Anti-jamming Broadcast Communication using Uncoordinated Spread Spectrum Techniques

Christina Pöpper, *Student Member, IEEE,* Mario Strasser, *Member, IEEE,* and Srdjan Čapkun, *Member, IEEE*

***Abstract*—Jamming-resistant communication is crucial for safety-critical applications such as emergency alert broadcasts or the dissemination of navigation signals in adversarial settings. In such applications, mission-critical messages are broadcast to a large and unknown number of (potentially untrusted) receivers that rely on the availability, integrity, and authenticity of the messages; here, availability primarily refers to the ability to communicate in the presence of jamming. Common techniques to counter jamming-based denial-of-service attacks such as Frequency Hopping (FH) and Direct Sequence Spread Spectrum (DSSS) cannot be applied in such settings because they depend on secret pairwise or group keys shared between the sender and the receivers before the communication. This dependency entails serious or unsolvable scalability and key-setup problems or weak jamming-resistance (a single malicious receiver can compromise the whole system). As a solution, in this work, we propose *uncoordinated* spread spectrum techniques that enable anti-jamming broadcast communication without shared secrets. Uncoordinated spread spectrum techniques can handle an unlimited amount of (malicious) receivers. We present two instances (Uncoordinated FH and Uncoordinated DSSS) and analyze differences in their performance as well as their combination. We further discuss the applications of these techniques to anti-jamming navigation broadcast, bootstrapping of coordinated spread spectrum communication, and anti-jamming emergency alerts.**

***Index Terms*—Anti-jamming, Broadcast, DSSS, Frequency Hopping, Spread Spectrum Communication, Wireless Security**

I. Introduction

Under the threat of attacker efforts to jam mission- or safety critical wireless transmissions (such as emergency alerts or navigation signals), Spread Spectrum (SS) techniques represent a common way to achieve anti-jamming communication [1]–[3]. Anti-jamming communication is used in commercial and military applications, both between paired devices and from one sender to multiple receiving devices (in multicast or broadcast settings). Spread spectrum techniques use data-independent, random sequences to spread a narrowband information signal over a wide (radio) band of frequencies. Under the premise that it is hard or infeasible for an attacker to jam the entire frequency band, the receiver can correlate the received signal with a replicate of the random sequence to retrieve the original information signal. Important instances of spread spectrum techniques are Frequency Hopping (FH) and Direct-Sequence Spread Spectrum (DSSS). Essential for both FH- and DSSS-based communication is that the sender and the receiver share a secret prior to their communication which enables the receiver to generate the random sequence and to detect and decode the sender's spread signal. This reliance on a pre-shared secret generally precludes unanticipated transmissions between unpaired devices as well as communication from a sender (or base station) to an unknown set of receivers (some of which might be malicious and try to compromise the receptions of other receivers). This problem can best be illustrated as follows: If a base station wants to broadcast a message to a set of receivers in a jamming-resistant manner, it would need to share one or several secret spreading sequences with all the receivers, and the sequences would need to be hidden from the attacker (that could otherwise jam the transmissions using the spreading sequences). In a number of scenarios—such as in those where receivers cannot be trusted or where they are unknown before the actual communication (e.g., in local or global navigation systems)—the assumption about shared secret spreading sequences is unrealistic and will typically prevent the application of anti-jamming communication. We denote this as the *anti-jamming broadcast problem* (further elaborated in Section V-C).

Out-of-band key pre-distribution between the sending and receiving devices generally does not solve the anti-jamming broadcast problem: key pre-distribution is not feasible in the case of unknown receivers (e.g., for navigation) and even if the receivers are known, it suffers from serious scalability issues. An established public-key infrastructure does not solve this problem either because SS techniques require shared secret keys and the devices still need to communicate in order to agree on a shared secret (Diffie-Hellman) key while communicating may be impossible in the presence of a jammer. This leads to an *anti-jamming/key-establishment dependency cycle*.

In this work, we propose *Uncoordinated Spread Spectrum* (USS) techniques that enable anti-jamming communication between sender and receivers that do not share any secret keys. These techniques constitute a solution to the problem of anti-jamming broadcast and anti-jamming key establishment. USS techniques randomize the selection of the spreading key (sequence) such that neither external attackers nor malicious (dishonest) receivers (insiders) are able to jam the communication in a targeted way (the best they can do is to jam using guessed spreading sequences similar to jamming coordinated SS techniques); legitimate USSS receivers only possess public information that cannot be misused for targeted jamming. The jamming resistance of USS communication is comparable to the jamming resistance of their coordinated counterparts. USS techniques achieve this by removing the

Manuscript received April 01, 2009; revised October 15, 2009.

C. Pöpper and S. Čapkun are with the Computer Science Department, ETH Zurich (e-mail: {poepperc,capkuns}@inf.ethz.ch).

M. Strasser is with the Computer Engineering and Networks Laboratory, ETH Zurich (e-mail: strasser@tik.ee.ethz.ch).

Digital Object Identier 10.1109/JSAC.2010.1006xx.

requirement of pre-shared secrets (keys) at the expense of a reduced communication throughput.

Our contributions in this work are as follows: We present three instances of USS techniques, *Uncoordinated Frequency Hopping* (UFH) [4], *Uncoordinated DSSS* (UDSSS) [5], and hybrid UFH-UDSSS. All three techniques enable anti-jamming communication without the requirement of pre-shared secrets. UFH resembles Frequency Hopping but randomizes the selection of the frequency channels while UDSSS randomizes the selection of the spreading codes. For the analysis of USS techniques, we introduce a novel attacker model that enables their joint analysis and comparison. We further implement USS techniques on a software radio platform and present their performance results. Finally, we discuss different applications of USS techniques including emergency alert broadcasts and navigation.

The remainder of this paper is organized as follows: In Section II, we define our considered system and attacker model. We describe Uncoordinated Spread Spectrum Techniques in Section III, including UFH, UDSSS, and hybrid UFH-UDSSS. In Section IV, we provide the results of our prototype implementations as well as performance details. We describe applications of USS techniques in Section V, present related work in Section VI, and conclude in Section VII.

II. System and Attacker Models

Our system consists of a sender A and a set of g receivers $\{B_1, B_2, \ldots, B_g\}$ which are all located within the transmission range of A. For the main part of this work, we focus on the setting where the goal of the sender is to disseminate a single (signed) message M to the receivers in the presence of communication jamming. The setting is easily extendable to scenarios with multiple senders or multiple communication rounds. We assume that each device is computationally capable of efficiently performing (e.g., elliptic-curve-based) public-key operations. In addition, each receiver holds an authentic public key of the sender or of the central authority that can certify the sender's public key. The central authority may be unreachable or off-line at the time of communication.

We further assume that each device is equipped with a radio transceiver that enables reception (and transmission) in a set C of $c = |C|$ orthogonal communication channels. This is achieved using either FH, DSSS, or a combination of both; with FH the communication channels correspond to frequency channels and with DSSS they correspond to spreading codes. We assume the communication channels to be mutually exclusive and define N as the spreading gain of the DSSS communication (number of code chips per data bit). The number of communication channels on which a device is able to transmit and receive in parallel is denoted as c_t and c_r, respectively. In this context, an undisturbed transmission between sender A and a receiver B_i is successful if A and B_i use the same communication channel at the same time.

In order to achieve resistance against unintentional interference and bit errors, A uses an error-encoding scheme with jamming resistance ρ and code rate r_c. Thus, the (signed) message M of length $|M|$ is encoded into a message of length $|M|/r_c$ and the receivers can correctly decode M if not more than $\rho|M|/r_c$ message bits have been disrupted.

Attacker Model

We consider an omnipresent but computationally bounded adversary J who is able to eavesdrop and insert messages arbitrarily, but can only alter or erase messages by adding her own (energy-limited) signals to the wireless medium; that is, she cannot disable the communication channel by blocking the propagation of signals (e.g., by placing a Faraday cage around a node). The goal of the attacker is to prevent all communication between the sender A and all or some of the receivers B_i. In order to achieve this, the attacker is not restricted to message jamming but can also modify existing or insert new messages. More precisely, the attacker can choose among the following actions:

- She can *jam messages* by transmitting (high-power) signals that cause the original signal to become unreadable by the receiver. The fraction of the message that the attacker has to interfere with to successfully jam depends on the used coding scheme (i.e. on ρ).
- She can *modify messages* by either flipping single message bits or by entirely overshadowing original messages. In the former, the attacker superimposes a signal on the radio channel that converts one or several bits in the original message from zero to one or vice versa. In the latter, the attacker's signal is of such high power that it entirely covers the original signal at the receiver. As a result, the original signal is reduced to noise in the attacker's signal and the original message is replaced by the attacker's message. In either case, in this attack the message remains readable by the receiver.
- She can *insert messages* that she generated herself, by using known (cryptographic) functions and keys as well as previously overheard messages. Depending on the signal strength and used spreading codes, the inserted messages might interfere with regular transmissions.

Following previous classification from [1], we further distinguish different types of jammers: static, sweep, random, and reactive jammers. Static, sweep, and random jammers do not sense for ongoing transmissions but jam the channel permanently; they only differ in the regularity of their jamming signals. Reactive jammers initially solely sense for ongoing transmissions and start jamming only after the detection of a message transfer. Repeater jammers [6] are a subclass of reactive jammers against DSSS that intercept the signal, low-noise amplify, filter and re-radiate it (without requiring or getting knowledge of the used spreading codes). Hybrid jammers are a combination of the above types that jam while searching for message transmissions.

In the description of the individual USS schemes we will explain how to prevent or mitigate signal modification and insertion attacks; we will thus focus on jamming attacks in our performance evaluation. We define the strength of the attacker by the number of communication channels c_b that she can block during the transmission of a message. The probability that a message is successfully jammed is then equal to $p_j := \frac{c_b}{c}$. Let $n_j := \frac{t_m}{\rho t_m + t_j}$ and $n_s := \frac{t_m - \rho t_m - t_j}{t_s}$ denote the number of jamming and sensing cycles per message that the attacker can achieve, where t_m is the transmission time of a message, ρ is the minimal fraction of the message that the attacker needs to jam, t_s is her required time to detect a transmission on a specific channel, and t_j is her required time to switch the jamming channels. Given the number of channels c_j (c_s) on which the attacker can jam (sense) simultaneously, we can compute the number of blocked channels c_b for different types of jammers. For the jammer types defined above we obtain: $c_b = c_j$ for static jammers, $c_b = n_j c_j$ for sweep jammers, $c_b = c(1-(1-\frac{c_j}{c})^{n_j})$ for random jammers, $c_b = n_s c_s$ for responsive jammers, $c_b = c_j + n_s c_s$ for responsive-static jammers, $c_b = n_j c_j + n_s c_s$ for responsive-sweep jammers, and finally $c_b = c(1-(1-\frac{c_j}{c})^{n_j}) + n_s c_s$ for responsive-random jammers.

We point out that, for UFH, the attacker's sensing time t_s is determined by the capabilities of her radio transceiver, whereas for UDSSS t_s mainly depends on her computing power. Despreading one bit in UDSSS requires Nq additions and multiplications, where q is the number of samples taken per chip and N is the chip length. The expected time to test whether a certain spreading code was used for the transmission of M is thus $t_s = \frac{1}{2} qnN \frac{|M|}{r_c} / \Lambda_J(N)$, where n accounts for the number of possible spreading sequences and $\Lambda_J(N)$ denotes the number of bits that the attacker can despread per second. Furthermore, the number of channels c_j on which the attacker can jam the communication of a receiver B_i simultaneously is upper bounded by $\lfloor \frac{P_T}{P_j} \rfloor$, where P_T is the maximal power that she is able to achieve at B_i and P_j is her minimal power required for jamming at B_i.

III. Anti-Jamming Broadcast Communication using Uncoordinated Spread Spectrum Techniques

Uncoordinated spread spectrum (USS) techniques enable the jamming-resistant transmission of messages without shared secrets. We first present the basic concepts of USS techniques and then provide details for three instances (UFH, UDSSS, and hybrid UFH-UDSSS).

A. Basic Concept of USS Techniques

USS techniques are closely related to SS techniques in that they spread the information signal over a frequency band that is much larger than the band required for transmitting the information signal. However, unlike SS techniques, in USS the sender does not apply a pre-agreed spreading sequence but uses a public set $C := \{c_1, c_2, \ldots, c_n\}$ of communication channels within the available frequency band; this set is known to all receivers (and to potential attackers). In UFH, the commmunication channels correspond to frequency channels and in UDSSS they correspond to spreading code sequences. The sender chooses the communication channels for the transmission of each message randomly from C and keeps its choice secret. The receivers try to guess the sender's selection in order to receive the message. They overcome their lacking knowledge of the sender's spreading operation by accepting a delay in the reception of the message during which they repeatedly try to guess the sender's spreading sequence.

Due to randomization in the transmission, USS techniques require the sender to transmit the message repeatedly to ensure its successful reception (Figure 1), e.g., for a pre-defined number of times or during a time interval. The message repetitions enable receivers that are not synchronized to the

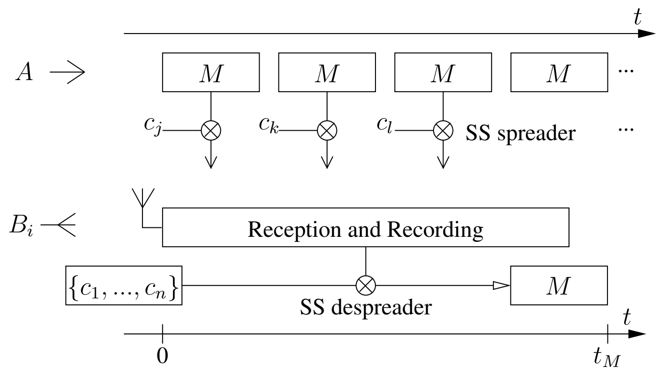


Fig. 1. Principle of Uncoordinated Spread Spectrum (USS) techniques. Sender A repeatedly spreads the (signed) message M using randomly selected spreading sequences from the public set of communication channels $C := \{c_1, c_2, \ldots, c_n\}$ and transmits the resulting signals continuously. Each receiver B_i tries to guess the sender's choice by applying sequences from C to despread the message. After the expected time t_M, B_i will succeed in despreading and verifying M.

beginning of the transmission to receive the message and they empower receivers to get the message even if certain transmissions are jammed. After enough reception attempts (in the case of UFH) or enough decoding attempts (in the case of UDSSS), the receiver will successfully receive the message.

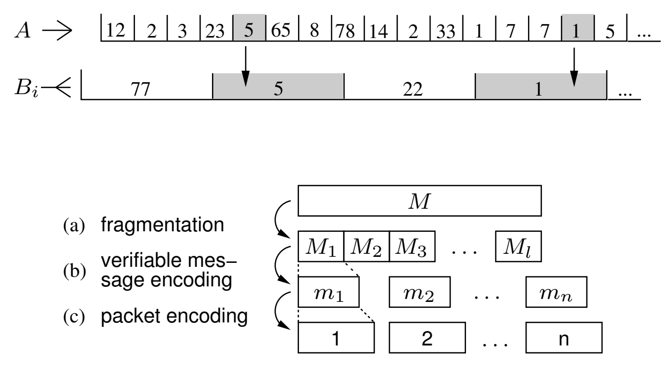
We are now ready to present the details of USS instances.

B. Uncoordinated Frequency Hopping (UFH)

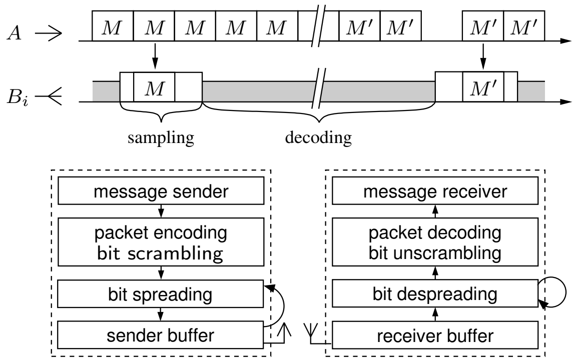
Uncoordinated Frequency Hopping is closely related to (coordinated) frequency hopping (FH) and is based on the assumption that the attacker cannot jam all frequency channels on which the nodes are able to communicate at the same time, so that the sender and receivers can still communicate through the remaining channels. However, as opposed to coordinated frequency hopping, with UFH the sender and receivers do not agree on a secret channel sequence beforehand, but choose the channels on which they send and listen randomly from a predefined set of frequency channels, see Figure 2(a). Communication is possible because, at recurring points in time, the sender and the receivers will be sending and listening on the same frequency channel. Intuitively, given 200 channels and given a sender hopping among the channels at a high rate of, for instance, 2 kHz, a receiver will be listening on the frequency where the sender is transmitting in average 2000/200 = 10 times per second (independent of the receiver's choice of the reception channel). This concept is later formalized into the probality p_m that a transmission of the sender is received by the receiver.

In order for (coordinated or uncoordinated) frequency hopping to resist reactive jamming attacks, the sender can dwell on the same frequency channel only for a short period of time. Each message M is thus split into a set of fragments $M_1, M_2, \ldots, M_l$ with a typical size of a few hundred bits only. After the fragmentation, the sender encapsulates each fragment M_i into a packet m_i, encodes the packets with error correcting codes, and repetitively transmits the encoded packets one after another on randomly chosen frequency channels (Figure 2(a)).

We note that although splitting M into fragments is a straight-forward operation, the reassembly of the received



(a) UFH



(b) UDSSS

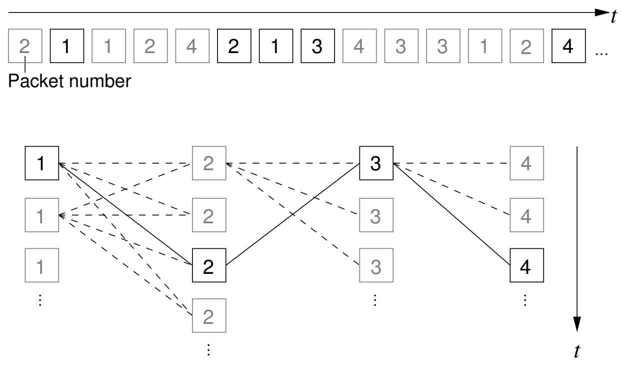
Fig. 2. Instances of USS techniques. (a) UFH: The sender A and each receiver B_i choose the channels on which they send and listen randomly from the set C of available frequency channels, respectively. In this example, two packets get successfully transmitted over channels 5 and 1 (top). Messages are too long to fit on one frequency hop and are thus fragmented, message encoded and packet encoded by the sender; after reception, the messages are reassembled and verified at the receiver (bottom). (b) UDSSS: The sender A repeatedly spreads and transmits the message using randomly selected spreading code sequences from C; the receivers record the channel and subsequently try to identify the used spreading sequence in order to despread the message (top). In contrast to UFH, UDSSS does not require any message fragmentation, but requires processing capabilities for the despreading operation (bottom).

fragments at the receiver is non-trivial if an attacker inserts additional fragments or modifies transmitted ones (that may be hard to distinguish from legitimate fragments); the attacker may easily achieve this for UFH because the receivers do not know the sender's channel selection. By way of example, consider that a legitimate message is divided into l fragments and that z adversarial packets successfully arrive at the receiver, as depicted in Figure 3(a). The number of possible messages that the receiver must reassemble and verify is thus in $O((\frac{z}{l})^l)$. If l is not predefined, adversarial insertions may even lead to an exponential number of message reassemblies and verifications at the receiver (i.e., $O(q^{\lfloor z/q \rfloor})$, where q is the number of unique packets that the attacker inserts per legitimate message fragment). We argue that an attacker can easily insert $z \gg l$ unique packets because the receiver needs l specific packets to reassemble M while the attacker can send any (unique) packets. The expected number z of adversarial packet insertions for different attacker strengths during the legitimate message transmission is displayed in Figure 3(b). In this figure, p_m and p_a denote the probabilities that the sender's and the attacker's packets, respectively, are received by the receiver B_i. We observe that the lower the sender's probability p_m to successfully transmit a packet is, the more packets the attacker will be able to insert for a constant attacker strength. We thus need to provide means for the receiver to efficiently assemble legitimate packets into the sender's message.

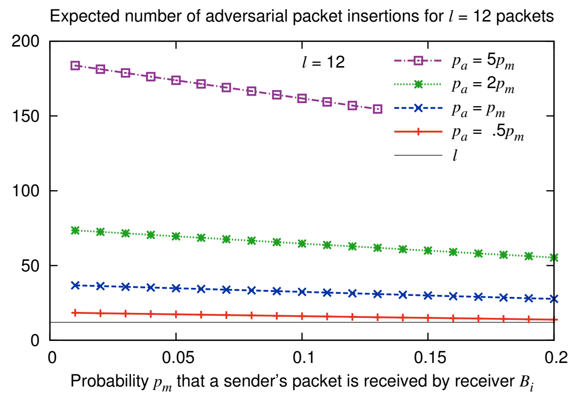
Enabling the receiver to efficiently reassemble the received fragments requires measures that allow the efficient identification of sets of fragments that belong to the same message (without using a shared key). One technique to achieve this is by linking all fragments of the same message to form a hash chain where each fragment is linked to its successor with a hash. Given a collision-resistant hash function, the attacker cannot create branches in the sender's packet chains and is restricted to creating independent fragment chains or to merging partial chains into the sender's chain [4]. A drawback of this scheme is that all l fragments of a message must be received in order to verify the chain and reassemble the message. A more efficient technique is to use *erasure codes* (e.g., Online, LT, or Raptor codes) [7] that allow for splitting a message M into a (possibly infinite) set of L fragments so that any subset of l fragments can be used to reassemble M. We further propose the use of cryptographic *one-way accumulators* based on bilinear maps [8], [9] for the verification of message fragments. Here, the sender computes for each fragment M_i generated by the erasure coding a witness w_i and encapsulates it along with the fragment M_i, the message id, the fragment number i, and the number of required fragments l into the packet $m_i := id||i||l||M_i||w_i$. For each received packet m_i, the receiver computes the accumulator $y := f(w_i, M_i)$. Due to the quasi-commutative property of the one-way accumulator, the accumulator y is identical for all fragment/witness pairs of the same message and thus identifies the message that the fragment belongs to. This scheme is described in more details in [10]. In order for the receiver to accept a phony inserted fragment M_i', the attacker must find a witness w_i' such that $f(w_i, M_i) = f(w_i', M_i')$. Given that $f(\cdot,\cdot)$ is a collision-resistant hash function, finding such a collision is considered infeasible for a computationally bounded attacker. A message M can then be reassembled as soon as a subset of l genuine fragments have been received.

We point out that the purpose of the fragment verification is to ensure the efficiency of the message reassembly process. In particular, it is not intended to provide message authentication or confidentiality (i.e., the attacker may imitate the sender's transmissions by inserting entire self-composed messages or by replaying overheard messages). These security goals can be achieved on the application layer which runs on top of the UFH scheme, e.g., by making use of public-key cryptography, timestamps, and message buffers.

Optimal Number of Frequency Channels and Optimal Channel Selection: In coordinated FH, the jamming resistance increases the more channels the sender and receivers use. However, using all available frequency channels is not optimal for UFH: while the jammer's chances to jam the right channel decrease the more channels are used, so do an uncoordinated receiver's chances to listen on the right channel. Moreover, since using a single channel would be optimal in the absence



(a) UFH packet reception and reassembly at B_i



(b) Malicious packet insertions for UFH

Fig. 3. Effect of packet insertions by the attacker on UFH. (a) UFH message reassembly. Top: Example of the packet arrival at the receiver; black packets were sent by the sender, gray packets were inserted by the attacker (at the time of reception they cannot be distinguished). Bottom: The receiver sorts unique packets according to their fragment number (and message id). Without verifiable message coding, the receiver must reassemble and check all combinations of packets including z attacker packets (indicated by dashed lines, only a subset is shown); the number of such combinations is in $O((\frac{z}{l})^l)$ (in this example $l = 4$). Verifiable message coding enables an efficient reassembly of valid combinations only (solid lines). (b) Malicious packet insertions. The graph shows the expected number of adversarial packet insertions during the legitimate transmission of l packets for different probabilities p_a that the attacker's packets are successfully received. We observe that the smaller p_m is, the more packets the attacker will be able to insert for a constant attacker strength and the more important efficient verifiable message coding mechanisms become.

of jamming, the optimal number of channels also depends on the attacker's strength. As an example, consider a sender and receivers that can send and receive on one frequency channel ($c_t = c_r = 1$) and select the frequency channels uniformly at random from a set of c channels. Given an attacker that blocks c_b channels (Section II), the probability that a packet is successfully received using UFH is $p_m = \frac{1}{c}(1 - \frac{c_b}{c})$, which is maximized for $c = 2c_b$. We show that a similar result holds for the general case where the sender (receiver) sends (receives) on c_t (c_r) frequency channels [10]. In particular, in the presence of an attacker that prevents communication on $c_b \leq c$ channels, the probability p_m that a packet is successfully received with UFH is maximized if the sender and receiver choose the c_t (c_r) frequency channels on which they send (receive) uniformly at random from a set of size c^*, where

$$c^* = \begin{cases} c_b + 1 & \text{if } c_b < c_r \text{ and } c_b < c_t \\ \max\{\approx 2c_b, c_r, c_t\} & \text{otherwise.} \end{cases}$$

C. Uncoordinated DSSS (UDSSS)

UDSSS follows the principle of DSSS in terms of spreading the data using spreading sequences. However, in contrast to anti-jamming DSSS where the spreading sequence is secret and shared exclusively by the communication partners, in UDSSS, a public set C of spreading sequences is used by the sender and the receivers. To transmit a message, the sender repeatedly selects a *fresh*, randomly selected spreading sequence c_i from C and spreads the message with this sequence. The code sequences are used to spread the entire message of length $|M|/r_c$ (each c_i thus contains $N|M|/r_c$ chips). Hence, UDSSS does neither require message fragmentation at the sender nor message reassembly at the receivers.

The receivers record the signal on the channel and despread the message by applying sequences from C, using a trial-and-error approach. More precisely, each receiver samples the radio channel (using an A/D converter with sampling rate R_s and q samples per chip) during the sampling period $t_s = 2t_m$ and stores the samples in a buffer; t_m here denotes the message transmission time. Note that the receivers are not synchronized to the beginning of the sender's message and thus record for (at least) twice the message transmission time, the factor 2 is optimal [5]. While a receiver fills its buffer, it will reject all other arriving signals (Fig. 2(b) top) until the message in the buffer is successfully despread (and its authenticity is verified). After the sampling, the receiver tries to decode the data in the buffer by using code sequences from C and by applying a sliding-window protocol in which the current window is shifted in intervals of $1/R_s$; a complete run of the despreading operation is denoted as one *decoding*. In order to speed-up the decoding operation, the receiver uses a multi-stage operation: based on the threshold integration of despreading $k > 1$ bits, the respective code sequence c_i will be used to despread the entire message, now benefiting from the identified chip synchronization. k is usually larger than the number of bit errors that can be corrected by the used error correcting codes to avoid false rejections. Depending on the available hardware, the despreading operation can partially be performed in parallel or a multi-stage solution can be used [1].

The efficiency of UDSSS is therefore determined by the time that the receivers need to find the right spreading code and its synchronization as well as by the attacker's jamming success. While the processing gain G of UDSSS against noise and unintentional interference corresponds to the processing gain of DSSS ($G = 10\log_{10} N$ dB), the UDSSS advantage over non-responsive jammers (that know C) accounts for nN ($10\log_{10}\log_2(nN)$ if expressed in dB). Figure 4 displays the UDSSS processing gain and advantage over jammers in dB.

UDSSS relies on balanced spreading codes with good auto- and cross-correlation properties in order to achieve precise synchronization at the receivers and for minimal mutual interference of the spreading codes, respectively. Codes for UDSSS that satisfy these properties are shift-register sequences, in particular Gold- and Kasami-codes [11], as well as pseudo-

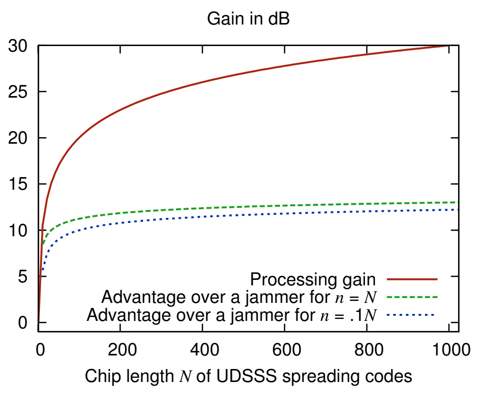


Fig. 4. UDSSS processing gain and advantage over a jammer (being aware of C) for different chip lengths N. The missing synchronization ($\approx N$) and the code set size n both contribute to the advantage over a jammer by the factor nN.

randomly created sequences [12].

Optimal Number of Codes and Optimal Code Selection: Let $p(c_i)$ denote the probability with which code sequence $c_i \in C$ is selected by the sender. Without loss of generality, let further $1 \geq p(c_1) \geq p(c_2) \geq \ldots \geq p(c_n) \geq 0$ and $\sum_i p(c_i) = m$ ($m > 1$ for multiple senders or one sender with multiple transmission antennas).

Theorem 1 (Optimal Code Distribution): Selecting c_i under a uniform distribution from a set of n^* codes (i.e., $p(c_i) = m/n^*$ for $1 \leq i \leq n^*$ and $p(c_i) = 0$ for $n^* < i \leq n$) is optimal with respect to the expected time t_r to receive a message.

Proof Sketch (the full proof is given in [5])*:* The code distribution function $p(\cdot)$ is given with $\sum_{i=1}^{n} p(c_i) = m$. The best strategy for the attacker is to focus her jamming on those codes that are the most likely to be used. Let $\tilde{n}_j = np_j(n_j)$ be the expected number of codes that the attacker can use in parallel to effectively block ongoing transmissions. Hence we get $p_e := \prod_{i=\tilde{n}_j+1}^{n}(1 - p(c_i))$ for the probability that the transmission was jammed. The expected reception time t_r is minimized if p_e is minimized, that is, if $p(c_i) = m/n^*$ for $1 \leq i \leq n^*$ and $p(c_i) = 0$ for $n^* < i \leq n$; the optimal number n^* of codes can be derived (numerically) once $p(c_i)$ and $\tilde{n}_j$ are given. □

Code-based Jamming Attacks: An attacker has three options for performing a code-based jamming attack on UDSSS: i) she can guess the spreading code and try to jam the signal using this code, ii) she can repeat the recorded signal, trying to create a collision with the original transmission, and iii) she can despread (part of) the spread signal and use the identified spreading sequence for jamming the rest of the message during its transmission. As shown in [5], the success probabilities of these attacks are usually low (i.e., $< 1/N$). We point out that the attacker cannot construct a code that would enable her to jam two or more (orthogonal) codes from the code set simultaneously with less than twice the power—hence, the most (energy-efficient) attack is to use the codes from the set.

D. Uncoordinated FH-DSSS

In a similar manner as FH and DSSS can be combined into an FH-DSSS system, UDSSS can be used together with UFH to form a hybrid UFH-UDSSS system. In such a combined scheme, not only the spreading code but also the carrier frequency is chosen randomly from a predefined set for each message transmission. The data signal is first spread by applying the randomly selected spreading code c_k in the DS-spreader. The resulting signal is then fed into a frequency hopper that transmits the signal on a randomly chosen carrier frequency channel c_h. At the receiver, the reverse process takes place. The receiver chooses the carrier frequency $\hat{c}_h$ to listen on randomly and tries to decode the sampled signal by trying all spreading codes. Thus, the transmitted message is only received if the receiver was listening on the right frequency channel (i.e., if $\hat{c}_h = c_h$) and used (guessed) the correct spreading code. The same also applies to the attacker; that is, the attacker has to use the right carrier frequency and either the correct spreading code or enough power in order to jam the transmission.

The advantage of UFH-UDSSS is that it provides covertness (i.e., low probability of intercept) as well as frequency diversity over a large (non-continuous) spectrum (frequency hops can be apart). Given a sufficiently large processing gain of the UDSSS part, the sender's signal is hidden in the background noise. Detecting this signal requires more advanced (and thus typically more time-consuming) techniques than a simple narrow-band signal strength indicator. Hence, the attacker might no longer be able to detect the used carrier frequency before a complete message (or packet) is transmitted and her impact is basically reduced to the one of a non-responsive random or sweep jammer.

E. Discussion

Message Integrity, Authenticity & Confidentiality: Common to all USS techniques is that they require measures that ensure the integrity of the broadcast messages. Specifically, UDSSS (UFH) requires that messages (packets) are identifiable and that bit modifications can be detected; UFH further requires that fragments which belong to the same message can be efficiently identified. Message authentication or confidentiality are in general not part of USS techniques but can be achieved on the application layer which runs on top of the USS schemes, e.g., by making use of public key cryptography, timestamps, and message buffering. We account for the use of such mechanisms by flexible messages sizes (of few thousand bits) which allow for the inclusion of the required material (signatures, timestamps, etc.) in the sender's messages.

Efficiency of USS: Although the message latency under jamming is longer for USS techniques than for the coordinated counterparts, the same message latency as (non-synchronized) spread-spectrum communication can be achieved in the absence of jamming. This requires two parallel sending operations at the USS sender (e.g., using two transmitters). One transmitter broadcasts the message using one channel only ($C_1 = \{c_1\}$ with probability $p(c_1) = 1$) and the second transmitter uses a USS set of communication channels C_2 (where $\forall c_i \in C_2 : p(c_i) = \frac{1}{|C_2|}$). In the absence of jamming, the first communication channel (frequency or spreading code) $c_1 \in C_1$ used by the receiver will succeed. This procedure does, however, not achieve the same robustness against un-

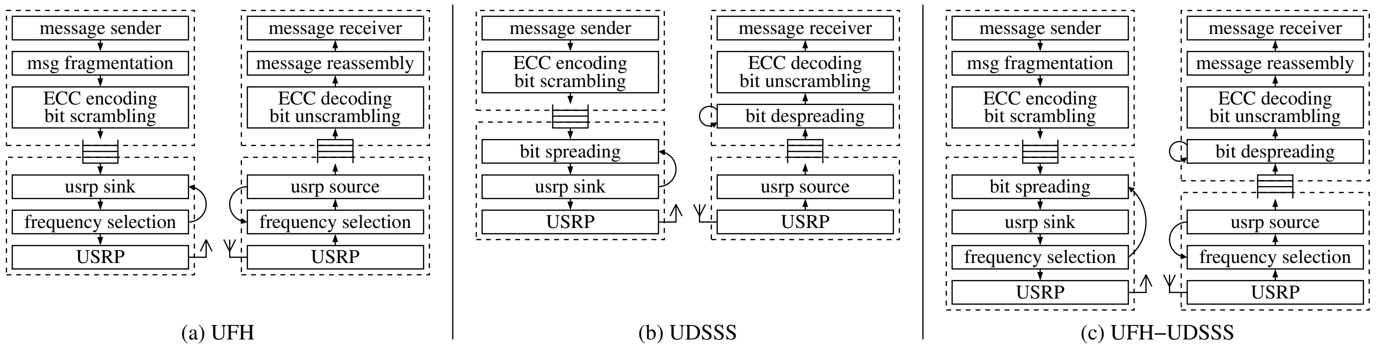


Fig. 5. Schematic descriptions of our UFH (a), UDSSS (b), and UFH-UDSSS (c) sender and receiver implementations.

intentional (internal) interference as coordinated FH or DSSS because only one communication channel is used.

USS for Low-Cost Devices: As our performance analysis in Section IV will reveal, USS techniques entail requirements on the hopping or processing capabilities of the devices. Under a weaker attacker model (which may be realistic, e.g., in sensor networks), both UFH and UDSSS can be modified into more efficient (but also less jamming-resistant) schemes. The basic idea is to use a random key for spreading the data and to *also transmit the selected key* to the receiver(s) using USS techniques. For UFH, this requires *early* key disclosure while UDSSS supports the more secure *delayed* key disclosure.

With early key disclosure UFH, the sender first sends the key that it will later use to generate the hopping sequence of the message transmission, or, alternatively, augments each packet with the key for the next n hops. The advantage of such a scheme is that a receiver that has been able to receive a key (or packet) can deduce the frequency channels of the next hops and no longer has to guess them, but comes at the cost of a higher susceptibility to reactive jamming and insertion attacks.

With delayed-key UDSSS, the sender spreads a message using a random key $K \in \{0,1\}^{|M|}$ and subsequently transmits K (including redundancy) and the message length $|M|$ using UDSSS. The receiver will record $M, |M|, K$ and try to extract K. Once K is known, ordinary (fast) DSSSS decoding can be used to despread M. This procedure is likely to be more efficient since the key (a seed to a PRNG) is in general significantly shorter than the message and thus also less prone to reactive jamming; thus the public key set can be smaller.

Further USS Techniques: Apart from the presented UFH- and UDSSS-based schemes, the principle of Chirp SS may be used to provide anti-jamming communication without pre-shared secrets. Given a fixed chirp duration (up-chirps and down-chirps presenting one bit each), in *Uncoordinated Chirp SS*, the sender randomizes both the selection of the chirp start frequency and the chirp rate. While the random selection of the start frequency resembles UFH communication, the sender achieves higher transmission unpredictability by further applying frequency sweeps within the covered frequency band. In order to prevent the attacker from inserting single bits or replacing message parts, public frequency/chirp-rate sequences need to be used. We leave the details and analysis of such a scheme open for future research.

IV. Prototype Implementation and Performance Evaluation

A. Implementation Details

In order to demonstrate the feasibility of the proposed uncoordinated spread spectrum communication techniques, we created a prototype implementation based on Universal Software Radio Peripherals (USRPs) [13] and GnuRadio [14]. The USRPs include an A/D (D/A) converter that provides an input (output) sampling rate of 64 Mb/s (128 Mb/s) and an input (output) sample resolution of 12 bits (14 bits); the employed RFX2400 daughterboards are able to use a carrier frequency of $2.3 - 2.9$ GHz. In our setup, the USRPs were each connected via a 480 Mb/s USB 2.0 link to a Lenovo T61 ThinkPad (Intel Core 2 Duo CPU @ 2.20 GHz) running Linux (kernel 2.6.27) and GnuRadio (version 3.0.3). For performance reasons and for ease of deployment, our sender and receiver applications were written entirely in C++, which required porting some GnuRadio libraries from Python to C++. Moreover, to achieve synchronization between writing the signal data to the USRP and the actual signal generation that is accurate enough to enable frequency hopping, the USRP drivers had to be adapted. A schematic description of our implementation of UFH, UDSSS, and UFH-UDSSS is given in Figure 5.

UFH Implementation: In our UFH implementation, we use LT erasure codes [7] with optimal degree distributions [15] for the message fragmentation and reassembly; the fragments are mutually linked with an accumulator witness of 144 bit. The sender encapsulates the fragments into packets, encodes them with a (8,4)-Hamming code, and scrambles (interleaves) the bits according to a public pseudo-random permutation. Packets are transmitted on a randomly chosen frequency from the set {2.301 GHz, 2.303 GHz, …, 2.700 GHz} of 200 channels using Gaussian Minimum Shift Keying (GMSK) at a bitrate of $B = 1$ Mb/s. After the last bit of a packet has been transmitted, the USRP driver switches to the new frequency channel and waits until the switching procedure has completed. With the used USRPs, switching a frequency channel took up to $\approx 500\,\mu$s. Given this rather long switching time, the output signal of the transceiver is set to zero during the transition to avoid that a signal is emitted before the actual packet transmission (which would help the attacker in detecting the transmission). The overall packet transmission time is thus $t_m \approx s/B + 500\,\mu$s, where s is the length of an encoded packet. Note that purpose-built FH transceivers can provide much shorter frequency switching times (in the order of one microsecond) and higher data rates (of several tens of Mb/s).

The UFH receivers switch the input channels at a rate of about 100 Hz. If a signal is detected, they continuously try to decode the received data. Successfully received fragments are verified and stored in the corresponding message buffer. Once enough fragments are available, the message is reassembled and the used packets are removed. If the message is further accepted by the application as being authentic (i.e., if the signature is valid), the whole message buffer is discarded and the message gets appended to the message history buffer.

UDSSS Implementation: The sender first encodes the message with a (8,4) Hamming code and scrambles (interleaves) the bits according to a public pseudo-random permutation. Next the sender chooses a spreading code sequence uniform at random, spreads the (encoded and scrambled) message with this code, and sends the resulting chip sequence to the USRP using a differential encoding: the current phase of the baseband signal remains unchanged for a $+1$ and its phase is shifted by $180°$ for a -1. This step (i.e., choosing a code, spreading and sending the message) is repeated until the sender stops the message transmission.

The receiver samples the channel for a duration of $2t_m$, where t_m is the transmission time of a message, decodes the samples into a chip sequence, and stores the sequence into a FIFO buffer. A second thread reads the sequences from the FIFO buffer, decodes all possibly included messages by trying all n code sequences on all possible positions. To decide whether a code-position pair is valid, a two-level test is used: The sender first despreads two randomly selected bits. If the absolute value of the code-bit correlation for at least one of the bits is $\geq N/2$, it decodes (i.e., despreads, unscrambles, and error-corrects) the first 8 bytes of the message. If these 8 bytes are also valid, the whole message is decoded and the included signature is verified. Finally, if the signature is valid, the message is appended to the message history buffer.

UFH-UDSSS Implementation: The hybrid UFH-UDSSS scheme comprises the existing building blocks of the UFH and UDSSS implementation. As with UFH, the carrier frequency is chosen at random for each message transmission (or if the message was split into fragments, for each packet transmission). Next, the transmissions are additionally spread with a randomly selected spreading sequence and the resulting chip sequences are sent to the USRP using the above mentioned differential encoding.

B. Performance Evaluation

We consider a scenario where one sender wants to transmit a (signed) message M to a group of g receivers. In our experiments, we performed a series of message broadcasts using UFH, UDSSS, and UFH-UDSSS from one sender to three receivers. The presented measurement results for more than three receivers were obtained by combining multiple transmission runs for the same parametrization; this procedure is reasonable given the inherent randomness of the reception process. Since we were using frequencies that lie outside of the 2.4 GHz ISM band, the experiments were run in a shielded room. The size of the transmitted messages ranged from 1000 to 4000 bit of which 512 bit were used for the 256-bit ECDSA signature. We chose a spectrum-related processing gain of 23 dB (i.e., 200 channels or chips), an encoded packet size of $s = 720$ bit for UFH and a code set size of 50 code sequences for UDSSS. The evaluated performance metrics are the time t_M^1 after which a particular receiver has received the message and the time t_M^g after which all g receivers have received the message. Due to the lack of a powerfull jammer—the used USRPs can block at most one channel per packet transmission—we simulated the impact of jamming at the receivers by dropping a fraction of p_j of the received packets (or messages), where p_j is the probability that a packet is jammed.

Let $p_m = 1 - \prod_{i=0}^{c_r-1}(1 - \min\{\frac{c_t}{c-i}, 1\}(1 - p_j))$ be the probability that a single packet m is successfully received by the receiver. In our UFH scheme, a receiver can reassemble a message with high probability as soon as he has received a set of $\geq l$ message fragments. The expected number of required reception attempts until a UFH receiver has successfully received a message is thus

$$N(p_m) \approx \sum_{i=0}^{\infty}\sum_{j=0}^{L} D(L,j)\left((1-p_m)^i\right)^{L-j} \cdot \left(1-(1-p_m)^i\right)^j L\,, \qquad (1)$$

where $D(L,j)$ is the number of sets with cardinality j that do not allow the reconstruction of the message. For an ideal erasure code, $D(L,j) = \binom{L}{j}$ if $j < l$ and $D(L,j) = 0$ otherwise; for our implemented LT codes we have $D(L,j) = \binom{L}{j}$ if $j < l$ and $D(L,j) \approx \frac{1}{60}\frac{\sqrt{l}\log^2 l}{j-l+1}$ otherwise [7]. Hence, given the duration t_m of a frequency hop (i.e., the time to switch the frequency and transmit a packet) we get

$$t_M^1 \approx \sum_{i=0}^{\infty}\sum_{j=0}^{L} D(L,j)\left((1-p_m)^i\right)^{L-j} \cdot \left(1-(1-p_m)^i\right)^j L\, t_m \qquad (2)$$

and

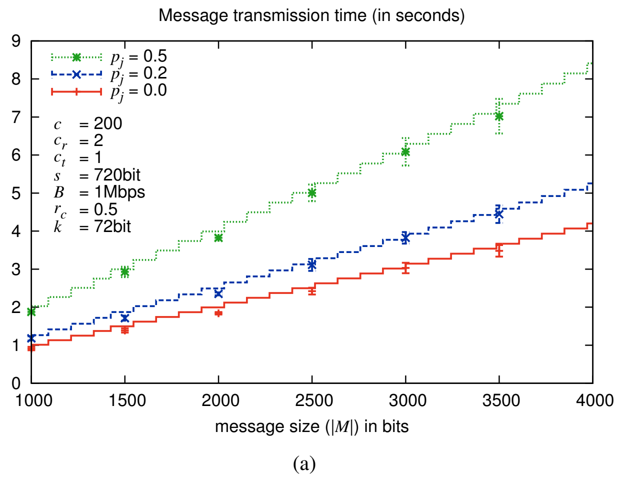
$$t_M^g \approx \sum_{i=0}^{\infty}\left(1-\left(1-\sum_{j=0}^{L} D(L,j)\left((1-p_m)^i\right)^{L-j} \cdot \left(1-(1-p_m)^i\right)^j\right)^g\right) L\, t_m\,. \qquad (3)$$

With UDSSS, the expected time to decode a message of length $|M|$ is $\frac{1}{2}qnNk\frac{|M|}{r_c}/\Lambda_B(N)$, where $\Lambda_B(N)$ is the number of bits that the receiver can despread per second. The expected time to receive a message with UDSSS is thus

$$t_M^1 \approx \frac{Nkq|M|/r_c}{\Lambda_B(N)}\left(\frac{n}{1-p_j} - \frac{n}{2}\right) \qquad (4)$$

for a single receiver and

$$t_M^g \approx \frac{Nkq|M|/r_c}{\Lambda_B(N)}\left(\sum_{i=1}^{\infty}\left(1-(1-p_j^i)^g\right)n + \sum_{i=1}^{n}\left(1-\left(\frac{i}{n}\right)^g\right)\right) \qquad (5)$$



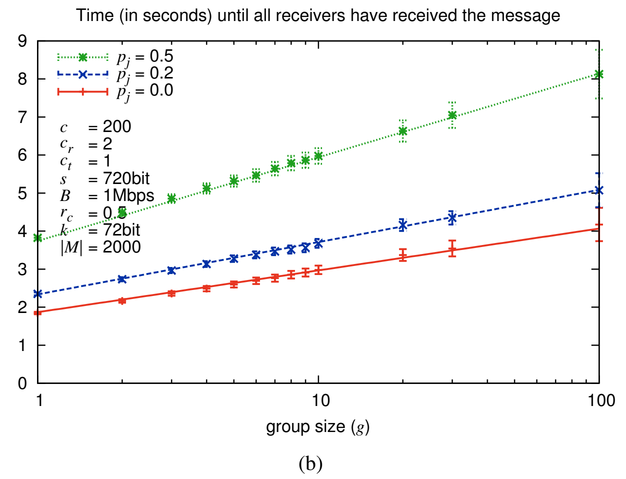
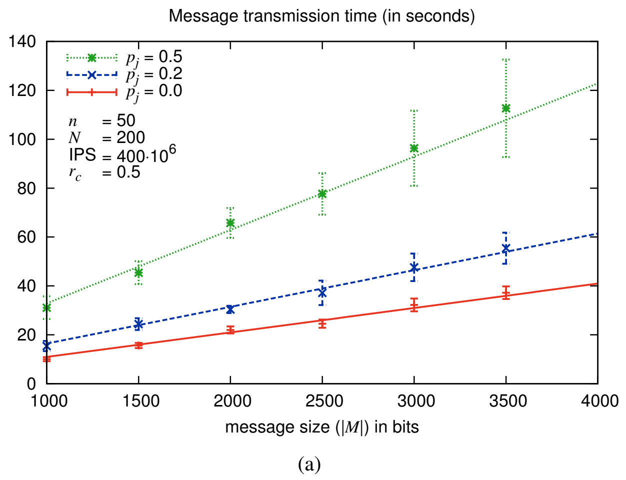


Fig. 6. Message transmission time for UFH as a function of (a) the message size $|M|$ and (b) the number of receivers g. The lines show the expected theoretical results, the points and confidence intervals the findings of our experiments. We observe that the transmission time increases linearly with the message size but only logarithmically with the number of receivers. With our USRP-based prototype implementation, the average time to disseminate a message of 2000 bits to 100 receivers is about 5 s if the processing gain of 23 dB is sufficient to thwart jamming and about 5 s if the attacker can still jam 20% of the packets. We point out that purpose-built FH transceivers or broadband receivers enable to decrease these times significantly.



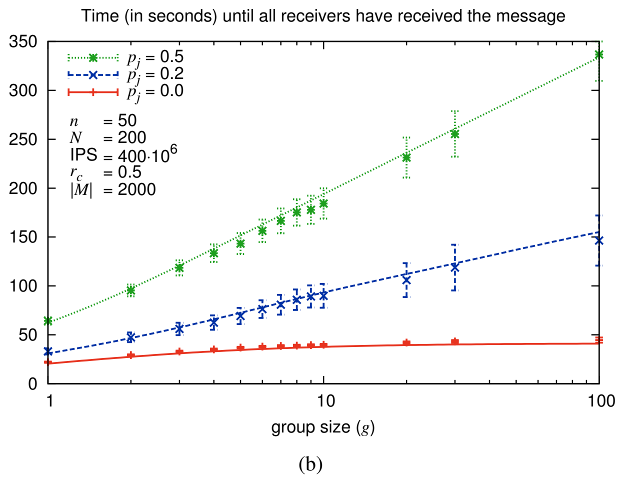


Fig. 7. Message transmission time for UDSSS as a function of (a) the message size $|M|$ and (b) the number of receivers g. The lines show the expected theoretical results, the points and confidence intervals the findings of our experiments. We observe that the transmission time increases linearly with the message size but only logarithmically with the number of receivers. With our USRP-based prototype implementation, the average time to disseminate a message of 2000 bits to 100 receivers is about 40 s if the processing gain of 23 dB is sufficient to thwart jamming and about 150 s if the attacker can still jam 20% of the messages. We point out that purpose-built DSSS transceivers enable to decrease these times significantly.

for a group of g receivers (assuming that $\Lambda_B(N)$ is equal for all g receivers). Finally, the times t_M^1 and t_M^g for the hybrid UFH-UDSSS scheme can be obtained by substituting t_m with $\frac{1}{2}qNk\frac{|m_i|}{r_c}/\Lambda_B(N)$ in Equations (2) and (3), respectively.

Figures 6 and 7 show the message transmission time t_M for our UFH and UDSSS implementations. We observe that t_M increases linearly with the message size but only logarithmically with the number of receivers for both schemes. Disseminating a message to a large number of receivers with UFH or UDSSS therefore scales well with a large or increasing number of receivers. In absolute terms, our UFH implementation performs better than UDSSS; however, this holds only for our system based on USRPs and GNU Radio and this result cannot be generalized for different hardware.

The message transmission time for the hybrid UFH-UDSSS scheme is depicted in Figure 8. In this plot, the processing gain is fixed at 23 dB (i.e., at factor 200) and the x-axis shows how much of this gain is contributed by frequency hopping. That is, an x-value of 50 means that 50 frequency channels and a code length of 150 chips are used. Surprisingly, the results show that the best outcome is achieved if either a pure UFH or UDSSS scheme is used (which one depends on the absolute performance of the two schemes). Note that this evaluation assumes that the attacker's strength is only determined by the achieved processing gain and thus remains constant. However, this might not be a realistic assumption in all cases as UDSSS signals are generally harder to detect. Spreading the UFH signal with a (short) code from a small code set might therefore still be advisable as it forces the attacker to use more advanced (and thus typically more time-consuming) signal detection techniques than a simple narrow-band signal strength indicator.

We point out that the main purpose of this USRP/CPU-based system is to demonstrate the feasibility of the pro-

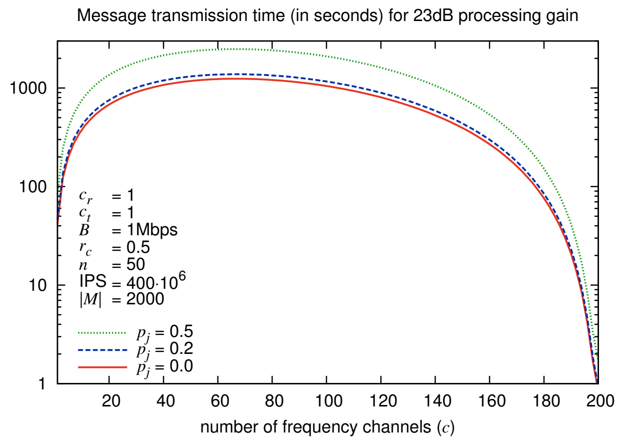


Fig. 8. Message transmission time for UFH-UDSSS. In this plot, the processing gain is fixed at 23 dB (i.e., at 200) and the x-axis shows how much of this gain is contributed by uncoordinated frequency hopping. That is, an x-value of 50 means that 50 frequency channels and a code length of 150 chips are used. The results show that the best outcome is achieved if either a pure UFH or UDSSS scheme is used (with our implementation UFH performs better). Note that this evaluation assumes that the attacker's strength is only determined by the achieved processing gain and thus remains constant.

posed uncoordinated spread spectrum schemes. The achieved transmission times should thus not be considered as final performance benchmarks. Purpose-built frequency hopping transceivers can provide frequency switching times in the order of one microsecond and bit-rates of several tens of Mb/s; the decoding of a bit is typically a single-step operation on hardware-based DSSS receivers. Realistic message transmission times of purpose-built UFH or UDSSS transceivers are thus likely to be 10 to 100 times lower than what we achieve with the presented implementations.

V. Applications of Uncoordinated Spread Spectrum Communication

In this section, we demonstrate the relevance of USS to mission-critical networking and communication by presenting applications that strongly benefit from USS anti-jamming broadcast. As we will describe, USS techniques can be used to either transmit application content or to disseminate system data (e.g., run a Diffie-Hellman protocol) that is crucial for subsequent coordinated anti-jamming communication; in the latter, USS communication is only used for communication bootstrapping. Furthermore, the applications of USS techniques differ regarding short-time use (e.g., emergency broadcast or bootstrapping SS communication) and continuous use (e.g., for localization, navigation, and time-synchronization). The scenarios described in the following share a risk of jamming and of potentially malicious users; in these settings, conventional SS communication would either be infeasible or could easily be disrupted by jammers. We demonstrate that the delays which are introduced by the USS transmissions (Section IV-B) still enable practical and security-relevant applications.

A. *Anti-jamming Navigation Broadcast*

A notedly well-suited application for USS communications is the broadcast of navigation signals which are primarily used for time synchronization, localization, and navigation. Examples of navigation systems include satellite navigation (e.g., GPS [16]) and terrestrial systems such as Loran [17] (based on TDoA) and DME-VOR [18] (based on distance/angle measurements). Localization and time-synchronization systems require the reception of navigation signals from multiple base stations; in general, three or four different signals are necessary for most localization methods [18]. The broadcast stations are precisely time-synchronized (e.g., via wired links) and located at static or predetermined positions. Each broadcast station transmits navigation signals either continuously due to a fixed schedule (GPS, Loran-C) or sends replies to individual localization requests (DME-VOR, WLAN-localization), based on which the localized device determines its position.

Without appropriate protection, navigation signals are vulnerable to signal spoofing, signal synthesis, and jamming attacks [19], [20]. Even though current civilian implementations using GPS satellite signals [16] or terrestrial WLAN signals [21], [22] (based on the 802.11b standard) apply spreading to make the transmissions resistant to *unintentional* interference, they do not provide any means to counteract targeted Denial-of-Service (DoS) attacks because their spreading codes are public and can thus easily be misused for jamming.

USS techniques offer an enhancement to the dissemination of navigation signals that counters targeted jamming. Navigation messages are typically in the order of several hundred bits (e.g., 1500 bit for GPS messages [23]) and will—even comprising authentication credentials—fit into the considered USS message lengths (in our evaluation in Sec. IV-B, messages were up to 4000 bits long). The property of the wireless channel enables the receivers to record samples of several navigation signals in one reception phase, see Figure 9. The receivers each hold the authentic public key of the base stations (although they do not share secret keys with them) and can extract three (or more) individual messages, verify their authenticity, and can therefrom derive position and/or time information. In order to get the precise arrival times required for accurate localization and time-synchronization, UDSSS is the right choice because it easily allows an updated timestamp in each message sent by the base stations (in contrast, UFH timestamps would need to be added to (some of) the packets rather than to the message, which not all UFH packet encoding schemes support). Depending on the implementation and underlying hardware, the delay in the reception of UDSSS messages may vary. However, even if UDSSS causes a delay, the computed position and time are accurate since UDSSS still enables the receiver to record the exact arrival times of the signals it receives.

So far, we have only discussed the implications of UDSSS on navigation signals in terms of anti-jamming. We now further show that UDSSS equally helps to secure navigation against *spoofing* attacks. In [19], Kuhn showed that time-of-arrival-based navigation systems (like GPS) can be secured against signal-synthesis and selective-replay attacks in which the attacker inserts navigation signals as they would be received at the spoofed location. Without protection, an attacker can manipulate the (nanosecond) relative arrival times by pulse-delaying or replaying (individual) navigation signals with a delay. The asymmetric scheme proposed in [19] is made

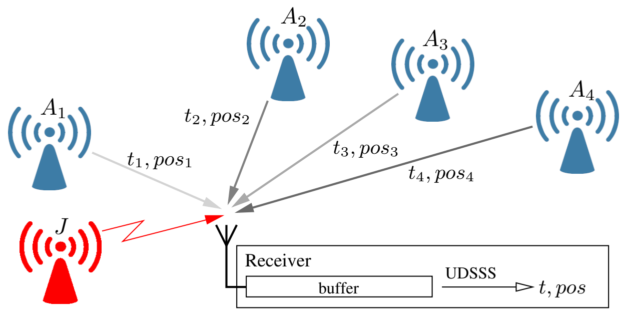


Fig. 9. Application of UDSSS: jamming- (and spoofing-)resistant reception of navigation signals used for positioning (pos) and/or time-synchronization (t). The receiver records the signals of multiple senders which were spread using randomly selected spreading sequences and uses UDSSS decoding to retrieve the sent messages and compute its position and/or local time.

resistant against these kinds of attacks by decoupling the time-critical signal transmission from a delayed disclosure of the applied spreading code; the first signal is spread and hidden below noise level whereas the second signal (the spreading code along with time and position information) is transmitted above the noise level after a delay ρ. A replay attack can now be performed only with a delay $> \rho$. By choosing ρ large enough (e.g., several seconds), even receivers with a low-quality clock can discover the delay in the received timestamps. UDSSS achieves a similar anti-spoofing protection as the scheme in [19]. Due to the steganographic properties of the UDSSS signal, the attacker can only extract and delay individual navigation signals after having successfully identified the used spreading sequences. Due to a comparison of the received timestamp with the local time, the receiver can identify signal delays that exceed a certain accepted threshold; the threshold basically depends on the accuracy of the receiver's clock. This (probabilistic) approach secures against attacks in which the attacker's decoding takes longer than this threshold.

B. Anti-jamming Emergency Alerts

Consider the following examples: (1) A central (governmental) authority needs to inform the public about the threat of an imminent or ongoing (terrorist) attack while minimizing the risk that the attackers can jam the alert transmission. (2) A distress call in high sea operations (nautics) needs to be undertaken in face of an (imminent) adverse invasion.

Even under jamming, information dissemination in these settings is crucial and shared secrets could easily be misused for jamming by malicious receivers. Being able to disseminate the information within a delay (even of seconds to few minutes) under jamming is clearly preferred over not being able to disseminate any information at all. USS communication permits delays as short as their coordinated counterparts in the absence of jamming (see Section III-E) and, once the information has been received by some entities, other communication means (e.g., speech or landline) may additionally support its dissemination to more people or authorities concerned.

Apart from the single-hop broadcast scenarios given above, the anti-jamming emergency alert property of USS communication can also be used for (multi-hop) jammer detection and localization in MANETs. Jamming is a menacing threat

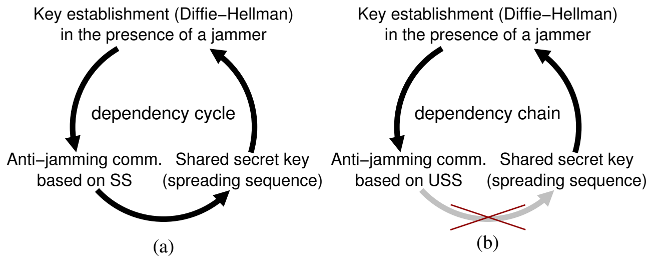

Key establishment (Diffie–Hellman) in the presence of a jammer
dependency cycle
Anti–jamming comm. based on SS
Shared secret key (spreading sequence)
(a)
Key establishment (Diffie–Hellman) in the presence of a jammer
dependency chain
Anti–jamming comm. based on USS
Shared secret key (spreading sequence)
(b)

Fig. 10. Bootstrapping anti-jamming SS communication. (a) The execution of a key-establishment protocol relies on jamming-resistant communication which requires the availability of a shared secret key that shall, in turn, be set up. (b) The circular dependency is broken open by USS techniques that enable (Diffie-Hellman) key establishment communication without shared secrets.

to wireless networks because it deactivates the communication channel and thus, apart from disrupting normal network communication, it also disables the transmission of jamming alerts and communication targeting to counteract the ongoing jamming. Strategies proposed to detect jammers in wireless (sensor) networks [24]–[27] and to avoid the jammed region during future communications are very specific to their (sensor) environments. USS communication in MANETs can be used for the timely delivery of short warning messages outside of the jammed region from where external countermeasures can be taken. In this setting, anti-jamming USS broadcast can be used both to transmit messages despite the ongoing jamming and in order to send jamming alerts that target at localizing and interrupting the jammer.

C. Bootstrapping SS Communications

USS communication can be used for bootstrapping anti-jamming SS communication in mobile ad-hoc settings in which devices encounter each other in an unanticipated manner and need to establish communication under the threat of jamming. In such settings, pairs or groups of devices generally do not share a secret key and would not be able to communicate at all using conventional SS techniques due to a circular dependency between secret keys, key establishment, and secret-based anti-jamming communication (Figure 10(a)).

USS techniques can be used for establishing the required secret key under jamming (Figure 10(b)). The nodes first execute a key establishment protocol using a selected USS technique (UFH or UDSSS, depending on the available hardware) in order to agree on a shared secret key; e.g., by using an authenticated Station-to-Station (STS) Diffie-Hellman key establishment protocol. This requires the exchange of two messages each containing the node identity, key contribution, a freshness indicator, and a digital signature; the authentication of such messages relies either on pre-exchanged authenticated public keys or on a PKI (the central authority does not need to be online at the time of communication). Using elliptic curve cryptography, each message can easily fit within 2500 bits. After the USS-based key-establishment, each node transforms the established key into a hopping sequence or into a spreading code sequence (e.g., using linear feedback shift registers and channel mappers [1]). The subsequent communication among the nodes can now be based on coordinated SS and benefit from higher anti-jamming throughput (compared to uncoordinated SS communication). Since the established key is never used for encrypting or signing sensitive data but only for establishing the channel sequence, a weak choice does not disclose any confidential data.

VI. Related Work

Wireless communication jammers have been widely analyzed and categorized in terms of their capabilities (e.g., broadband or narrowband) and behavior (e.g., constant, random, responsive, sweep) [1], [24], [25]. Many jammer models used in prior works [1], [24]–[26] cover the interference with transmissions in terms of signal jamming as well as dummy packet/preamble insertions. In our work we include more comprehensive, protocol-specific DoS attacks and extend the attacker model to capture signal jamming and overshadowing as well as message modification and insertion.

In [28], [29], the respective authors address broadcast jamming mitigation based on spread-spectrum (SS) communication. Common to these broadcast schemes as well as to other proposed countermeasures against denial-of-service attacks in wireless networks [24], [26], [28]–[30] is that they all rely on secret keys, shared between the sender and receiver(s) prior to their communication. However, pre-establishing keys between devices in ad-hoc networks for subsequent SS communication suffers from scalability and network dynamics problems.

Key-establishment approaches that rely on device proximity [31]–[35] can be used in this context, but require the nodes to be physically close to each other and to use communication channels that are not being jammed (e.g., infrared, wire, or visual). Furthermore, if some of the receivers in multi- or broadcast communications are not trustworthy, relying on pre-shared or established group keys allows malicious receivers to receive messages themselves while withholding (jamming) or modifying them for others [19]. Unlike these approaches, the proposed USS schemes enable (broadcast) communication anti-jamming and key establishment over longer ranges using exclusively radio communication channels.

Recent observations [36], [37] identify the shortcoming of non-existing methods for jamming-resistant communication without shared secrets and propose solutions to this problem [36]–[38]. The solution proposed by Baird et al. [36] uses concurrent codes in combination with UWB pulse transmissions. The achieved jamming resistance is, however, not one-to-one comparable to spread-spectrum-based techniques: While the attacker of SS techniques must have enough transmission power to overcome the processing gain, in [36] the limiting factor is the number of pulses that the attacker can insert, i.e., the energy of the attacker. Jin et al. [38] propose zero pre-shared key DSSS to establish a secret key between a pair of nodes; in contrast to our USS schemes, their solution is targeted for pairwise communication. Dolev et al. present f-AME [37], a round-based, randomized protocol to set up group keys in the presence of message collisions and insertions, but require a (fully connected) group of size $> 3(t+1)^2+2(t+1)$, where t is the number of channels that the attacker can jam (usually t is in the order of tens or hundreds of channels requiring a group of hundreds or even thousands of nodes).

In addition to [37], a substancial number of theoretical and algorithmic results on jamming-resistant networking have been achieved recently, examples include [39]–[42]. The proposals address multiplayer problems under malicious interference, such as anti-jamming MAC protocols [39], gossiping [40], neighbor discovery [41], and leader election and binary consensus [42].

VII. Conclusion

In this work, we addressed the problem of anti-jamming broadcast communication among entities that do not share secret keys. We proposed solutions for the jamming-resistant dissemination of data (e.g., navigation signals or emergency alerts) to a group of (partially) unknown or potentially malicious receivers and for the bootstrapping of conventional anti-jamming communication in ad-hoc settings. Our solutions are based on uncoordinated spread spectrum communication, a novel class of anti-jamming techniques that does not rely on shared secrets. Notably, we presented Uncoordinated Frequency Hopping (UFH), Uncoordinated DSSS (UDSSS), and a hybrid scheme called UFH-UDSSS. The feasibility and practicability of our schemes was demonstrated by means of a USRP/GNU Radio based prototype implementation. Our evaluation results show that even with our prototype, the average time to disseminate a message of 2000 bits to 100 receivers is well below 5 s (40 s) with UFH (UDSSS) (for a processing gain of 23 dB). We argue that these times are reasonable, given that purpose-built hardware as well as multi- and broadband receivers enable to decrease the delays significantly and that with conventional (key-dependent) anti-jamming techniques the devices would not be able to broadcast jamming-resistant messages at all.

Acknowledgment

The work presented in this paper was partially supported by the Swiss National Science Foundation under Grant 200021-116444.

References

[1] R. A. Poisel, *Modern Communications Jamming Principles and Techniques*. Artech House Publishers, 2006.

[2] D. Adamy, *A first course in electronic warfare*. Artech House, 2001.

[3] B. Sklar, *Digital communications: fundamentals and applications*. Prentice-Hall, 2001.

[4] M. Strasser, C. Pöpper, S. Čapkun, and M. Čagalj, "Jamming-resistant key establishment using uncoordinated frequency hopping," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2008.

[5] C. Pöpper, M. Strasser, and S. Čapkun, "Jamming-resistant broadcast communication without shared keys," in *Proceedings of the 18th USENIX Security Symposium*, The USENIX Association, 2009.

[6] W. Hang, W. Zanji, and G. Jingbo, "Performance of DSSS against repeater jamming," in *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pp. 858–861, 2006.

[7] M. Luby, "LT Codes," in *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2002.

[8] L. Nguyen, "Accumulators from Bilinear Pairings and Applications," in *Topics in Cryptology – CT-RSA*, vol. 3376/2005 of *Lecture Notes in Computer Science*, pp. 275–292, Springer Berlin / Heidelberg, 2005.

[9] N. Bari and B. Pfitzmann, "Collision-Free Accumulators and Fail-Stop Signature Schemes Without Trees," in *Advances in Cryptology EUROCRYPT*, vol. 1233/1997 of *Lecture Notes in Computer Science*, pp. 480–494, Springer Berlin / Heidelberg, 1997.

[10] M. Strasser, C. Pöpper, and S. Čapkun, "Efficient uncoordinated FHSS anti-jamming communication," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2009.

[11] B. Natarajan, S. Das, and D. Stevens, “An evolutionary approach to designing complex spreading codes for DS-CDMA,” *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2051–2056, 2005.
[12] D. V. Sarwate and M. B. Pursley, “Crosscorrelation properties of pseudorandom and related sequences,” in *Proceedings of the IEEE*, vol. 68, pp. 593–619, May 1980.
[13] Ettus, “Universal software radio peripheral (USRP).” http://www.ettus.com.
[14] “GNU Radio Software.” http://gnuradio.org/trac.
[15] E. Hyytiä, T. Tirronen, and J. T. Virtamo, “Optimal degree distribution for LT codes with small message length,” in *Proceedings of the IEEE International Conference on Computer Communications (Infocom)*, 2007.
[16] U. Government, “Global positioning system.” http://www.gps.gov, 2009.
[17] I. L. Association, “LORAN: LOng Range Aid to Navigation.” http://www.loran.org.
[18] A. Bensky, *Wireless Positioning Technologies and Applications*. GNSS Technology and Applications Series, Artech House, 2008.
[19] M. Kuhn, “An asymmetric security mechanism for navigation signals,” in *Proceedings of the Information Hiding Workshop*, 2004.
[20] K. B. Rasmussen, S. Čapkun, and M. Čagalj, “SecNav: Secure broadcast localization and time synchronization in wireless networks,” in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 310–313, 2007.
[21] P. Bahl and V. N. Padmanabhan, “RADAR: An in-building RF-based user location and tracking system,” in *Proceedings of the IEEE International Conference on Computer Communications (Infocom)*, 2000.
[22] N. O. Tippenhauer, K. B. Rasmussen, C. Pöpper, and S. Čapkun, “Attacks on public WLAN-based positioning,” in *Proceedings of the ACM/Usenix International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2009.
[23] N. Space and M. S. Center, “Navstar global positioning system: Interface specification IS-GPS-200.” http://www.losangeles.af.mil, March 2006.
[24] W. Xu, W. Trappe, Y. Zhang, and T. Wood, “The feasibility of launching and detecting jamming attacks in wireless networks,” in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2005.
[25] M. Li, I. Koutsopoulos, and R. Poovendran, “Optimal jamming attacks and network defense policies in wireless sensor networks,” in *Proceedings of the IEEE International Conference on Computer Communications (Infocom)*, 2007.
[26] M. Čagalj, S. Čapkun, and J.-P. Hubaux, “Wormhole-based antijamming techniques in sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 1, pp. 100–114, 2007.
[27] A. D. Wood and J. A. Stankovic, “Denial of service in sensor networks,” *IEEE Computer*, vol. 35, pp. 54–62, Oct. 2002.
[28] J. Chiang and Y.-C. Hu, “Dynamic jamming mitigation for wireless broadcast networks,” in *Proceedings of the IEEE International Conference on Computer Communications (Infocom)*, 2008.
[29] Y. Desmedt, R. Safavi-Naini, H. Wang, C. Charnes, and J. Pieprzyk, “Broadcast anti-jamming systems,” in *Proceedings of the IEEE International Conference on Networks (ICON)*, p. 349, 1999.
[30] G. Noubir and G. Lin, “Low-power DoS attacks in data wireless LANs and countermeasures,” *SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 29–30, 2003.
[31] F. Stajano and R. J. Anderson, “The resurrecting duckling: Security issues for ad-hoc wireless networks,” in *Proceedings of the 7th International Workshop on Security Protocols*, Springer-Verlag, 2000.
[32] J. M. McCune, A. Perrig, and M. K. Reiter, “Seeing-is-believing: Using camera phones for human-verifiable authentication,” in *Proceedings of the IEEE Symposium on Security and Privacy*, 2005.
[33] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun, “Loud and clear: Human-verifiable authentication based on audio,” in *Proceedings of the IEEE International Conference on Distributed Computing Systems*, 2006.
[34] S. Čapkun and M. Čagalj, “Integrity regions: authentication through presence in wireless networks,” in *Proceedings of the 5th ACM workshop on Wireless Security (WiSe)*, pp. 1–10, 2006.
[35] C. Gehrmann, C. J. Mitchell, and K. Nyberg, “Manual authentication for wireless devices,” *RSA Cryptobytes*, vol. 7, no. 1, 2004.
[36] L. C. Baird, W. L. Bahn, M. D. Collins, M. C. Carlisle, and S. C. Butler, “Keyless jam resistance,” in *Proceedings of the IEEE Information Assurance and Security Workshop (IAW)*, pp. 143–150, June 2007.
[37] S. Dolev, S. Gilbert, R. Guerraoui, and C. Newport, “Secure communication over radio channels,” in *Proceedings of the 27th ACM symposium on Principles of Distributed Computing (PODC)*, 2008.
[38] T. Jin, G. Noubir, and B. Thapa, “Zero pre-shared secret key establishment in the presence of jammers,” in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 219–228, ACM Press, 2009.
[39] B. Awerbuch, A. Richa, and C. Scheideler, “A jamming-resistant MAC protocol for single-hop wireless networks,” in *Proceedings of the ACM symposium on Principles of Distributed Computing (PODC)*, 2008.
[40] S. Dolev, S. Gilbert, R. Guerraoui, and C. Newport, “Gossiping in a multi-channel radio network (an oblivious approach to coping with malicious interference),” in *Proceedings of the 21st International Symposium on Distributed Computing (DISC)*, pp. 208–222, 2007.
[41] D. Meier, Y. A. Pignolet, S. Schmid, and R. Wattenhofer, “Speed dating despite jammers,” in *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2009.
[42] S. Gilbert, R. Guerraoui, and C. Newport, “Of Malicious Motes and Suspicious Sensors,” *Theoretical Computer Science*, vol. 410, no. 6-7, pp. 546–569, 2009.

Christina Pöpper received the Dipl. Ing. degree in Computer Science from ETH Zurich, Switzerland, in 2005. From 2005 to 2007, she worked at the European Space Agency, Paris, France. She is currently working towards a PhD degree at ETH Zurich. Her research interests are in security and privacy in the context of wireless communications, jamming-resistance, and localization. She is a student member of the IEEE.

Mario Strasser Mario Strasser is a security researcher with the Communication Systems Group at ETH Zurich. He received a Master’s and a PhD degree in Computer Science from ETH Zurich, Switzerland, in 2005 and 2009, respectively. His PhD research focused on the security of wireless networks, with a special interest in jamming and dependability. He is a member of the ACM and IEEE.

Srdjan Čapkun is an Assistant Professor in the Department of Computer Science, ETH Zurich. He received the Dipl. Ing. degree in Electrical Engineering / Computer Science from University of Split, Croatia in 1998, and the Ph.D. degree in Communication Systems from EPFL (Swiss Federal Institute of Technology - Lausanne) in 2004. Prior to joining ETH he was a postdoctoral researcher in the Networked & Embedded Systems Laboratory (NESL), University of California Los Angeles and an Assistant Professor in the Informatics and Mathematical Modeling Department (IMM), Technical University of Denmark (DTU). His research interests include the design and the analysis of security protocols for wireless and wireline networks. He is an associate editor of IEEE Transactions on Mobile Computing and an area editor of ACM MC2R. He was a program chair of the ACM Conference on Wireless Network Security (WiSec), 2008. He is a member of the Zurich Information Security Center (ZISC) and of the RFID Consortium for Security and Privacy (RFID-CUSP).