

# Integrity (*I*) codes: Message Integrity Protection and Authentication Over Insecure Channels

Mario Čagalj<sup>†\*</sup>   Srdjan Čapkun<sup>§\*</sup>   RamKumar Rengaswamy<sup>‡</sup>  
 Ilias Tsigkogiannis<sup>‡</sup>   Mani Srivastava<sup>‡</sup>   Jean-Pierre Hubaux<sup>†</sup>

<sup>†</sup>School of Computer and Communication Sciences  
 Ecole Polytechnique Fédérale de Lausanne (EPFL)  
 CH-1015 Lausanne

<sup>§</sup>Informatics and Mathematical Modelling Department  
 Technical University of Denmark

<sup>‡</sup>Networked & Embedded Systems Laboratory (NESL)  
 Electrical Engineering Department

University of California Los Angeles, Los Angeles, CA, 90095

**Abstract**— We propose *integrity codes (I-codes)*, a coding scheme that enables integrity protection of messages exchanged between entities that do not hold any mutual authentication material (i.e. public keys or shared secret keys). The construction of our codes enables a sender to encode any message such that if its integrity is violated in transmission, the receiver is able to detect it. We analyze in detail the use of *I-codes* on a radio communication channel and we present their implementation on Mica2 wireless sensor platform as a *proof of concept*. We finally show how *I-codes* can be used for several applications, including for key establishment and for broadcast authentication over an insecure radio channel. We perform a detailed analysis of the security of our coding scheme and we show that it is secure within a realistic attacker model.

## I. INTRODUCTION

Conventional security goals like message confidentiality, integrity, and authentication are traditionally achieved through the use of certified public-keys or shared secret keys, and by the application of appropriate cryptographic primitives (i.e., encryption schemes, signatures, message authentication codes, etc.).

In this paper, we propose *I-codes*, a new security primitive that enables integrity protection of the messages exchanged between entities that do not hold any shared secrets or mutual authentication material (i.e. public keys or shared secret keys). The construction of *I-codes* enables a sender to encode any message, such that if its integrity is violated in transmission, the

receiver is able to detect it. Our approach to message integrity protection involves three main components: *on-off keying*, *signal anti-blocking* and *I-coding*. On-off keying is a modulation by which the bit “1” is transmitted on the channel as the *presence* of a signal and the bit “0” is transmitted as the *absence* of a signal. Signal anti-blocking means that the energy of the signal (bit “1”) cannot be annihilated by an adversary (we show several ways how to ensure this). Finally, by *I-coding* we mean that a message is encoded using *I-codes* (described in Section III) before its transmission over an insecure channel.

With these three components, we can ensure that only bits “0” (but not bits “1”) can be flipped by the adversary on the channel and that if a bit is flipped, this will be detected at the receiver, which is guaranteed by the properties of *I-codes* (Section III).

We further show how this approach based on *I-codes* can be implemented on a radio communication channel. To validate our concept, we implement and test *I-codes*, on-off keying and signal anti-blocking components on Mica2 wireless sensor network platform; our implementation demonstrates that the approach based on *I-codes* can be implemented using existing radio and processing hardware and protocols at virtually no extra cost. Ensuring integrity protection over insecure radio channels is particularly important for preventing “man-in-the-middle”-based attacks, which could otherwise be perpetrated on the radio channel. By taking advantage of the characteristics of the radio channel, the *I-codes* help to completely prevent this attack.

\*Equally contributing authors.

Using *I*-codes, we develop a novel concept called *authentication through presence*, which enables message authentication based solely on the awareness of presence in the power range of an entity. We show the application of authentication through presence in two examples: (1) IEEE 802.11 access point authentication, and (2) key establishment over insecure radio channels.

We perform a detailed analysis of the security of *I*-codes on a radio channel and we show that they are secure assuming a realistic attacker model. This analysis takes into account the characteristics of the radio channel such as phase shifts, noise, and the attacker's ability to detect, jam and alter the messages on the channel.

The paper is organized as follows. In Section II, we state our problem and we describe our system and the attacker model. In Section III, we formally introduce *I*-codes and we provide details about their properties. In Section IV, we present the results of the *I*-codes implementation. In Section V, we show how to use *I*-codes for authentication. In Section VI, we present the security analysis of *I*-codes. In Section VII we describe the related work. Finally, we conclude the paper in Section VIII.

## II. PROBLEM STATEMENT AND ASSUMPTIONS

We observe the following problem: *Assuming that two entities (A and B) share a common communication channel, but do not share any secrets or authentication material (e.g., shared keys or authenticated public keys), how can the messages exchanged between these entities be authenticated and how can their integrity be preserved in the presence of an attacker (M)?* Here, by message integrity, we mean that the message must be protected against any malicious modification, and by message authentication we mean that it should be clear who the sender of the message is.

We assume that the two entities involved in the communication (*A* and *B*) do trust each other; otherwise, little can be done. Whenever we speak of the security of a given protocol, we implicitly assume that the entities involved in the protocol are not compromised. We do assume that the entities know the (public) protocol parameters.

We adopt the following attacker model. We assume that the attacker Mallory (*M*) controls the communication channel in a sense that he can eavesdrop messages, and modify transmitted messages by adding his own messages to the channel. We further assume that the attacker cannot disable the communication channel (e.g., use a Faraday's cage to block the propagation of radio signals). The attacker can jam the transmission and in that way prevent the transmission of the information

contained in the message. However, the receiver will still receive the message from the sender, superimposed by the attacker's messages. Finally, we assume *M* to be *computationally bounded*.

It is interesting to observe that the security of *I*-codes themselves does not depend on the attacker being computationally bounded. However, authentication schemes derived from *I*-codes presented in Section V do require the attacker to be computationally bounded.

Our attacker model is similar to the the Dolev-Yao model in that the attacker controls the communication channel, but it differs in that we assume that the attacker cannot fully schedule message transmission as it cannot disable the communication channel. This means that the attacker cannot trivially remove the energy of the signal from the channel (we discuss this in more detail in Section VI).

Before introducing our solution to the above stated problem, we give some examples of attacks on message integrity on the radio channel, which are relevant to our proposal. Fig. 1 shows two types of such attacks. The first type of attack is called *bit flipping*, in which the attacker introduces a signal on the channel that converts bit "0" into "1" or vice-versa. This attack is shown on Fig. 1(a) and Fig. 1(b) for messages modulated using amplitude and frequency modulation, respectively. Here, the bit is flipped such that the attacker adds to the channel the signal of the opposite phase to the one representing the bit and the signal representing the opposite bit. The second type of attack is the *signal overshadowing attack*, shown on Fig. 1(c). In this attack, the attacker adds to the channel a signal representing a bit string different from the one sent by the honest entity with a significantly higher power than the one of the original signal. In this way, the original signal, regardless of its format or modulation, becomes entirely overshadowed by the attacker's signal, and is treated as noise by the receiver.

In the following sections, we show how these and similar attacks on message integrity can be detected through the use of *I*-codes in conjunction with on-off keying and signal anti-blocking components. Even though we make a clear distinction between *I*-codes and on-off keying, that is, signal anti-blocking, we will often abuse the terminology and call the triple (*I*-codes, on-off keying, signal anti-blocking) an *I*-code.

## III. INTEGRITY (*I*)-CODES

In a way similar to a message authentication code (MAC), involving a shared secret key, and a signature scheme, involving certified public keys, an integrity code (*I*-code) provides a method of ensuring the integrity (and a basis for authentication) of a message transmitted

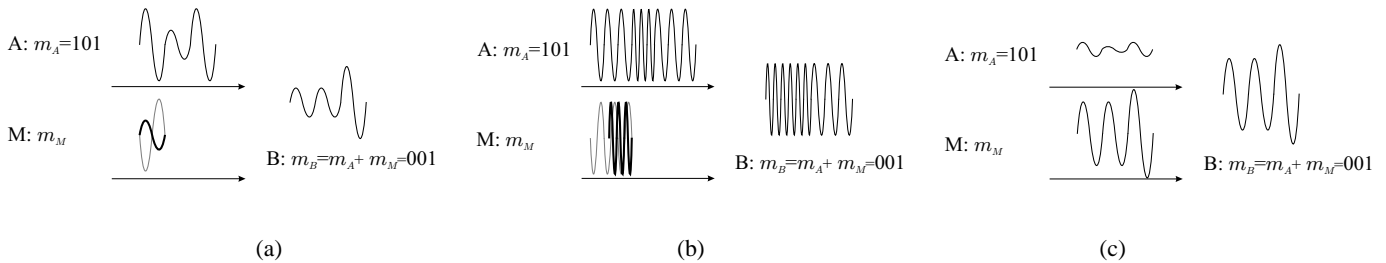


Fig. 1. Example of attacks on message integrity: (a) Bit flipping; signals modulated using amplitude modulation (AM); (b) Bit flipping; signals modulated using frequency modulation (FM); (c) Signal overshadowing; signals modulated using amplitude modulation.

over a public channel. The main difference is that an  $I$ -code removes the assumption that the parties involved in the message exchange share some prior secrets or/and certified public keys.

#### A. Definition

$I$ -codes allow a receiver  $B$  to verify the integrity of the message received from the sender  $A$ , based solely on message coding. We now give a more formal definition of integrity codes and the terminology we will use.

*Definition 1:* An integrity code is a triple  $(\mathcal{S}, \mathcal{C}, e)$ , where the following conditions are satisfied:

1.  $\mathcal{S}$  is a finite set of possible source states (plaintext)
2.  $\mathcal{C}$  is a finite set of binary codewords
3.  $e$  is a source encoding rule  $e : \mathcal{S} \rightarrow \mathcal{C}$ , satisfying the following:
  - o  $e$  is an injective function
  - o it is not possible to convert codeword  $c \in \mathcal{C}$  to another codeword  $c' \in \mathcal{C}$ , such that  $c' \neq c$ , without changing at least one bit “1” of  $c$  to bit “0”.

To make the above definition more concrete, we now give two examples of  $I$ -codes.

*Example 1 (Complementary encoding.):* The encoding rule ( $e$ ) is the following:

$$\begin{aligned} 1 &\longrightarrow 10 \\ 0 &\longrightarrow 01 . \end{aligned}$$

Assume now that we want to encode messages from the set  $\mathcal{S} = \{00, 01, 10, 11\}$  using the above encoding rule. Then,  $\mathcal{C} = \{0101, 0110, 1001, 1010\}$ , i.e.,  $e(00) = 0101$ ,  $e(01) = 0110$ ,  $e(10) = 1001$ , and  $e(11) = 1010$ . This encoding rule is clearly injective. Note further that each codeword  $c \in \mathcal{C}$  is characterized by the equal number of “0”s and “1”s. Therefore, it is not possible to convert one codeword  $c \in \mathcal{C}$  to a different codeword  $c' \in \mathcal{C}$ , without flipping at least one bit “1” to bit “0”. For example, to convert  $c = 0110$  into  $c' = 0101$ , the third bit of  $c$  has

to be changed to 0. By Definition 1, the triple  $(\mathcal{S}, \mathcal{C}, e)$  is an  $I$ -code.

*Example 2 (Codes with fixed Hamming weight.):* We encode each source state  $s \in \mathcal{S}$  into a binary sequence (codeword) of the fixed length ( $\ell$ ) and fixed Hamming weight ( $w$ ). For binary sequences, Hamming weight is the number of bits “1” in the binary sequence. As in the previous example, suppose  $\mathcal{S} = \{00, 01, 10, 11\}$ . Let  $\ell = 4$  and  $w = 3$ . Then the number of possible binary sequences of length  $\ell$  and with Hamming weight  $w$  is  $\binom{\ell}{w} = \binom{4}{3} = 4$ ; i.e.,  $\{0111, 1011, 1101, 1110\}$ . Let us define the set of codewords  $\mathcal{C}$  as follows:  $\mathcal{C} \equiv \{0111, 1011, 1101, 1110\}$ . Suppose further the following source encoding rule:

$$\begin{aligned} 00 &\longrightarrow 0111 \\ 01 &\longrightarrow 1011 \\ 10 &\longrightarrow 1101 \\ 11 &\longrightarrow 1110 , \end{aligned}$$

that is,  $e(00) = 0111$ ,  $e(01) = 1011$ ,  $e(10) = 1101$  and  $e(11) = 1110$ .

Clearly,  $e$  is injective. Moreover, no codeword  $c \in \mathcal{C}$  can be converted into a different codeword  $c' \in \mathcal{C}$ , without flipping at least one bit “1” of  $c$  to bit “0”. Therefore, by Definition 1, the triple  $(\mathcal{S}, \mathcal{C}, e)$  is an  $I$ -code.

In the following section, we show how  $I$ -code can be used on a *radio channel* to ensure the message integrity. However, as we will show,  $I$ -codes are applicable to any communication media (channel) for which we can ensure that it is not possible to block emitted signals on it, except with a negligible probability.

#### B. $I$ -codes on the Radio Channel

Let us consider the simple example shown on Fig. 2. Here,  $m$  denotes the message for which the integrity should be checked. Using the given  $I$ -code (i.e., the complementary encoding rule), the sender first encodes  $m$  into the corresponding  $I$ -code codeword  $c$ . Due to

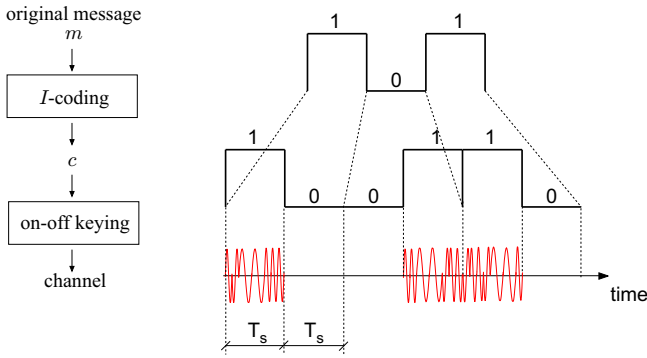


Fig. 2. An example of  $I$ -coding at the sender using the complementary encoding rule:  $1 \rightarrow 10$  and  $0 \rightarrow 01$ .

the injective property of  $I$ -codes (Definition 1), it is possible to recover unambiguously message  $m$  from the codeword  $c$ . In order to transmit  $c$  over a given radio channel, the sender uses the following *on-off keying* modulation at the physical layer. For each symbol “1” of  $c$ , the sender emits some signal (waveform) during the period  $T_s$  (the *symbol period*). For each symbol “0” of  $c$ , however, the sender emits nothing during period  $T_s$  (Fig. 2). The waveforms that are transmitted do not carry any information, but it is the *presence* or *absence* of energy in a given time slot of duration  $T_s$  that conveys information<sup>1</sup>.

In order to retrieve the codeword transmitted, the receiver simply measures the energy in the corresponding time slots of duration  $T_s$ . We will assume for the moment that the sender and the receiver are synchronized at the physical layer and with respect of the beginning and the end of the transmission of  $c$ ; later in the paper, we discuss how this can be achieved. Let  $P_r$  denote the average power that the receiver measures in a given time slot of duration  $T_s$ . Let us also denote with  $P_0$  a predefined *threshold power level*. For the given time slot, the receiver decodes the received signals as follows: (1) if  $P_r \geq P_0$ , output symbol “1”, and (2) if  $P_r < P_0$ , output symbol “0”.

In our example on Fig. 2, the receiver (which is, by assumption, synchronized with the transmitter), listens on the channel during time period  $6 \times T_s$  and for each time slot of duration  $T_s$  it applies the above decoding rule. Finally, the receiver uses the inverse of the used encoding rule (i.e.,  $01 \rightarrow 0$ ,  $10 \rightarrow 1$ ) to retrieve the emitted message  $m = 101$ .

Note that the receiver does not have to know the waveform emitted by the sender. All the receiver has to know is the frequency band used by the sender; the

receiver can be thought of as being a bank of radiometers measuring the energy in the given frequency band.

Assume that we can ensure for the used radio channel that it is not possible to block (annihilate) signals emitted over it, except with a negligible probability. Also, the transmitter should transmit signals using the power level high enough so that the average power as measured by the receiver is above the threshold  $P_0$ .

*Theorem 1:* Assuming that the sender and the receiver are synchronized with respect to the beginning and the end of the transmission of the codeword  $c$ , an adversary cannot trick the receiver into accepting the message  $m'$  when  $m \neq m'$  is sent, except with a negligible probability.

*Proof:* From the injective property of the  $I$ -code (Definition 1) we have

$$m' \neq m \Rightarrow c' \neq c,$$

where  $c'$  is the unique  $I$ -code codeword corresponding to message  $m'$ . Furthermore, converting the codeword  $c$  to another *valid* codeword involves flipping at least one symbol “1” of  $c$  into symbol “0” (Definition 1). Finally, the on-off keying modulation implies that the adversary has to delete (cancel) at least one signal (waveform) emitted on the channel (see Fig. 2).

However, according to our assumption, the adversary can delete the signal emitted on the used radio channel only with a negligible probability. The need for the synchronization between the sender and the receiver is clear. ■

We note that the adversary can still convert symbol “0” to symbol “1”. In this case, however, the receiver will simply drop the received codeword, since such a codeword cannot be demodulated properly. Referring to the example on Fig. 2, assume that the adversary flips the third symbol “0” into symbol “1” in the original codeword  $c = 100110$ . The receiver will decode the altered codeword as  $101110$ . But this codeword cannot be related to any message, since there is no transformation defined for the pair 11. Therefore, flipping symbol “0” to symbol “1” can be thought of as a DoS attack, which the adversary can mount in any case against a radio channel (no matter which modulation scheme is used).

### C. Preventing the attacker from erasing symbol “1”

In order to erase the signal from the channel (symbol “1”), the attacker needs to be able to predict the shape of the signal at the receiver and send the inverted signal to the receiver to cancel it out. There are two major factors that make it difficult for the attacker to erase the signal from the channel: the randomness of the channel and

<sup>1</sup>Note that this is similar to the *pulse position modulation* (PPM).

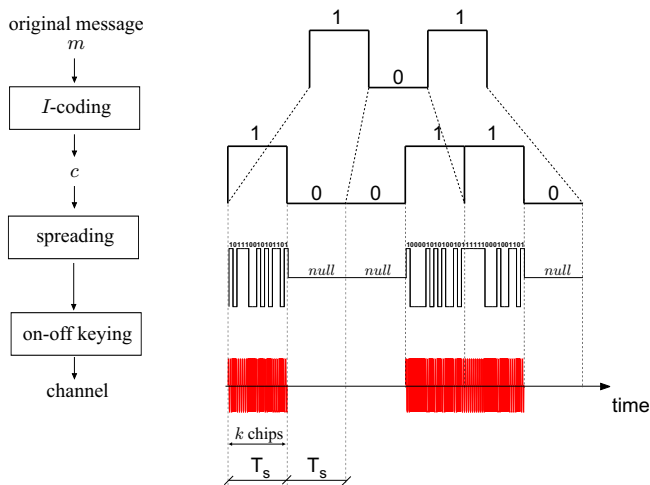


Fig. 3. An example of  $I$ -coding with spreading using FSK modulation.

the randomness of the signal generated at the sender. In Section VI, we analyze in detail the effects of the randomness of the radio channel on the attacker’s ability to erase the signal from the channel. Here, we focus on the randomness of the signals generated at the sender.

To prevent the attacker from erasing the signal, we implement the following scheme: the sender randomizes the signals corresponding to symbols “1”. Namely, to prevent signal erasure, each symbol “1” of the  $I$ -coded message  $c$  is transmitted as a random signal of duration  $T_s$ . Note that we can randomize amplitude, phase, frequency etc. For example, on Fig. 2, we have randomized the frequency. Given the randomness of this signal, it is difficult for the attacker to flip symbol “1” to “0” as it would need to predict the shape of the random signal in order to cancel it.

However, generating arbitrarily random signals using off-the-shelf wireless devices is challenging and, with most devices, not feasible. This is mainly due to the implemented signal modulation schemes which does require the bits to be encoded in a predefined fashion (e.g., in the case of FSK modulation, symbols “1” are transmitted as a sinusoid waveform at one frequency, and symbols “0” is transmitted as the same waveform but at a different frequency). We therefore propose a simple, yet effective solution how to randomize the transmission of symbol “1”, which is compatible with the underlying modulation schemes. For this, we introduce an additional step of encoding called *signal spreading*. This is shown on Fig. 3. An  $I$ -coded message  $c$  is spread such that symbols “1” are converted into random sequences of  $k$  chips each; symbols “0” are converted into *null* symbols. On the channel, chips “1” and “0” are transmitted using the modulation scheme available to the sender (in

our example we use FSK modulation), whereas the *null* symbol is transmitted as the absence of signal.

Here, the ability of the attacker to flip symbol “1” to “0” essentially depends on his ability to guess one of the chip sequences. If the attacker fails to guess the entire sequence, the receiver will still (correctly) decode this signal into symbol “1”. The probability that the attacker guesses the chip sequence of a specific bit is  $2^{-k}$ . For the fixed codeword  $c$ , the attacker’s probability to flip one of the symbols “1” is therefore

$$1 - \left(1 - 2^{-k}\right)^n \approx 1 - e^{-n/2^k},$$

where  $n$  is the number of symbols “1” in  $c$  and the approximation is valid for small  $2^{-k}$ . For example, if  $k = 48$  and  $n = 80$ , this probability is  $2^{-40}$ . Obviously, by increasing  $k$ , this probability can be made arbitrarily small.

In Section VI, we threat this issue of ensuring that the attacker cannot cancel the signal in greater detail.

#### D. Synchronization and Complementary Encoding

Thus far, we have assumed that the sender and the receiver are synchronized with respect to the beginning and the end of the transmission of the given codeword  $c$ . In this section, we show how this can be achieved. Let us start with a simple example.

##### Example 3 (Straightforward synchronization):

Assume that Alice meets Bob and wants to send a message  $m$  to him, using the  $I$ -codes approach. In this scenario, a simple synchronization scheme would consist of using codewords of the fixed length that is publicly known, and letting Alice check if Bob is listening on the correct channel, before she starts transmitting the message. In order to let Bob’s device know as of when it should start demodulating the message transmitted, we can use the convention that every  $I$ -code codeword is prefixed with symbol “1”. When Alice finishes with the transmission, she informs Bob who, in turn, “notifies” his device (e.g., by a push on a button). In this way, Bob informs his device that it may begin to demodulate the received message. The important point is that the Bob’s device should take into account all the symbols it received between the time instant at which the first symbol “1” has arrived and the time instant at which Bob has notified his device (i.e., the push on the button).

As far the synchronization at the physical layer is concerned, by appropriately setting  $T_s$ , we can easily ensure that the transmitter and the receiver remain synchronized throughout the transmission. In Section IV, we report on our experience with a concrete real-life implementation.

Clearly, the approach to the synchronization from the previous example is not very flexible. We next describe a more flexible approach. Let us first introduce some definitions and terminology.

*Definition 2:* For the fixed set of codewords  $\mathcal{C}$ , we define an *incongruous delimiter* (shortly, *i-delimiter*) to be a finite *minimum-length* string of bits that satisfies the following conditions:

1. No *substring* (of subsequent bits) of any codeword  $c \in \mathcal{C}$  can be converted into the *i-delimiter*, without flipping at least one bit “1” of  $c$  to bit “0”.
2. The *i-delimiter* cannot be converted into a *substring* (of subsequent bits) of any  $c \in \mathcal{C}$ , without flipping at least one bit “1” of the *i-delimiter* to bit “0”.

*Example 4:* Consider the set  $\mathcal{C}$  such that  $c = 10100110 \in \mathcal{C}$ . Consider also the following candidate for the *i-delimiter*:  $x = 11011$ . We will show that bit-string  $x$  does not satisfy Definition 2 and therefore is not an *i-delimiter* for the set  $\mathcal{C}$ . This is easily seen by observing that  $10100110 \rightarrow 10110110$ , i.e., it is sufficient to flip only the 4th bit of  $c$  so that  $x$  emerges as a substring of  $c$ . Therefore, the first condition of Definition 2 is not met.

*Theorem 2:* Consider the set of codewords  $\mathcal{C}$  obtained by applying the complementary encoding rule ( $1 \rightarrow 10, 0 \rightarrow 01$ ) to the set of source states (messages)

$\mathcal{S} = \{0, 1, 00, 01, \dots, \overbrace{11 \dots 1}^k\}$ , for arbitrary  $k < \infty$ . The shortest *i-delimiter* for the set  $\mathcal{C}$  has length of 6 bits. Moreover, a string 111000 is an *i-delimiter* for the set  $\mathcal{C}$ .

*Proof:* The first part of the proof can be carried out by mere inspection of all the strings of length smaller than 6 bits, which we omit due to the space considerations.

Consider now the string 111000. It is easily seen that in each codeword  $c \in \mathcal{C}$  the number of subsequent bits 0 and the number of subsequent bits 1 is at most two. Therefore, (i) 111000 cannot be converted into any codeword  $c \in \mathcal{C}$  without flipping at least one of the leading bits “1” in 111000 to bit “0”, and (ii) no substring of any codeword  $c \in \mathcal{C}$  can be converted into 111000, without flipping at least one bit “1” of  $c$  to bit “0”. Since the string 111000 is 6 bits long, we conclude that it satisfies Definition 2, that is, 111000 is an *i-delimiter* for the set  $\mathcal{C}$ . ■

Now we show how to use *i-delimiters* to synchronize the sender and the receiver with respect to the beginning and the end of the transmitted codeword. Let us assume that the sender wants to transmit the following codeword  $c = 1010011001$  (which corresponds to the message  $s = 11010$  under the complementary encoding rule). The

sender simply keeps emitting (using the on-off keying – Fig. 2) the following repetitive sequence:

$$\dots \underbrace{111000}_{i\text{-delimiter}} \overbrace{1010011001}^c \underbrace{111000}_{i\text{-delimiter}} \overbrace{1010011001}^c \underbrace{111000}_{i\text{-delimiter}} \dots$$

The receiver first has to make sure that the peer sender is active (transmitting the above repetitive sequence). Then it decodes a codeword received between any two subsequent *i-delimiters*. If the codeword can be converted back to a message using the inverse of the complementary encoding rule (i.e.,  $(10 \rightarrow 1, 01 \rightarrow 0)$ ), the receiver accepts this message as being authentic. At this stage, the peer sender can stop transmitting the above repeated sequence.

The security of this approach follows directly from Theorem 1 and Theorem 2. Two distinguished properties of this approach based on the combination of *I*-codes and *i-delimiters* are: (i) the receiver does not have to know the length of the codeword being transmitted in advance, and (ii) any *successfully demodulated codeword*<sup>2</sup>, received between two subsequent *i-delimiters*, is authentic.

In the following sections, we report on our experience with the real-life implementation of *I*-codes and we describe the usage of *I*-codes for broadcast authentication and key agreement.

#### IV. IMPLEMENTATION AND RESULTS

We implemented *I*-codes (with spreading) on Mica2 sensor networking platform [1]. This platform consists of a processor and a CC1000 radio. CC1000 is a single-chip RF transceiver, has a programmable frequency (300-1000 MHz) and uses FSK modulation spectrum shaping. It has programmable output power, (-20 to 10 dBm) and a high receiver sensitivity (-110 dBm).

In our *I*-code implementation, we use pairs of sensors running the SOS operating system [12]. Each original message  $m$  is first *I*-coded such that each “1” is transformed into a “10” and “0” into a “01”. An *I*-coded message is then transmitted such that each “1” is transmitted as an SOS packet containing a random payload of length  $k$  (the payload is chosen randomly for each packet) and each “0” is transmitted as an absence of signal of duration  $T_s$  (in our implementation the number of chips per symbol “1” is  $k = 48$  bits and  $T_s = 10$  ms – Fig. 3). Each packet consists of a preamble and of a payload. The preamble is 12 bytes long and with the payload makes a total of 18 bytes per packet.

<sup>2</sup>By “successfully demodulated codeword”, we mean the codeword for which the transformation  $(10 \rightarrow 1, 01 \rightarrow 0)$  exists.

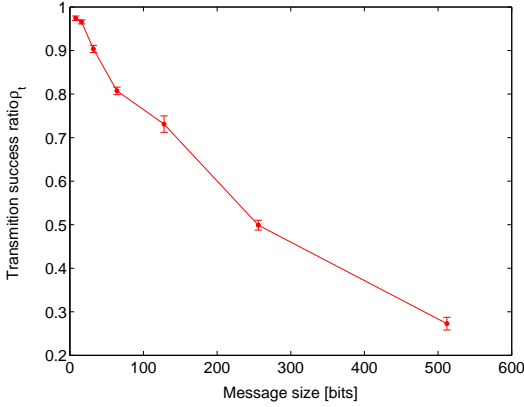


Fig. 4. Robustness of  $I$ -codes. The figure shows the message transmission success ratio  $\rho_t$  as a function of the size of transmitted messages. The results are obtained through measurements on Mica2 sensor motes.

The decoding process at the receiver is implemented as follows. A “silence period” on the channel of the duration of 10 ms is interpreted as a “0”, whereas the presence of a packet is interpreted as “1”. Here, the “silence on the channel” is defined as a period during which the received signal strength on the receiver remains below a preset RSSI level. If the signal level remains above the preset RSSI level, but the received information cannot be interpreted as a packet, the signal is interpreted as “1”.

We experimented with this implementation of  $I$ -codes, by sending 8 to 512 bits long messages (pre-coded message from 16 to 1024 bits). To transmit an  $\ell$ -bit long message using  $I$ -codes, due to the complementary encoding, we actually transmit  $\ell$  “0”s (10 ms of the absence of signal) and  $\ell$  random packets (each 18 bytes long). We measured the message transmission success  $\rho_t$  as a ratio between the number of correctly transmitted messages and the total number of attempts. Here, we consider that a message is correctly transmitted if the message originating from the sender is the same one received by the receiver. The results of our measurements are shown on Fig. 4. From our measurement results we further observed that no messages were altered on the channel such that they appear to the receiver as correct  $I$ -coded messages, but are different from the messages sent by the sender. This is important as it shows that the integrity of the messages transmitted with  $I$ -codes is preserved in our implementation.

These results further show that  $I$ -codes are best suited for reasonably short messages. For longer messages, we would need to be transmitted them multiple times in order for one of the messages to be transmitted correctly. For this purpose, we rely on the  $i$ -delimiters intro-

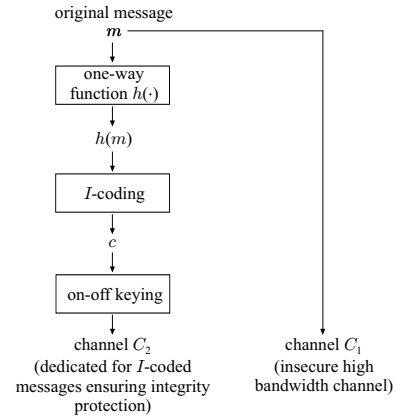


Fig. 5. Typical usage of  $I$ -codes for integrity protection. Original message is transmitted over an insecure high-bandwidth channel  $C_1$ , whereas the integrity protection is enabled with  $I$ -codes on a different channel  $C_2$ .

duced in Section III-D. From these measurements we conclude that  $I$ -codes provide sufficient robustness for the transfer of short messages (e.g., public keys, public parameters, message digests, etc). For example, a 160 bit message (a typical size of the message digest) has a 70% chance of being transmitted correctly, meaning that transmitting it correctly with a 0.999 probability takes approximately 6 successive transmissions; on average it will take  $1/0.7 \approx 2$  retransmissions. These numbers can, however, vary depending on the channel conditions (the level of interference on the channel can be also estimated by the sender and taken into account in estimating the number of transmissions).

With the Mica2 communication speed of 19.2 Kbps, each packet (representing a “1”) is transmitted in 7.5 ms. This means that each bit of the original message gets transmitted in 17.5 ms (single “0” and a single “1”) which means that the communication speed of transmitting the original message with  $I$ -codes is 57 bps. Although  $I$ -codes reduce the speed of communication, this speed is sufficient to enable the integrity-preserving transmission of a message digest (the size of which typically is 160 bits), which then guarantees the integrity-preserving transmission of the entire message.

Furthermore, in some scenarios, only the integrity of a public key needs to be preserved, whereas protecting the rest of the communication can be enabled using the previously transmitted public key.

## V. AUTHENTICATION THROUGH PRESENCE

Using  $I$ -codes, we develop a novel concept called *authentication through presence*, which enables (broadcast) message authentication based solely on the awareness of the presence in the power range of an entity. We first

$A \rightarrow B$  (on  $C_1$ ) :  $m$   
 $A \rightarrow B$  (on  $C_2$ ) :  $\dots \underbrace{111000}_{i\text{-delimiter}} I\text{-code}(h(m)) \underbrace{111000}_{i\text{-delimiter}} \dots$   
 $B$  : Verify the integrity and the authenticity of  $h(m)$  using  $I$ -codes.  
 Verify the integrity and the authenticity of  $m$  using  $h(m)$ .

Fig. 6. A protocol enabling the *authentication through presence* property;  $h(\cdot)$  represents a one-way function.

introduce this concept and then we describe its use in two application scenarios: broadcast authentication and key establishment.

We describe our concept thorough and example involving two parties: the sender  $A$  and the receiver  $B$ . Note that the sender and the receiver do not share *any* authentication material. The main idea of our approach is shown on Fig. 5. The message  $m$ , whose integrity needs to be protected, is sent over a channel  $C_1$  which does not protect its integrity and over which its authenticity cannot be verified. This channel can be realized as *any* communication channel. The message digest  $h(m)$  (e.g., the message hash) is sent over a separate communication channel  $C_2$ , dedicated for integrity protection (we have shown through our implementation in Section IV that this dedicated channel can be realized using existing communication channels). Thus, if  $A$  wants to send a message to  $B$ , she will use the protocol shown on Fig. 6.

In this protocol,  $h(\cdot)$  represents a one-way function used to protect the integrity of the transmitted message. This function can be implemented as a simple hash.  $I\text{-code}(h(m))$  represents the  $I$ -coded message digest  $h(m)$ . The sequences preceding and following after  $I\text{-code}(h(m))$  are  $i$ -delimiters (Section III-D), which ensure that the receiver knows the beginning and the end of the  $I$ -coded message.

In this protocol, the integrity and the authenticity of the message  $m$  is verified through the verification of the authenticity and integrity of its digest  $h(m)$ . The authenticity and the integrity of  $h(m)$  is guaranteed with  $I$ -codes if and only if the following conditions are met: (i) the receiver  $B$  knows that it is in the power range of the sender  $A$ , (ii) the receiver  $B$  knows that  $A$  has started transmitting on the integrity channel ( $C_2$ ). The first condition is *the condition of presence* which ensures that the receiver is receiving signals from the sender. The second condition is *the condition of synchronization* which ensures that the receiver knows at which time is the transmission of data performed. If the receiver

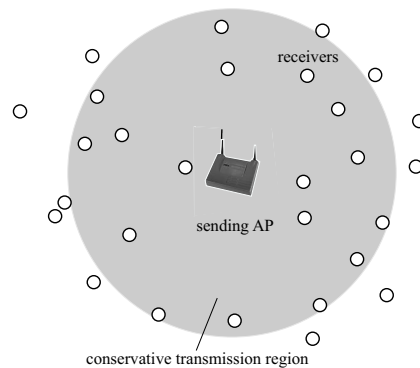


Fig. 7. Broadcast integrity and authentication with an access point. By the “conservative transmission region” we mean the region where the received power of a signal transmitted by the AP exceeds some predefined threshold level (which is a security parameter in our case).

wrongly believes that the transmitter is transmitting, or if it wrongly believes to be in the power range of the sender, a (malicious) entity can insert false data on the channel and these data will be accepted as valid by the receiver. This follows from the properties of  $I$ -codes, which assume the presence of the signal from the legitimate sender on the channel.

In the following two sections, we show in which scenarios the conditions of presence and synchronization are fulfilled and in which, therefore,  $I$ -codes can be used for authentication and integrity protection.

#### A. Access point authentication

Here, we show that authentication through presence can be a useful tool for the broadcast authentication of messages from fixed access points (AP).

Our scenario is depicted on Fig. 7. Here,  $I$ -codes are used by the AP to advertise its public key. This key can be later used to provide authentication and integrity protection of all messages generated by the AP.

This enables any user that comes into the range of the AP to know that the advertised public key of this access point is authentic and belongs to the access point in whose range they lie. If the user trusts the environment in which the access point is placed (a bank or an office), it will trust all information coming from that access point and will use the public key of the AP to establish a secure connection to the station. Here, it is important that the user knows that the environment in which she is placed is covered by at least one legitimate AP. If this condition is fulfilled, it is of little importance if there are any rogue APs present in this space, as long as the legitimate APs are active.

We assume that the sender (AP) is static. The (conservative) reach of its transmission is known to the



receivers. The receivers therefore know before they start receiving the data if they are in the sender's power range or not; this knowledge is a publicly available information. The receivers also know the integrity channel used by the AP to emit its public key. In the case of, for example, IEEE 802.11a, one of the 12 orthogonal channels can be allocated for this purpose.

The AP continuously sends its key on the integrity channel ( $C_2$  on Fig. 5). When it is not advertising its public key, the AP jams the integrity channel to prevent any fake public keys being transmitted over the same channel. As the AP is continuously active, there is no need for synchronization with the receivers; the receivers will start receiving the data when they come into AP's power range. This power range can be estimated by the receiver (a room where the AP is placed), or can even be marked. Furthermore, to avoid attacks during the time when the AP fails, its status (activity) can be signalled to the receivers through some visual channel (e.g. a blinking LED).

### B. Key Establishment Over Insecure Channels

In this section we show how authentication through presence can be used for key establishment over an (insecure) radio link in peer-to-peer networks.

Our key establishment protocol is based on the Diffie-Hellman ( $DH$ ) key agreement protocol, which is known to be vulnerable to the man-in-the-middle attack if the two users involved in the protocol do not share any authenticated information about each other (e.g., public keys, certificates, passwords, shared keys, etc.) prior to the protocol execution. We solve this problem by leveraging on  $I$ -codes that can enable message integrity protection and thus prevent man-in-the-middle attacks.

Before giving details of our protocol, we first motivate the need for carefully developed key agreement protocols in the setting we consider in this paper.

1) *Why are "obvious" solutions not appropriate?:* A simple approach to ensuring the integrity of the  $DH$  public keys, in the setting where two parties  $A$  and  $B$  share no authentication material, consists in first exchanging  $g^{X_A}$  and  $g^{X_B}$  over an insecure channel, and in turn, having say party  $A$  communicate  $I$ -coded message digest  $h(g^{X_A} || g^{X_B})$  to party  $B$ . In turn, party  $B$  simply verifies that  $h(g^{X_A} || g^{X_B})$  matches  $h(g^{X_A} || g^{X_B})$ ; where  $h(\cdot)$  is a hash function satisfying certain security properties and " $||$ " denotes a concatenation.

Now, for this approach based on  $I$ -codes to be efficient, the message digest should be relatively short (see Fig. 4). Therefore, the output of the hash function  $h(\cdot)$  should be truncated to a relatively short length (e.g., around 50 bits).

With this approach, an adversary is successful if he can find values  $a$  and  $b$  such that  $h(g^{X_A} || a) = h(b || g^{X_B})$ ; an adversary can find a *collision* on the truncated output of  $h(\cdot)$ . Note that it is not sufficient for an adversary to find any collision on  $h(\cdot)$ . On the contrary, the adversary is not constrained to finding a *second pre-image*<sup>3</sup> for a single fixed image value  $g^{X_A}$  or  $g^{X_B}$ ; an adversary controls inputs to  $h(\cdot)$  through the values  $a$  and  $b$ . Furthermore, the outcome of the used hash function is truncated (e.g., 50 bits long). Therefore, even if  $h(\cdot)$  is a second pre-image resistant hash function, this still may not be a sufficient guarantee that the adversary cannot find a collision between truncated  $h(g^{X_A} || a)$  and  $h(b || g^{X_B})$ . In Section VII, we will describe a similar problem with an approach proposed by [16], where the users compare the truncated output of a hash function applied to the shared key  $K = g^{X_A X_B}$ .

In the following, section we describe a protocol that enables us to achieve the optimal trade-off between the length of a message to be  $I$ -coded and the security of the protocol.

2) *Diffie-Hellman key agreement protocol with  $I$ -codes:* We base our protocol on a two-party key agreement protocol developed in [2]. This protocol is developed for the settings where the parties (users) share no prior secret or certified public key. It aims at minimizing the users' involvement in the protocol execution; all that the users have to do, in order to achieve a high level of security, is to compare (visually or vocally) a single *short authentication string*. The security of this protocol is proven in a formal model in [2]. Moreover, as it is shown in [2], this protocol allows the users to "arbitrarily" trade-off the security with the length of the authentication string to be compared (i.e., with the users' involvement). For example, with this protocol even if the authentication string is 1 bit long, the advantage of an attacker in a single run of the protocol is 1/2.

Our protocol ( $DH^I$ ) based on  $I$ -codes is shown on Fig. 8. It is essentially the same protocol as the one developed in [2]. The only difference is that instead of having the users compare the short authentication strings  $s_A$  and  $s_B$  via face-to-face voice or visual communication, the authentication string  $s_A$  (or  $s_B$ ) is communicated using  $I$ -codes.

Our protocol unfolds as shown on Fig. 8. Both Alice ( $A$ ) and Bob ( $B$ ) have selected their secret exponents  $X_A$  and  $X_B$ , respectively, randomly from the set  $\{1, 2, \dots, q\}$  ( $q$  being the order of an appropriate

<sup>3</sup>For a given  $x$ ,  $x'$  is said to be a second pre-image if  $x \neq x'$  and  $h(x) = h(x')$  [17].

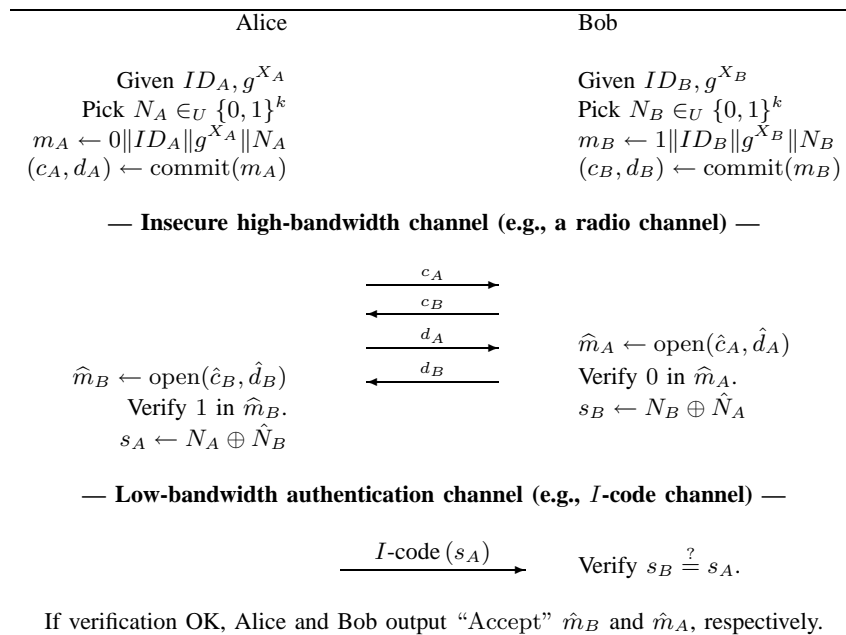


Fig. 8. Diffie-Hellman key agreement protocol based on  $I$ -codes ( $DH^{IC}$ )

multiplicative group) and calculated DH public parameters  $g^{X_A}$  and  $g^{X_B}$ , respectively.  $A$  and  $B$  proceed by generating  $k$ -bit random strings  $N_A$  and  $N_B$ , respectively. Finally,  $A$  and  $B$  calculate commitment/opening pairs for the concatenations  $0 || ID_A || g^{X_A} || N_A$  and  $1 || ID_B || g^{X_B} || N_B$ , respectively. Here, 0 and 1 are two public (and fixed) values that are used to prevent a *reflection attack* [17].  $ID_A$  and  $ID_B$  are human readable identifiers belonging to parties  $A$  and  $B$  (e.g., e-mail addresses).

The following four messages are exchanged over an insecure (radio) link. In the first message,  $A$  sends to  $B$  the commitment  $c_A$ .  $B$  responds with his own commitment  $c_B$ . In turn,  $A$  sends out  $d_A$ , by which  $A$  opens the commitment  $c_A$ .  $B$  checks the correctness of the commitment/opening pair  $(\hat{c}_A, \hat{d}_A)$  and verifies that 0 appears at the beginning of  $\hat{m}_A$ . If the verification is successful,  $B$  sends, in the fourth message,  $d_B$ , by which  $B$  opens the commitment  $c_B$ .  $A$  in turn checks the commitment and verifies that 1 appears at the beginning of  $\hat{m}_B$ . If this verification is successful,  $A$  and  $B$  proceed to the final phase (Fig. 8).

In the final phase,  $A$  and  $B$  first generate the authentication strings  $s_A$  and  $s_B$ , respectively, as shown on Fig. 8 ( $\oplus$  is the bitwise “xor” operation). The length of each of these strings is  $k$ . Finally, Alice sends  $s_A$  over the integrity channel to Bob, which then compares it to  $s_B$ . If they match, Alice and Bob accept the DH public keys  $g^{X_B}$  and  $g^{X_A}$ , and the corresponding identifiers  $ID_B$  and  $ID_A$ , respectively, as being authentic. At this stage,

Alice and Bob can safely generate the corresponding secret DH key ( $g^{X_A X_B}$ ).

A security analysis of the original  $DH$  protocol based on short authentication strings can be found in [2]. In this paper, we only state the result and we extend this analysis to the  $DH^{IC}$  protocol through an analysis of the security characteristics of  $I$ -codes (Section VI). We denote with  $\gamma$  the maximum number of sessions (successful or abortive) of the  $DH^{IC}$  protocol that any party can participate in. Also, we assume that the used commitment scheme is “ideal”, in the sense that the hiding and binding properties of it always hold (i.e., the attacker’s advantage to break the commitment scheme is zero).

*Theorem 3 (cf. [2]):* The probability that an attacker succeeds against a targeted user of the  $DH^{IC}$  protocol is bounded by  $\gamma 2^{-k}$ .

Here, we assume that prior to the protocol execution, the entities know the system parameters and are aware of each others’ presence in the communication range. Therefore, the following condition must be met: the sender has to make sure that the receiver is turned on and is listening on the (correct) channel during the sender’s transmission. This can be easily enforced if two users approach each other to establish a common secret key.

Let us give an example of possible values for the above parameters. Assume that any party can participate in at most  $\gamma = 2^{40}$  sessions (successful or abortive) in its lifetime. Then, by choosing  $k = 55$  we obtain that the highest probability of success by the attacker

(having seen a huge number  $\gamma = 2^{40}$  of protocol runs) is at most  $\gamma 2^{-k} = 2^{-15}$ . Note that  $k$  also represents the length of the verification string  $s_A$  (and  $s_B$ ) to be communicated through  $I$ -codes. From Fig. 4, we can see that with  $I$ -codes, in normal circumstances, it will take on average around  $(1/0.85) < 2$  repetitions of the message of length  $k = 55$  bits, before it is successfully received by the given receiver. This is rather negligible cost, given that all the messages are transmitted over a radio link.

Therefore, with  $I$ -codes, the involvement of the users in the protocol execution is rather minimal.

## VI. SECURITY ANALYSIS OF $I$ -CODES

In this section, we discuss security of  $I$ -codes from the signal cancellation point of view. As we already mentioned in Section III-C, the security of  $I$ -codes depends on the inability of the attacker to flip symbols “1” into “0”, by which she breaks the integrity of the exchanged messages. By a successful attack on  $I$ -codes, we consider that the attacker is able to break the integrity of the transmitted message, meaning that the receiver accepts a message as valid even if it has been modified by the attacker on the channel. Note that we reason about the security of  $I$ -codes within the system and the attacker model described in Section II.

We focus on the security of  $I$ -codes used over the radio communications channel. In order to delete (cancel) a signal  $s(t)$  emitted on a radio channel, the only hope for the adversary is to have its signal  $s'(t)$  arrive at the receiver with the same amplitude as  $s(t)$  but opposite in phase, that is,  $s'(t) = -s(t)$ . There are two main factors that make it hard for the attacker to cancel the signal at the receiver: (1) the unpredictability of the channel conditions (2) the unpredictability of the signal generated by the sender. In order to cancel the signal at the receiver, the attacker needs to estimate the channel conditions (to know how the channel will shape the original signal), and predict the shape of the signal generated at the sender (to know which form to generate to cancel the signal). Channel conditions are highly influenced by the environment and in high-frequency communication systems (e.g., 2.4 GHz), it is nearly impossible for the attacker to predict them due to the un-predictable amplitudes and phases, the multipath fading effects, etc.

In this section, we analyze how channel and signal unpredictability affect the attacker’s ability to cancel-out the signal on the channel. We show that the odds of the adversary to cancel the signal  $s(t)$  are indeed negligible.

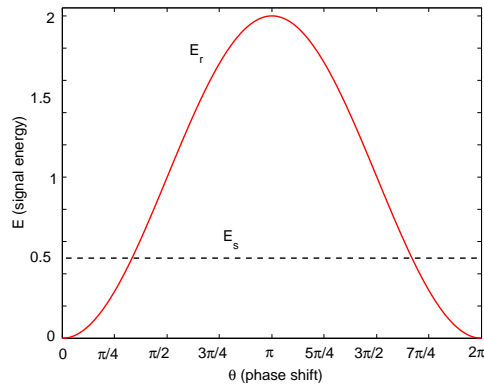


Fig. 9. The energy of the signal  $r(t) \equiv \cos(\omega_0 t) - \cos(\omega_0 t - \theta)$  and the signal  $s(t) = \cos(\omega_0 t)$  normalized with respect to  $T_s$  (the average power).

### A. Anti-Blocking Property of the Radio Channel

We first start by showing how channel conditions affect the attacker’s ability to cancel the radio signal.

Let us assume that the sender emits cosine signal  $s(t)$  with unit amplitude and frequency  $f_0$ , i.e.,  $s(t) = \cos(\omega_0 t)$ , where  $\omega_0 = 2\pi f_0$ . We assume that the adversary knows somehow the exact value of the amplitude of the signal received at the receiver. Furthermore, we assume that there are no multipath fading effects and that the adversary knows  $s(t)$ . Note that with these assumptions, we only make the task of the adversary a lot easier. In reality, multipath effects and interferences from other transmitters can easily make the channel sufficiently random to forbid the attacker to even estimate the state of the signal at the receiver  $r(t)$ .

Let us define  $r(t) \equiv \cos(\omega_0 t) - \cos(\omega_0 t - \theta)$ , where  $\theta \in [0, 2\pi)$ . Here,  $r(t)$  can be thought of as the signal obtained as the superposition of the adversary’s annihilating signal  $s'(t) = -\cos(\omega_0 t - \theta)$  and  $s(t)$ ;  $\theta$  accounts for the potential *phase shift*. The energy  $E_r$  of the signal  $r(t)$ , with duration  $T_s$ , can be calculated as follows [20]:

$$\begin{aligned} E_r &= \int_0^{T_s} r^2(t) dt \\ &= \frac{1}{\omega_0} \sin^2\left(\frac{\theta}{2}\right) (2\omega_0 T_s - \sin(\theta) + \sin(\theta - 2\omega_0 T_s)) \\ &\stackrel{(1)}{\approx} 2T_s \sin^2\left(\frac{\theta}{2}\right), \end{aligned} \tag{1}$$

where the approximation (1) is valid for high frequencies  $f_0$  (e.g.,  $f_0 = 2.4$  GHz), since  $-1 \leq \sin(\cdot) \leq 1$  implies  $\sin(\cdot)/\omega_0 = \sin(\cdot)/(2\pi f_0) \rightarrow 0$ .

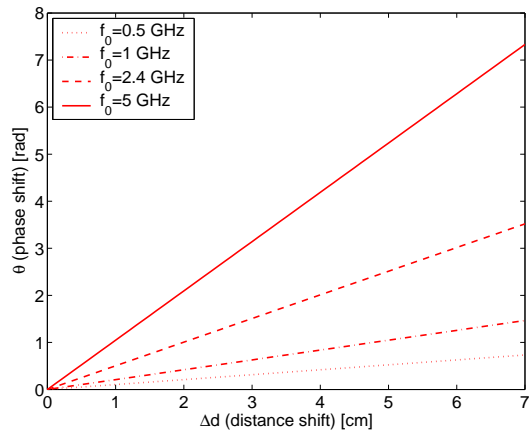


Fig. 10. The phase shift  $\theta$  as a function of the distance shift  $\Delta d$  for different frequencies  $f_0$ .

We plot the expression (1) on Fig. 9; note that we normalize the energy with respect to  $T_s$  (therefore obtaining the average power of the signal). On the same figure, we also plot the energy of the unobstructed signal  $s(t) = \cos(\omega_0 t)$ , i.e.,  $E_s = \int_0^{T_s} \cos^2(\omega_0 t) dt = T_s/2$ . A striking result on this figure is that for most values of  $\theta$  the adversary actually contributes to the energy of the original signal  $s(t)$ . In order to at least attenuate  $s(t)$ , the adversary has to ensure that  $\theta \in (-\theta_0, \theta_0)$ , where  $\theta_0$  is calculated as follows:

$$\frac{E_r}{E_s} = 4 \sin^2 \left( \frac{\theta}{2} \right) < 1 \Rightarrow \sin \left( \frac{\theta}{2} \right) < \pm \frac{1}{2}, \quad (2)$$

and therefore,  $\theta_0 = 2 \arcsin \left( \frac{1}{2} \right) = \frac{\pi}{3}$ . Therefore, the attacker attenuates<sup>4</sup>  $s(t)$  for  $\theta \in [0, \frac{\pi}{3}] \cup (\frac{5\pi}{3}, 2\pi]$  (see Fig. 9); note that this interval represents  $1/3$  ( $\approx 33\%$ ) of all the possible phase shifts.

We now show how demanding it is for the attacker to keep the phase shift  $\theta$  within the given bounds. We know that  $\theta = \omega_0 \Delta t$ , for a *time shift (delay)*  $\Delta t$ . In time  $\Delta t$ , the electromagnetic wave can travel the distance  $\Delta d = \Delta t \cdot c$ , where  $c$  is the propagation speed of the wave. We call  $\Delta d$  the *distance shift*. Combining these expressions we have:

$$\theta = \frac{2\pi f_0}{c} \Delta d. \quad (3)$$

On Fig. 10 we plot expression (3) for different frequencies  $f_0$ . We can see that the higher the frequency of the signal is, the higher the impact of the fixed distance shift  $\Delta d$  on the phase shift  $\theta$  is. More importantly, for  $f_0 = 5$  GHz (IEEE 802.11a), a  $\Delta d$  as small as 1 cm results in phase shift of  $\frac{\pi}{3}$ . As we discussed above, the adversary has to ensure that  $\theta \in [0, \frac{\pi}{3}] \cup (\frac{5\pi}{3}, 2\pi]$ , in order to at least attenuate the signal  $s(t)$ . A more

reasonable goal for the adversary would be to reduce the energy of the signal  $s(t)$  for say 50%, which requires, for  $f_0 = 5$  GHz,  $\theta \in [0, 0.7227] \cup (5.5605, 2\pi]$ . This phase shift corresponds to  $\Delta d \approx 7$  mm. Therefore, for high frequencies, the adversary has to estimate the distances between himself and both the sender and the receiver with a very high accuracy. Otherwise, he cannot hope to have the phase shift fall within the desired interval.

If the distance between the sender and the receiver continuously changes (in a fashion unpredictable to the attacker), the uncertainty of the adversary is further increased (note that this can be a very limited motion, in the order of  $\Delta d$ ). Therefore, in a sense, mobility helps security. Another source of the uncertainty for the adversary is the time delay  $\Delta t = \Delta d/c$ . For example, a distance shift  $\Delta d = 7$  mm is equivalent to a delay of  $\Delta t \approx 23$  ps. Therefore, the adversary has to operate with an extremely high time accuracy, otherwise he cannot keep  $\theta$  within the desired bounds, at least not deterministically.

Finally, if we assume that the receiver is equipped with two (or more) mutually separated antennas (as in *multiple antenna systems* [20]), then a signal from some transmitter will most likely arrive at the antennas with different phases. Moreover, this shift between the phases of the signals received by will depend on the distances between the antennas as well as the relative position of the attacker with respect to the antennas. As we already saw above, at very high frequencies, even a very small distance shift will cause a significant phase shift. Any uncertainty in the distance shift (e.g., due to distance estimation errors, uncertainty regarding the positions of the antennas, etc.) implies uncertainty in the phase shift. We therefore conclude that it is reasonable to model phase shift  $\theta$  by a random variable with appropriate distribution.

### B. Randomization at the Sender: The Impact of Spreading

We already saw on Fig. 9 that for  $1/3$  of the possible phase shifts, the adversary actually attenuates the sender's signal. Therefore, when using only a single waveform (e.g.,  $\cos(\omega_0 t)$ ) during the whole period  $T_s$ , the adversary may have a non-negligible probability to attenuate the desired signal. For example, assuming  $\theta$  is a sample of a random variable  $\Theta$  with uniform distribution on  $[0, 2\pi)$ , the adversary attenuates the signal in the single time interval  $T_s$  with probability  $1/3$ . We now apply a solution similar to spreading, already described in Section III-C.

The idea is to split the time interval  $T_s$  into  $K$  smaller and equal time slots  $T_m$  when the symbol "1" is to be

<sup>4</sup>Not necessarily causing sufficient signal attenuation.

sent. Then, for each *mini-slot*  $T_m$ , the sender generates a signal with the phase chosen uniformly at random from  $[0, 2\pi)$  and emits these  $K$  signals on the channel during the time  $T_s$ . For example, these  $K$  signals can be described by the following random process  $S(t) = \cos(\omega_0 t + \Phi)$ , where  $\Phi$  is a random variable with uniform distribution on  $[0, 2\pi)$ .

From the discussion in the previous section, it is reasonable to model the phase shift as a random variable  $\Theta$ . Let us assume  $\Theta$  to be uniformly distributed on  $[0, 2\pi)$ ; later in this section, we also consider Gaussian distribution. Let  $p_\alpha$  be the probability that the adversary attenuates the signal emitted in a given mini-time slot for at least  $(1 - \alpha) \times 100\%$ , that is,  $E_r/E_s \leq \alpha$ , where  $\alpha \in [0, 1]$ . We say that any such mini-slot signal is  $\alpha$ -attenuated<sup>5</sup>. For  $\Theta$  uniform random variable, i.e.  $f_\Theta(\theta) = \frac{1}{2\pi}$ , we have

$$\begin{aligned} p_\alpha &= P \left[ \frac{E_r}{E_s} \leq \alpha \right] \\ &\stackrel{(1)}{=} P \left[ \sin \left( \frac{\theta}{2} \right) \leq \pm \frac{\sqrt{\alpha}}{2} \right] \\ &= P [\theta \in [0, \theta_\alpha] \cup (2\pi - \theta_\alpha, 2\pi)] \\ &\stackrel{(2)}{=} \frac{\theta_\alpha}{\pi}, \end{aligned} \quad (4)$$

where  $\theta_\alpha = 2 \arcsin(\sqrt{\alpha}/2)$ , the equality (1) follows from expression (2), and the equality (2) follows from the distribution of  $\Theta$ .

We further note that  $\Phi$  and  $\Theta$  are independent random variables; indeed,  $\Theta$  models the inability of the adversary to perfectly estimate the required distances and/or any delay that the adversary introduces. Therefore,  $p_\alpha$  (as given in expression (4)), is the same for all the  $K$  mini-slots. Then, for the fixed time interval  $T_s$ , the probability that the number  $K_a$  of  $\alpha$ -attenuated mini-slot signals is exactly  $k \leq K$ , can be calculated from the binomial distribution with parameters  $p = p_\alpha$  and  $q = 1 - p_\alpha$  as follows

$$P[K_a = k] = \binom{K}{k} \frac{1}{\pi^K} \theta_\alpha^k (\pi - \theta_\alpha)^{K-k}, \quad (5)$$

where  $\theta_\alpha = 2 \arcsin(\sqrt{\alpha}/2)$ . For the binomial distribution (5), we can calculate the expected ratio  $K_a/K$  of the  $\alpha$ -attenuated mini-slots as follows,

$$E \left[ \frac{K_a}{K} \right] = \frac{E[K_a]}{K} = \frac{\theta_\alpha}{\pi} \leq \frac{1}{3}, \quad (6)$$

where the last inequality follows from the fact that  $\theta_\alpha \leq \theta_1 = \frac{\pi}{3}$ . Therefore, on average, at most 1/3 of the total

<sup>5</sup>Note that even if the adversary does attenuate the energy of the original signal  $s(t)$  by 50%, the average power as measured by the receiver may still be well above the threshold  $P_0$ .

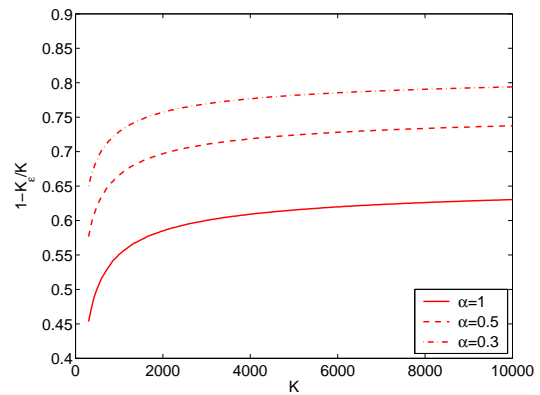


Fig. 11. The ratio of mini-slot signals that are not  $\alpha$ -attenuated as a function of  $K$ ;  $\epsilon = 10^{-14}$ .

number of mini-slot signals will be  $\alpha$ -attenuated, i.e.,  $E_r/E_s \leq \alpha$ .

Note, however, that the expected value of the ratio  $K_a/K$  is independent of  $K$ , and therefore it does not give any useful information about the role of  $K$  and what value we should choose for it. We next study this aspect. Let us denote with  $K_\epsilon$  ( $K_\epsilon \leq K$ ) the *smallest* threshold for which the following holds

$$P[K_a \leq K_\epsilon] \geq 1 - \epsilon, \quad (7)$$

where  $\epsilon \in [0, 1]$ . Note that  $P[K_a \leq K_\epsilon] = \sum_{k=0}^{K_\epsilon} P[K_a = k]$ , with  $P[K_a = k]$  given by (5). Note further that  $P[K_a \leq K_\epsilon]$  is related to a single time interval  $T_s$  during which the symbol “1” is transmitted. By the independence, the probability  $P^n[K_a \leq K_\epsilon]$  that  $K_a \leq K_\epsilon$  after  $n$  symbol “1” transmissions ( $n$  time intervals  $T_s$ ) satisfies

$$P^n[K_a \leq K_\epsilon] \geq (1 - \epsilon)^n \approx e^{-n\epsilon},$$

where the last approximation is valid for small  $\epsilon$ . For the given  $n$ , by choosing  $\epsilon$  such that  $e^{-n\epsilon}$  is reasonably close to 1, we essentially make  $K_\epsilon$  an “upper bound” on the number of mini-slot signals that are  $\alpha$ -attenuated in any given time slot  $T_s$  (out of the total of  $n$  slots). Likewise,  $(K - K_\epsilon)$  provides a “lower bound” on the number of mini-slot signals that are not  $\alpha$ -attenuated.

On Fig. 11, we plot the ratio  $(1 - K_\epsilon/K)$  of the mini-slot signals that are not  $\alpha$ -attenuated as a function of  $K$ , for  $\epsilon = 10^{-14}$ . For  $n = 10^{10}$ , we have  $e^{-n\epsilon} \approx 0.0001$ , i.e., even after as many as  $10^{10}$  transmissions of the symbol “1”, the probability that  $K_a \leq K_\epsilon$  is at least 0.9999. If we transmit on average one symbol 1 per second (meaning that we do nothing else but transmitting such signals), then it takes around 310 years to see all the  $n$  symbols. In this case, the smallest  $K_\epsilon$  for which the bound (7) holds, is a reasonable upper bound on  $K_a$ .

Coming back to Fig. 11, we can see that if  $K$  is set too low, we cannot hope to achieve a very high ratio of non  $\alpha$ -attenuated mini-slot signals for all the  $n$  transmissions of the symbol “1”. Therefore,  $K$  should be chosen based on the expected  $\alpha$  and the desired ratio  $1 - K_\epsilon/K$ .

### C. Energy Content of the Emitted Signals

We already argued that it is reasonable to model the phase shift as a random variable  $\Theta \in [0, 2\pi)$ . It is then interesting to calculate the energy of the resulting random signal. Let us define a random process  $R(t) = \cos(\omega_0 t) - \cos(\omega_0 t - \Theta)$ . We will calculate the energy of this process for two different distributions of  $\Theta$ , namely, uniform distribution on  $[0, 2\pi)$  and Gaussian distribution with zero mean and variance  $\sigma_\theta^2$ .

**Uniform distribution of  $\Theta$ .** We have  $f_\Theta(\theta) = \frac{1}{2\pi}$ ,  $\forall \theta \in [0, 2\pi)$ . The energy content  $\mathcal{E}_R$  of the random process  $R(t)$ , within the time interval  $T$ , is defined as [20]:

$$\mathcal{E}_R = E \left[ \int_0^T R^2(t) dt \right] = \int_0^T E [R^2(t)] dt . \quad (8)$$

Now, for  $E [R^2(t)]$  we have:

$$\begin{aligned} E [R^2(t)] &= \int_0^{2\pi} r^2(t) f_\Theta(\theta) d\theta \\ &= \frac{1}{2\pi} \int_0^{2\pi} (\cos(\omega_0 t) - \cos(\omega_0 t - \theta))^2 d\theta \\ &= 1 + \frac{1}{2} \cos(2\omega_0 t) . \end{aligned} \quad (9)$$

Plugging this into the expression (8), we obtain:

$$\begin{aligned} \mathcal{E}_R &= \int_0^T \left( 1 + \frac{1}{2} \cos(2\omega_0 t) \right) dt \\ &= T + \frac{\sin(2\omega_0 T)}{4\omega_0} \\ &\stackrel{(1)}{\approx} T , \end{aligned} \quad (10)$$

where (1) is valid for high frequencies  $f_0$ , since  $-1 \leq \sin(\cdot) \leq 1$  implies  $\sin(\cdot)/(4\omega_0) = \sin(\cdot)/(8\pi f_0) \rightarrow 0$ .

Therefore, on average, the adversary only increases the energy of the resulting signal  $r(t)$ ; the energy content of  $r(t)$  without the adversary is  $T/2$  (Fig. 9)!

**Gaussian distribution of  $\Theta$ .** It is reasonable to assume that the adversary cannot perfectly estimate the distances between himself and both the sender and the receiver. This imperfection can be captured by considering the distance shift  $\Delta d$  to be a random variable, i.e., we can assume  $\Delta d$  to be a Gaussian random variable with zero mean and variance  $\sigma_d^2$ . From expression (3),  $\Theta$  is also a Gaussian random variable with zero mean

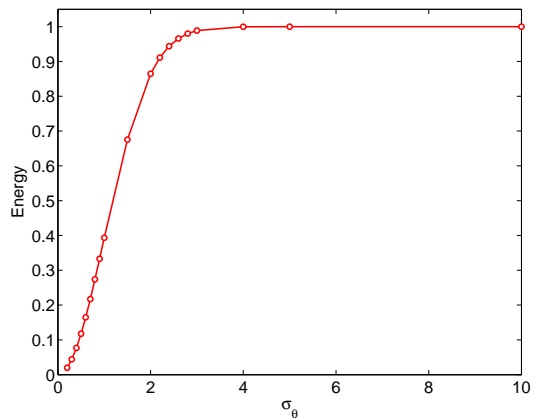


Fig. 12. The energy of  $R(t)$  (normalized to  $T$ ) for  $\Theta$  Gaussian variable with variance  $\sigma_\theta^2$ .

and variance  $\sigma_\theta^2 = (2\pi f_0/c)^2 \sigma_d^2$ . To calculate the energy content of  $R(t)$ , we proceed as in the case of the uniform distribution.

$$\begin{aligned} E [R^2(t)] &= \int_{-\infty}^{\infty} r^2(t) f_\Theta(\theta) d\theta \\ &= \int_{-\infty}^{\infty} (\cos(\omega_0 t) - \cos(\omega_0 t - \theta))^2 \\ &\quad \times \frac{1}{\sqrt{2\pi}\sigma_\theta} e^{-\theta^2/(2\sigma_\theta^2)} d\theta \end{aligned} \quad (11)$$

By plugging  $E [R^2(t)]$  in the expression (8), we obtain the expression for the energy content of the random process  $R(t)$ , with  $\Theta$  being the Gaussian variable. On Fig. 12 we plot the resulting values of the energy as a function of  $\sigma_\theta$ , for  $f_0 = 5$  GHz. As before, on average, the adversary increases the energy of the resulting signal, except for the low standard deviation  $\sigma_\theta = 1.189$  rad; note that this corresponds to  $\sigma_d = \sigma_\theta/(2\pi f_0) = 1.14$  cm. In addition, the adversary “only” halves the energy of the original signal  $s(t)$  for  $\sigma_\theta = 0.7578$  rad; this value corresponds to  $\sigma_d = 7.236$  mm.

From the analysis in this section, we conclude that we can easily ensure that the adversary cannot block the symbol “1” emitted over a radio channel, even under very advantageous assumptions for him (i.e., no multi-path fading effects, perfect estimate of signal amplitudes, etc.).

## VII. RELATED WORK

Providing integrity and authentication over insecure (radio) channels is a very active area of research. This provision has mainly focused on the key establishment after which the integrity and the authenticity of the messages is ensured by the use of known cryptographic techniques.

In this context, Stajano and Anderson propose the *resurrecting duckling* security policy model, [22] and [21], in which key establishment is based on the physical contact between communicating parties (their PDAs). A physical contact acts as a *location limited channel*, which can be used to transmit a key (or a secret) in plaintext. Thus, no cryptography is required at this stage. The potential drawback of this approach is that the realization of a physical contact can be cumbersome with bulky devices (e.g., laptops).

An approach inspired by the resurrecting duckling security policy model is proposed by Balfanz et al. [7]. In this work, the authors go one step further and relax the requirement that the location limited channel has to be secure against passive eavesdropping; they introduce the notion of a *location-limited channel* (e.g., an infrared link). A location-limited channel is used to exchange pre-authentication data and should be resistant to active attacks (e.g., man-in-the-middle). Once pre-authentication data are exchanged over a location-limited channel, users switch to a common radio channel and run any standard key exchange protocol over it. Possible candidates for a location-limited channel include: physical contact, infrared, and sound (ultrasound) [7]. Here again, the disadvantage of this approach is that it may be cumbersome to realize a link with bulky devices (e.g., laptops) in the case of infrared or physical contact. In addition, the infrared link itself is not well studied in the context of secure communications. Our key establishment mechanisms based on *I*-codes enable key establishment over a radio channel in a more practical way for the user as no physical contact is required.

Asokan and Ginzboorg propose another solution based on a shared password [6]. They consider the problem of setting up a session key between a group of people (i.e., their computers) who get together in a meeting room and who share no prior context. It is assumed that they do not have access to public key infrastructure or third party key management services. The proposed solution is the following. A fresh password is chosen and shared among those present in the room (e.g., by writing it on a sheet of paper or a blackboard). The shared password is then used to derive a strong shared session key. This approach requires users to type the chosen password into their personal devices.

It is well known that IT security systems are only as secure as their weakest link. In most IT systems the weakest links are the users themselves. People are slow and unreliable when dealing with meaningless strings, and they have difficulties remembering strong passwords. In [19], Perrig and Song suggest using hash visualization to improve the security of such systems. Hash visualiza-

tion is a technique that replaces meaningless strings with structured images. However, having to compare complex images can be cumbersome.

In US patent no. 5,450,493 [16], Maher presents several methods to verify DH public parameters exchanged between users. This technique had a flaw, discovered by Jakobsson [13]. Motivated by the flaw, Jakobsson [13] and Larsson [15] proposed two solutions based on a temporary secret shared between the two users (thus, for example, SHAKE stands for *Shared key Authenticated Key Exchange*). In our paper, we consider the same problem but in a more demanding setting, as we assume that the users share no secret key prior to the key exchange.

In [10] and [11], Gehrman et. al., propose a set of techniques to enable wireless devices to authenticate one another via an insecure radio channel with the aid of the manual transfer of data between the devices. In [2], we propose an optimal message authenticator, a more efficient protocol that enables provably secure authentication through the transfer of a short bit sequence over a secured channel. We further propose a set of simple techniques for key establishment over a radio link in peer-to-peer networks based on the Diffie-Hellman key agreement protocol.

In [8], Castelluccia and Mutaf propose an interesting movement-based pairing protocol for CPU-constrained devices. It is a pairing scheme that does not rely on public-key cryptography, out-of-band channels (such as a keyboard or a display) or specific hardware, making it inexpensive and suitable for CPU-constrained devices such as sensors. This protocol is an extension of the protocol initially proposed by Alpern and Schneider [5]. Alpern and Schneider present a protocol that allows two parties to agree on a secret key on channels for which an adversary cannot tell who is the source of each message.

We should mention other key-exchange protocols, proposed primarily for use in the Internet: IKE [3], JFK [4] and SIGMA [14]. All these protocols involve authentication by means of digital signatures, which clearly does not fit the problem we study here. We also should mention the work of Corner and Noble [9], who consider the problem of transient authentication between a user and his device, as well as the work of Čapkun et. al [23], where the authors show how to make use of users mobility to bootstrap secure communication in open ad hoc networks.

Finally, we acknowledge the contribution of Perrig et. al. in [18], where the authors propose Tesla, a protocol for broadcast authentication based on delayed key disclosure.

## VIII. CONCLUSION

In this paper, we introduced *integrity (I) codes*, a novel coding scheme that enables integrity protection of messages exchanged between entities that do not hold any mutual authentication material (i.e. public keys or shared secret keys). We have analyzed *I*-codes in detail and we have shown that they are secure in a realistic attacker model.

We further introduced a novel mechanism, called *authentication through presence* based on *I*-codes. We demonstrated the use of this mechanism in two application scenarios: broadcast authentication and key establishment.

We implemented *I*-codes on the Mica2 wireless sensor platform. We demonstrated that *I*-codes can be implemented efficiently and without the use of any specialized hardware.

## REFERENCES

- [1] Mica sensor platform. <http://www.xbow.com>.
- [2] Reference blinded for anonymity reasons.
- [3] RFC 2409 - The Internet Key Exchange (IKE). <http://www.ietf.org/rfc/rfc2409.txt?number=2409>.
- [4] W. Aiello, S. M. Bellovin, M. Blaze, R. Canetti, J. Ioannidis, A. D. Keromytis, and O. Reingold. Efficient, DoS-Resistant, Secure Key Exchange for Internet Protocols. In *Proceedings of ACM Computer and Communications Security (CCS) Conference*, pages 48–58, Washington, DC, 2000.
- [5] B. Alpern and F. Schneider. Key exchange using Keyless Cryptography. *Information processing letters*, 16(2):79–82, 1983.
- [6] N. Asokan and P. Ginzboorg. Key Agreement in Ad-hoc Networks. *Computer Communications*, 23(17):1627–1637, November 2000.
- [7] D. Balfanz, D. Smetters, P. Stewart, and H. Wong. Talking to Strangers: Authentication in Ad-Hoc Wireless Networks. In *Proceedings of the 9th Annual Network and Distributed System Security Symposium (NDSS)*, 2002.
- [8] C. Castelluccia and P. Mutaf. Shake Them Up! A movement-based pairing protocol for CPU-constrained devices. In *Proceedings of the ACM Conference on Mobile Systems, Applications and Services (MobiSys)*, 2005.
- [9] M. Corner and B. Noble. Protecting applications with transient authentication. In *First ACM/USENIX International Conference on Mobile Systems, Applications and Services (MobiSys'03)*, San Francisco, CA, May 2003.
- [10] C. Gehrmann, C.J. Mitchell, and K. Nyberg. Manual Authentication for Wireless Devices, 2004. RSA Cryptobytes, Vol. 7, No. 1.
- [11] C. Gehrmann and K. Nyberg. Enhancements to Bluetooth Baseband Security. In *Proceedings of Nordsec*, 2001.
- [12] S. Han, R. Rengaswamy, R. Shea, E. Kohler, and M. Srivastava. A Dynamic Operating System for Sensor Nodes. In *Proceedings of the ACM Conference on Mobile Systems, Applications and Services (MobiSys)*, 2005.
- [13] M. Jakobsson. Payments and Diffie-Hellman key exchange (presentation slides). <http://www.rsasecurity.com/rsalabs/-staff/bios/mjakobsson/teaching/index.html>.
- [14] H. Krawczyk. SIGMA. <http://www.ee.technion.ac.il/hugo/sigma.html>.
- [15] J.-O. Larsson and M. Jakobsson. SHAKE. Private communication with M. Jakobsson.
- [16] D.P. Maher. United States Patent (No. 5,450,493): Secure communication method and apparatus, 1993.
- [17] Wenbo Mao. *Modern Cryptography, Theory & Practice*. Prentice Hall PTR, 2004.
- [18] A. Perrig, R. Canetti, J.D. Tygar, and D. Song. The TESLA Broadcast Authentication Protocol. *RSA CryptoBytes*, 5(Summer), 2002.
- [19] A. Perrig and D. Song. Hash Visualization: A New Technique to Improve Real-World Security. In *Proceedings of the 1999 International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC '99)*, pages 131–138, July 1999.
- [20] John G. Proakis and Masoud Salehi. *Communication Systems Engineering – Second Edition*. Prentice Hall, 2002.
- [21] F. Stajano and R. Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *Proceedings of the 7th International Workshop on Security Protocols*, 1999.
- [22] Frank Stajano. *Security for Ubiquitous Computing*. John Wiley & Sons, Ltd., 2002.
- [23] S. Čapkun, J.-P. Hubaux, and L. Buttyán. Mobility Helps Peer-to-Peer Security. *IEEE Transactions on Mobile Computing*, 2006.