the adversary cannot make all nodes' SASes equal by modifying the messages sent by legitimate nodes except with negligible probability. On the other hand, if SASes are not equal, Chorus ensures that all the nodes will reject with high probability.

4.2 In-band MAP with Long Authentication String (LAS)

Alternatively, if we let the input authentication string in the Chorus be longer, the protocol structure can be simplified into two rounds, eliminating the need for commitment/decommitment (see Fig. 9). Using a collision-resistant hash function $H(x)$ (for example, SHA-1 with 160 bits output), the devices in $\in \mathcal{G}$ can compute and compare a long authentication string (using Chorus) whose inputs include all the received messages. Our GML protocol can be viewed as an extension of Vaudenay's non-interactive two-party message authentication protocol (Vau05) [41] to the group case. Its security can also be reduced to the collision-resistance of the hash function. However, the Vau05 protocol requires the use of a secure OOB channel, and the human work to send/compare a 160-bit LAS is quite heavy.

Remarks. Correct device counting is required for secu-

```
Input: $INFO_i = \{ID_i, m_i\}$ , $i \in (1, ..., N)$
0. User picks coordinator $k$ and enters group size $N$.
1. $\forall i \rightarrow_N \forall i'$: $INFO_i$
2. $k$ initiates Chorus by sending sync packet;
  $\forall i \Leftrightarrow_{Chorus} \forall i'$: $H(\{INFO_j\}_{j \in \mathcal{G}_i})$
  The rest is the same with the GMS.
```

Figure 9: In-band Group MAP with LAS (GML)

rity purposes in our protocols, because the authenticated group member IDs are not known in advance. An incorrect count may facilitate an attacker to join the group. In fact, this is common underlying any ad hoc group MAP protocol ($N > 2$) [29, 20, 33]. However, we note that this adds only *minimum user effort*, since counting can be done while deploying the devices. If the network size is known in advance, or there is a machine counter, we can easily scale up to hundreds of nodes (unlike previous OOB-based methods). The count input can be easily implemented in sensors with buttons; otherwise it only requires one coordinator device which has richer interfaces such like a mobile phone.

From MAPs to GAKA Protocols. In the above protocols, if we change the message to be authenticated (m_i) of each node to a public number $X_i = g^{x_i}$ where x_i is a secret random number and g is a generator in $\mathbb{Z}_p$, then both of our protocols can establish $(N^2 - N)/2$ pairwise keys securely using Diffie-Hellman key exchange. Or, if a contributory group key agreement scheme is adopted (e.g.,[12]), the GMS and GML become group authenticated key agreement protocols (GAKAs) that establish a group key.

5. PROVING SECURITY PROPERTIES

Next we formally prove the security of the GMS and GML protocols. We define secure message authentication of a MAP based on the well-known Bellare-Rogaway model [4], which introduced the notion of "matching conversations" [4]. Essentially, if all the parties have matching conversations, all messages transmitted by them will be received unaltered, i.e., authentically.

Definition 5.1 (Secure Message Authentication). *We say that Π is a (ϵ, T)-secure message authentication protocol with a group of participants $\mathcal{G}$ ($|\mathcal{G}| \geq 2$), if for any T-time adversary $\mathcal{A}$,*

(1) (Matching conversations $\Rightarrow$ acceptance) If all pairs of parties in $\mathcal{G}$ have matching conversations, then all parties accept.

(2) (Acceptance $\Rightarrow$ matching conversations) Letting $Adv_{\Pi}(\mathcal{A}) = Pr[\textsf{All-accept} \wedge \textsf{No-Matching}]$, where No-Matching refers to the event that the conversations are not matching, we have $Adv_{\Pi}(\mathcal{A}) \leq \epsilon$.

5.1 Security of the GMS Protocol

We have the following result, stated in concrete security guarantee. We use δ to denote the probability that all devices output accept when their SASes are not equal in Chorus.

Theorem 5.1. *Assume that all devices in $\mathcal{G}$ are within range, group count is correct, and the coordinator is uncompromised. If the digest function satisfies $2^{-\ell}$-key-based uniformity and $2^{-\ell}$-no uniform compensation, and the hash function $H()$ is (ϵ_h, T_h)-preimage resistant and (ϵ_b, T_b)-second preimage resistant, the GMS is $(2^{-\ell} + \epsilon_h + 2\epsilon_b + \delta, T_b + T_h)$-secure.*

The proof is in our technical report [16]. Essentially, this result bounds the adversary's (one-shot) deception probability to that of random guessing, as ϵ_h and ϵ_b are far smaller than $2^{-\ell}$ and can be neglected. In addition, δ is upper bounded by the attack success probability under single-bit difference: $P_a = 1 - P_d$ (derived in Sec. 3.3.2), which is around 10^{-6}. Thus, when SAS length $\ell = 16$ (32 slots in Chorus) this is about 10^{-5} (this can be freely tuned by the designer).

5.2 Security of the GML Protocol

Theorem 5.2. *Assume that all devices in $\mathcal{G}$ are within range, group count is correct, and the coordinator is uncompromised. If the hash function $H()$ is (ϵ_c, T_c)-collision resistant, then the GML is $(\epsilon_c + \delta, T_c)$-secure.*

The proof is in [16]. In practice, $\epsilon_c \approx 2^{-80}$ if we use SHA-1 with output of 160-bits. This is much smaller than δ, so δ dominates the adversary's success probability in GML.

Reducing the number of minislots. Note that in GMS, P_a was computed by considering the worst case that only one LAS bit differs. However, we show that in GML, if the output of the cryptographic hash can be regarded as an ideal random mapping, δ is actually much lower than P_a; or equivalently, to maintain $\delta \approx 10^{-6}$, the actual required P_a can be much higher, which dramatically reduces the number of FH minislots (m) to represent each ON/OFF slot in Chorus. Intuitively, this is because with high probability around half of the LAS bits of two nodes will differ.

First, the number of different bits (hamming distance, D) between any two hash outputs follows a binomial distribution $B(l, 0.5)$ where l is the hash length. Second, δ^D is the probability that attacker can make two nodes output success in Chorus when their LASes differ in D bits. Thus, the probability

$$\delta = \sum_{d=1}^{l} \mathrm{Pr}_{B(l,0.5)}[D = d] P_a^d. \tag{4}$$

We then find the maximum P_a: P_a^* s.t. $\delta \leq 10^{-6}$. Suppose $l = 160$, we obtain that $P_a^* \approx 0.83$. Considering the two node (worst) case, Let $P_c = 1 - P'_{nc}/h$ be the probability that one node fails to detect energy in one minislot when the other node is "ON", where $P'_{nc} = 1 - b/h$. And we have $P_a = P_c^m$. Thus, suppose $P'_{nc} = 0.5$ and $h = 4$, then $P_c = 0.875$, $m = 2$ (two FH minislots) should suffice.

5.3 Security of the GAKA Protocols

To show the security of GAKA protocol based on our group MAPs, the modular approach proposed by Bellare et. al. [3] can be applied. Specifically, it has two adversary models - the authenticated link model (AM) and unauthenticated link model (UM). If a protocol is proven to be secure under AM, then it can be shown to be secure in the UM, as long as each message transferred between the parties is authenticated by a protocol called message transfer (MT) authenticator. In our case, the GMS and GML protocols can be regarded as MT authenticators that authenticate all the nodes' messages. The group key agreement protocol in [12] is proven secure under the AM. Thus, our GAKA protocols are secure under the UM.

	Comm. time	Human Effort (bits)	Comp. cost/node	In-band
SPATE [23]	$O(N \cdot T_m)$	$O(N\ell)$	$O(N \cdot \mathsf{hash})$	No
SAS-GMA [20, 21]	$O(N \cdot T_m)$	$O(\ell)$	$O(N \cdot \mathsf{mod\_exp})$	No
I-Code (group) [7]	$N \cdot (2lT_S)$	0	$O(N \cdot \mathsf{hash})$	Yes
TEP (group) [14]	$N \cdot (T_m + T_{sync} + lT_S)$	0	$O(N \cdot \mathsf{hash})$	Yes
GMS (ours)	$2N \cdot T_m + T_{sync} + 2\ell T_S$	0	$O(N \cdot \mathsf{hash})$	Yes
GML (ours)	$\leq N \cdot T_m + T_{sync} + 2lT_S$	0	$O(\mathsf{hash})$	Yes

$l = 160$: hash length; $\ell = 16$; T_m: normal packet's duration; T_S: slot duration; T_{sync}: sync packet duration.

Table 3: Comparison of our protocols with representative existing ones.

6. EVALUATION

In this section, we analyze the complexity of our scheme first. Then we introduce our implementation and experiment results. Here we show the effectiveness of Chorus and the performance of GAKA protocols.

6.1 Complexity Analysis

We analyze and compare the complexity of our group message authentication protocols with previous representative ones based on OOB channels [23, 20, 21], and also with I-code [7] and TEP [14]. The latter two are extended to a group setting, by directly using I-code or TEA to authenticate each $INFO_i$. However, the same setup (device counting and push button on each device) is needed because the group is unknown in advance. To be fair in the security level, we assume that our UFH-based defense against the cancellation attack is also applied to both I-code and TEA (a slot will be extended to multiple minislots in the same way as ours). Note that I-code does not prevent an attacker from hiding the fact that a message was transmitted altogether using collisions or a capture effect [14].

We evaluate the costs only for the corresponding message authentication protocol (without key agreement computation and excluding human delay) in Table. 3. Our protocols are more scalable and efficient compared with them. For example, assume the example parameters we used before ($m = 105$, $\Delta d_0 = 2m$). On a 2.4GHz sensor platform, when $T_S = 5 * 105ms$, $T_m = 5ms$, $T_{sync} = T_S$, $N = 30$, our GMS protocol only needs 18.7s, while I-Code needs 5040s, and TEP-group requires nearly 2552s. The previous protocols [23, 20, 21] involve cumbersome user comparison of short digests, which incurs a large human delay linear with the SAS length.

6.2 Experimental Evaluation

Implementation. We implemented the FH-Chorus and the two GAKA protocols on a TinyOS 2.0 wireless sensor platform, with 24 Crossbow TelosB sensors. We choose node 1 as the controller. We turn off the CSMA to make our Chorus feasible. We set the length of each minislot as 5 ms. Each node will repeatedly check 5 RSSIs at each minislot of its OFF slots. The RSSI threshold is set to -65 dBm. Each of the 24 sensors will report its data to a gateway to help us evaluating the protocol's performance. We place 24 nodes on a table in an indoor environment. The experiments mainly consist of two parts. In the first part we evaluate the effectiveness of Chorus with SAS. In the second part we analyze the overhead of GAKA protocols.

6.2.1 *Chorus Effectiveness Analysis*

We use SAS to illustrate the effectiveness of Chorus. Based on our previous analysis, each slot contains 105 minislots. When the Chorus starts, each node will hop randomly within 4 channels out of all 16, send out random packets during the ON slots, and detect signal during OFF slots. We run Chorus 10 times, and show the results from a typical run.

Decomposition	Initialization	Crypto	Chorus	Total
GMS N=8 Time(s)	10	21	17	48
GMS N=24 Time(s)	30	28	17	75
GML N=8 Time(s)	10	20	8	38
GML N=24 Time(s)	30	24	8	62

Table 4: Time overhead of GAKA protocols

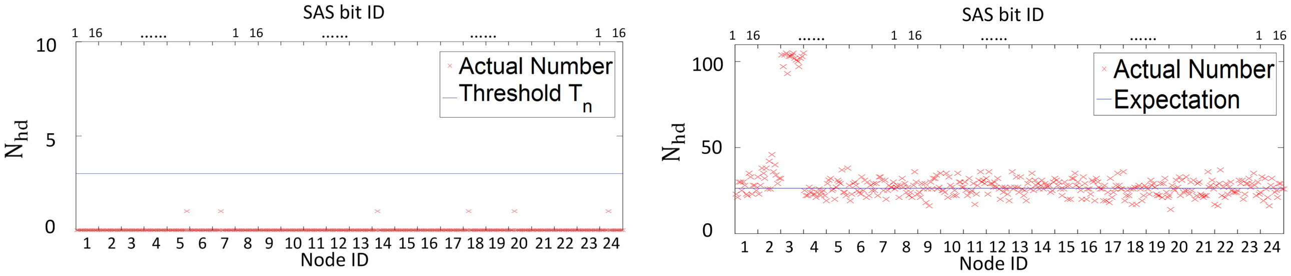
First we verify the correctness of Chorus by setting the same SAS for all nodes before Chorus starts. In Fig. 10(a), we show the number of high-level RSSI minislots (N_{hd}) detected by each node during its OFF slot at each bit position. We can see if using the same SAS, the number of detected high-level RSSI minislots is almost zero for all nodes. Though some nodes will detect high-level RSSI brought by interference (such as Wi-Fi signal), the numbers are all below our interference threshold. In other words, the Chorus will output ACCEPT message at the end.

Next we verify the robustness of Chorus. In the worst case, only 1 bit is different for different SASes. In order to verify the robustness of Chorus even in the worst case, we intentionally pick a random node (node 3) to generate a totally different SAS from other nodes (all bits are reversed) before the Chorus starts. Then we check whether the nodes could detect the differences of all the 16 bits, i.e. whether the nodes are able to detect high-level RSSI ("ON") minislots during all the 16 OFF slots of these bits. We record the number N_{hd} detected within all OFF slots of each node.

This result is shown in 10(b). As we can see, all the nodes can detect multiple "ON" minislots during every OFF slot, which is also above the threshold T_n. Besides, most of the N_{hd} values detected by all nodes are consistent with their expected values through analysis. This consistency proves the correctness of our previous analysis in section 3.3. Meanwhile, the number N_{hd} detected by node 3 is much higher than that of other nodes. This also verifies our analysis in section 3.2: more simultaneous transmitting nodes will increase the possibility of a bit difference being detected, even considering the self-cancellation effect.

The above also indicates that all nodes can be synchronized, and maintain synchronization throughout the comparison phase in Chorus (at least for 32×105=3360 minislots). We note that some nodes' average N_{hd}s are slightly different from the expected value. For example, the average N_{hd} of node 2 (which is closest to node 3) is higher than the expected value, which may be caused by adjacent-channel interference. We observe that in rare cases some nodes experience small N_{hd} values, which can be caused by imperfect synchronization among devices or channel fading. However, this can be solved by increasing the length of each minislot.

Now we consider the signal cancellation attack. The effectiveness of Chorus under this attack can be directly inferred from the above results. Denote the number of high-level RSSI under the presence of relay signal cancellation attack as N_{hd_a}. Using the attacker's cancellation proba-



(a) (b)

Figure 10: Verification of correctness and robustness of GAKA. All nodes are in the range 0.8m × 0.8 m. a) Number of high-level RSSI minislots N_{hd} detected using the same SAS. (b) Number of high-level RSSI minislots detected using SAS with one node's string inverted.

bility in one minislot $b/h < 1$, we can derive the relationship between N_{hd_a} and N_{hd} as $N_{hd_a} = (1 - b/h) \cdot N_{hd}$, and $\mathbb{E}[N_{hd_a}] = (1 - b/h) \cdot \mathbb{E}[N_{hd}]$, which is larger than T_n.

6.2.2 GAKA Efficiency Analysis

In this part we will analyze the time overhead of our GAKA protocols. We run the two GAKA protocols and recorded their running time in Table. 4. We can see that as the number of nodes increases, the time consumed by the cryptographic part (for key computation etc.) increases slowly. Besides, the time overhead of Chorus remains constant. Most of the overhead actually comes from system initialization, which involves one button press on each device (to start the device) and counting at the beginning. The average initialization time is about one second per device. Note that the GML uses fewer number of minislots ($m = 2$), so its Chorus time is lower than GMS.

6.2.3 Discussion

Our security model only considers two types of existing energy cancellation attacks. We believe that these types are complete, as the attacker either generates its own signal or does not. In theory, there could be a stronger correlated energy cancellation attack which requires more advanced hardware. That is, in addition to a pair of directional antennas, the attacker may use a wideband software defined radio (SDR) device which has a bandpass filter, and in realtime it computes and injects a phase delay corresponding to the legitimate signal's frequency. Ideally all channels' signals could get completely cancelled. But this attack can be difficult to carry out in practice as the phase delay must be very precisely generated.

We note that there is another similar type of attack – realtime reactive and selective jamming, where the the attacker can demodulate, interpret, and timely generate an interfering signal to a legitimate transmission that is already "on-the-air" [42]. When the interfering signal is the inverse of the legitimate one at the receiver, the latter can get cancelled. However, the attacker's response time needs to be extremely short. So far, it is reported that a reaction delay of $16\mu s$ can be achieved for sensor signals [42], which is nevertheless too large for complete signal cancellation.

7. CONCLUSION

In this paper we focused on the challenging problem of ad hoc trust initialization for a group of wireless devices without relying on an out-of-band channel. Our main contribution is Chorus, a novel primitive for authenticated equality comparison over the insecure wireless channel in constant time. Chorus achieves non-spoofable string equality comparison, which is based on the infeasibility of energy cancellation and unidirectional error detection codes. Through analysis, we show that Chorus is secure against all known signal cancellation attacks. Chorus can be readily applied to design group message authentication, and group authenticated key agreement protocols, which greatly enhances their scalability and simplifies the protocol structure. Future work will include extending Chorus to be robust under malicious jamming.

Acknowledgments This work is partially supported by NSF grants CNS-1116557 and CNS-1218085.

8. REFERENCES

[1] Top 50 internet of things applications - ranking. `http://www.libelium.com/top_50_iot_sensor_applications_ranking/`.

[2] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: authentication in ad-hoc wireless networks. In *NDSS '02*, 2002.

[3] M. Bellare, R. Canetti, and H. Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols (extended abstract). In *ACM STOC'98*, pages 419–428. ACM, 1998.

[4] M. Bellare and P. Rogaway. Entity authentication and key distribution. In *Advances in Cryptology - CRYPTO'93*, pages 232–249. Springer, 1994.

[5] M. Cagalj, S. Capkun, and J.-P. Hubaux. Key agreement in peer-to-peer wireless networks. *Proceedings of the IEEE*, 94(2):467–478, Feb. 2006.

[6] L. Cai, K. Zeng, H. Chen, and P. Mohapatra. Good neighbor: Ad hoc pairing of nearby wireless devices by multiple antennas. In *NDSS 2011, San Diego, California, USA*. The Internet Society, 2011.

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

[8] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE S & P '03*, page 197, 2003.

[9] C.-H. O. Chen, C.-W. Chen, C. Kuo, Y.-H. Lai, J. M. McCune, A. Studer, A. Perrig, B.-Y. Yang, and T.-C. Wu. Gangs: gather, authenticate 'n group securely. In *MobiCom '08*, pages 92–103, 2008.

[10] D. Dolev and A. Yao. On the security of public key

protocols. *Information Theory, IEEE Transactions on*, 29(2):198 – 208, mar 1983.

[11] W. Du, J. Deng, Y. Han, P. Varshney, J. Katz, and A. Khalili. A pairwise key predistribution scheme for wireless sensor networks. *ACM Transactions on Information and System Security (TISSEC)*, 8(2):228–258, 2005.

[12] R. Dutta and R. Barua. Provably secure constant round contributory group key agreement in dynamic setting. *IEEE Trans. on Inf. Theory*, 54(5):2007–2025, May 2008.

[13] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *CCS '02*, pages 41–47, 2002.

[14] S. Gollakota, N. Ahmed, N. Zeldovich, and D. Katabi. Secure in-band wireless pairing. In *USENIX*, SEC'11, pages 16–16, Berkeley, CA, USA, 2011. USENIX Association.

[15] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. Loud and clear: Human-verifiable authentication based on audio. In *In IEEE ICDCS 2006*, page 10, 2006.

[16] Y. Hou, M. Li, and J. D. Guttman. Chorus: Scalable in-band trust establishment for multiple constrained devices over the insecure wireless channel. In *Technical Report*, Feb. 2013.

[17] S. Jana, S. Premnath, M. Clark, S. Kasera, N. Patwari, and S. Krishnamurthy. On the effectiveness of secret key extraction from wireless signal strength in real environments. In *MobiCom '09*, pages 321–332. ACM, 2009.

[18] C. Kuo, M. Luk, R. Negi, and A. Perrig. Message-in-a-bottle: user-friendly and secure key deployment for sensor nodes. In *SenSys '07*, pages 233–246, 2007.

[19] S. Laur and K. Nyberg. Efficient mutual data authentication using manually authenticated strings. *Cryptology and Network Security*, pages 90–107, 2006.

[20] S. Laur and S. Pasini. SAS-Based Group Authentication and Key Agreement Protocols. In *Public Key Cryptography - PKC '08*, LNCS, pages 197–213, 2008.

[21] S. Laur and S. Pasini. User-aided data authentication. *International Journal of Security and Networks*, 4(1):69–86, 2009.

[22] M. Li, S. Yu, W. Lou, and K. Ren. Group device pairing based secure sensor association and key management for body area networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[23] Y.-H. Lin, A. Studer, H.-C. Hsiao, J. M. McCune, K.-H. Wang, M. Krohn, P.-L. Lin, A. Perrig, H.-M. Sun, and B.-Y. Yang. Spate: small-group pki-less authenticated trust establishment. In *Mobisys '09*, pages 1–14, 2009.

[24] K. Lorincz, D. Malan, T. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton. Sensor networks for emergency response: challenges and opportunities. *IEEE Pervasive Computing*, 3(4):16–23, Oct.-Dec. 2004.

[25] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, and N. Mandayam. Proximate: proximity-based secure pairing using ambient wireless signals. MobiSys '11, pages 211–224, New York, NY, USA, 2011. ACM.

[26] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In *MobiCom'08*, pages 128–139. ACM, 2008.

[27] J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *IEEE S & P*, pages 110–124, 2005.

[28] L. Nguyen and A. Roscoe. Authenticating ad hoc networks by comparison of short digests. *Information and Computation*, 206(2-4):250–271, 2008.

[29] L. Nguyen and A. Roscoe. Authentication protocols based on low-bandwidth unspoofable channels: a comparative survey. *Journal of Computer Security*, 19(1):139–201, 2011.

[30] R. Nithyanand, N. Saxena, G. Tsudik, and E. Uzun. Groupthink: Usability of secure group association for wireless devices. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 331–340. ACM, 2010.

[31] S. Pasini and S. Vaudenay. An optimal non-interactive message authentication protocol. CT-RSA'06, pages 280–294, 2006.

[32] S. Pasini and S. Vaudenay. SAS-based Authenticated Key Agreement. In *Public Key Cryptography - PKC '06*, volume 3958 of *LNCS*, pages 395 – 409, 2006.

[33] T. Perković, M. Čagalj, T. Mastelić, N. Saxena, and D. Begušić. Secure Initialization of Multiple Constrained Wireless Devices for an Unaided User. *IEEE transactions on mobile computing*, 2011.

[34] C. Pöpper, N. O. Tippenhauer, B. Danev, and S. Capkun. Investigation of signal and message manipulations on the wireless channel. ESORICS'11, pages 40–59, 2011.

[35] K. Rasmussen and S. Capkun. Realization of rf distance bounding. In *Proceedings of the USENIX Security Symposium*, 2010.

[36] K. Rasmussen, C. Castelluccia, T. Heydt-Benjamin, and S. Capkun. Proximity-based access control for implantable medical devices. In *ACM CCS*, pages 410–419. ACM, 2009.

[37] L. Shi, M. Li, S. Yu, and J. Yuan. Bana: body area network authentication exploiting channel characteristics. ACM WISEC '12, pages 27–38, 2012.

[38] F. Stajano and R. J. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *IWSP '00*, pages 172–194, 2000.

[39] M. Strasser, S. Capkun, C. Popper, and M. Cagalj. Jamming-resistant key establishment using uncoordinated frequency hopping. In *IEEE S & P*, pages 64–78. IEEE, 2008.

[40] A. S. Tanenbaum. *Computer networks (4. ed.)*. Prentice Hall, 2002.

[41] S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. CRYPTO'05, pages 309–326, 2005.

[42] M. Wilhelm, I. Martinovic, J. Schmitt, and V. Lenders. Reactive jamming in wireless networks: how realistic is the threat. *Proc. of ACM WiSec*, 11:47–52, 2011.