Integrity (I) Codes: Message Integrity Protection and Authentication Over Insecure Channels

Mario Čagalj[†*] Srdjan Čapkun[§*] Ramkumar Rengaswamy[‡] Ilias Tsigkogiannis[‡]
Mani Srivastava[‡] Jean-Pierre Hubaux[†]

[†]I&C-LCA, EPFL [§]IMM, Technical University of Denmark [‡]EE-NESL, UCLA

mario.cagalj@epfl.ch sca@imm.dtu.dk ram@ucla.edu ilias@ee.ucla.edu
mbs@ucla.edu jean-pierre.hubaux@epfl.ch

Abstract

Inspired by unidirectional error detecting codes that are used in situations where only one kind of bit errors are possible (e.g., it is possible to change a bit "0" into a bit "1", but not the contrary), we propose integrity codes (I-codes) for a radio communication channel, which enable 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 I-codes enables a sender to encode any message such that if its integrity is violated in transmission over a radio channel, the receiver is able to detect it. In order to achieve this, we rely on the physical properties of the radio channel. We analyze in detail the use of I-codes on a radio communication channel and we present their implementation on a Mica2 wireless sensor platform as a "proof of concept". We finally introduce a novel concept called "authentication through presence" that 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 with respect to a realistic attacker model.

1. 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.).

[*]Equally contributing authors.

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. In the literature such codes are known as *All-Unidirectional Error-Detecting* codes and are used in situations where it is possible to change, for example, a bit "0" into a bit "1" but the contrary is not possible (except with a negligible probability) [5, 7, 6]. An all-unidirectional error-detecting code is able to detect any number of unidirectional errors in the given codeword; in other words, for a given error-detection code, no unidirectional error can transform a (valid) codeword into another (valid) codeword. Unidirectional error-detecting codes find application, for example, in the encoding of unchangeable data on digital optical disks [13].

Our main goal in this study is to propose a mechanism to protect the integrity of messages exchanged between entities in the presence of an adversary who tries to convince the entities to accept modified messages as being authentic. We do not attempt to increase the reliability of message transmission – actually, as we will see shortly, we will have to sacrifice the reliability of message transfer in order to achieve our goal. For these reasons, we find it appropriate to call the error-detecting codes simply *integrity codes* (I-codes).

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 3) 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 3).

To validate our concept, we implement and test I-codes, on-off keying and signal anti-blocking components on the 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 particulary 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.

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 attackers ability to detect, jam and alter the messages on the channel.

The paper is organized as follows. In Section 2, we state our problem and we describe our system and the attacker model. In Section 3, we formally introduce I-codes and we provide details about their properties. In Section 4, we present the results of the I-codes implementation. In Section 5, we show how to use I-codes for authentication. In Section 6, we present the security analysis of I-codes. In Section 7 we describe the related work. Finally, we conclude the paper in Section 8.

2 Problem Statement and Assumptions

We observe the following problem: *Assuming that two entities (A and B) share a common (radio) 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) between A and B. 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 5 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 6).

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. Figure 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 in Figure 1(a) and Figure 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 in Figure 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.

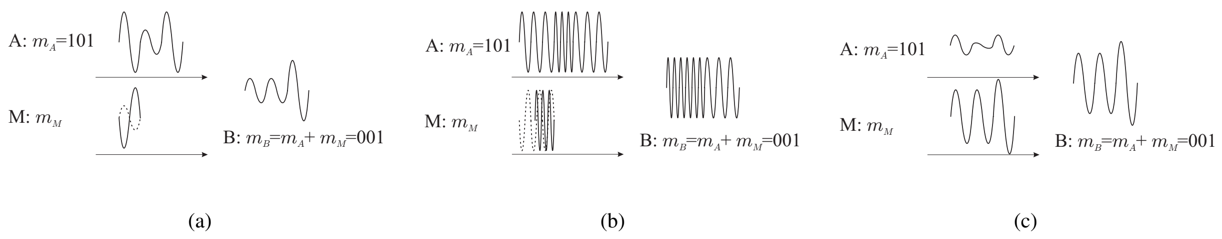
A: m_A=101
M: m_M
B: m_B=m_A+ m_M=001
(a)
A: m_A=101
M: m_M
B: m_B=m_A+ m_M=001
(b)
A: m_A=101
M: m_M
B: m_B=m_A+ m_M=001
(c)

Figure 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.*

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.

3 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 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.

3.1 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:*
   - e *is an injective function*
   - *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, Manchester code.) *The encoding rule* (e) *is the following:*

$$\begin{array}{ccc} 1 & \longrightarrow & 10 \\ 0 & \longrightarrow & 01\,. \end{array}$$

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* (ℓ) *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* ℓ *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* e*:* $00 \rightarrow 0111$*,* $01 \rightarrow 1011$*,* $10 \rightarrow 1101$ *and* $11 \rightarrow 1110$*. 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. The Merkle one-time signature scheme is also based on codes with fixed (known) Hamming weight [17].*

In the following section, we show how I-codes can be used on a *radio channel* to ensure the message integrity.

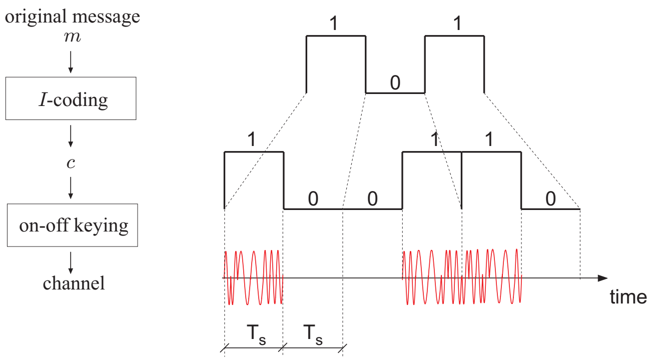

Figure 2. I-coding *An example of I-coding at the sender using the complementary encoding rule:* $1 \rightarrow 10$ *and* $0 \rightarrow 01$.

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.

3.2 I-codes on the Radio Channel

Let us consider the simple example shown in Figure 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 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 (Figure 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[1].

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 pre-defined *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".

[1]Note that this is similar to the *pulse position modulation* (PPM).

In our example in Figure 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 Figure 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 in Figure 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).

3.3 Preventing the attacker from erasing symbol "1"

In order to erase the signal (symbol "1") from the channel, 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 the randomness of the signal 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". It is important to stress that this measure makes sense only if the designated receiver can demodulate the signal at approximately the same speed as the attacker. Specifically, 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, in Figure 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.

In Section 6, we analyze in greater detail the effects of the randomness of the radio signal on the attacker's ability to erase the signal from the channel.

3.4 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).*

Clearly, the approach to synchronization of the previous example is not very flexible. We next describe a more flexible approach. 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) the following repetitive sequence

$$\ldots\text{delimiter}\ \overbrace{1010011001}^{c}\ \text{delimiter}\ \overbrace{1010011001}^{c}\ \text{delimiter}\ldots \tag{1}$$

Here, "delimiter" represents a specially constructed bit string such that any *successfully demodulated codeword*[2] received between any two consecutive "delimiters" is authentic (i.e., corresponds to 1010011001 in our example). We will show shortly how to construct such a delimiter for the complementary encoding rule.

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 consecutive "delimiters". If the codeword can be converted back to a message using the inverse of the complementary encoding rule (i.e., $(10 \to 1, 01 \to 0)$), the receiver accepts this message as being authentic. At this stage, the peer sender can stop transmitting the above repeated sequence. A nice property of this approach is that the receiver does not have to know the length of the codeword being transmitted in advance.

We next define more formally the notion of "delimiter". Then we construct the delimiter for the complementary encoding rule.

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 consecutive 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 consecutive bits) of any $c \in \mathcal{C}$, without flipping at least one bit* "1" *of the i-delimiter to bit* "0"*;*
3. *Any valid codeword (i.e., any $c \in \mathcal{C}$) received between two consecutive i-delimiters is authentic.*

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 $10100110 \to 10\mathbf{110110}$, i.e., it is sufficient to flip

[2]In our example, by "successfully demodulated codeword" we mean the codeword for which the transformation $(10 \to 1, 01 \to 0)$ exists.

only the fourth bit of c so that x emerges as the substring of c. Therefore, the first condition of Definition 2 is not met.

Assuming that an adversary cannot flip bit "1" into bit "0", we have the following result.

Theorem 2 *Consider the set of codewords $\mathcal{C}$ obtained by applying the encoding rule $(1 \rightarrow 10, 0 \rightarrow 01)$ to the set of source states (messages) $\mathcal{S} = \{0, 1, 00, 01, \ldots, \overbrace{11\ldots1}^{k}\}$, for arbitrary $k < \infty$. A string 111000 is an i-delimiter for the set $\mathcal{C}$.*

Proof. By mere inspection of all the strings of a length smaller than 6 bits, it easily follows that no such string satisfies Definition 2.

Consider now the string 111000. Observe that for every codeword $c \in \mathcal{C}$ the number of consecutive bits 0 and the number of consecutive 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". Thus, the string 111000 satisfies the first two conditions in Definition 2.

We next show that it satisfies the third condition as well. We observe that it is sufficient to focus on a codeword between two consecutive strings 111000, since three consecutive bits "1" never appear in any valid codeword from $\mathcal{C}$ and the adversary cannot flip a bit "1". Let us consider the following sequence of bits for any k-bit codeword (k being even) $c = (c_1c_2 \ldots c_{k-1}c_k) \in \mathcal{C}$

$$\ldots 111000\, c_1c_2 \ldots c_{k-1}c_k\, 111000 \ldots \tag{2}$$

We first show that the adversary cannot accomplish that the string 111000 emerges in any (other) part of the sequence (2) and that at the same time any resulting codeword $\hat{c}$ is valid. As the result the only hope for the adversary is to leave the original delimiters 111000 intact and try to transform the original codeword c into a different codeword $\hat{c}$ of the same length. Since c is an I-code codeword, the adversary would have to flip at least one bit "1" of c into a bit "0". However, by assumption he cannot accomplish this.

We now prove that the adversary cannot achieve that the 111000 emerges in any (other) part of the sequence (2) and that at the same time any resulting codeword $\hat{c}$ is valid. For this, let us consider all possible 6-bit substrings (of consecutive bits) in the sequence (2). These can be captured by one of the eleven cases given below:

1. $1\boxed{11000c_1}c_2 \ldots c_{k-1}c_k111000$
2. $11\boxed{1000c_1c_2}c_3 \ldots c_{k-1}c_k111000$
3. $111\boxed{000c_1c_2c_3}c_4 \ldots c_{k-1}c_k111000$
4. $1110\boxed{00c_1c_2c_3c_4}c_5 \ldots c_{k-1}c_k111000$
5. $11100\boxed{0c_1c_2c_3c_4c_5}c_6 \ldots c_{k-1}c_k111000$
6. $111000 \ldots c_{i-4}\boxed{c_{i-3}c_{i-2}c_{i-1}c_ic_{i+1}c_{i+2}} \ldots 111000$
7. $111000c_1c_2 \ldots c_{k-5}\boxed{c_{k-4}c_{k-3}c_{k-2}c_{k-1}c_k1}11000$
8. $111000c_1c_2 \ldots c_{k-4}\boxed{c_{k-3}c_{k-2}c_{k-1}c_k11}1000$
9. $111000c_1c_2 \ldots c_{k-3}\boxed{c_{k-2}c_{k-1}c_k111}000$
10. $111000c_1c_2 \ldots c_{k-2}\boxed{c_{k-1}c_k1110}00$
11. $111000c_1c_2 \ldots c_{k-1}\boxed{c_k11100}0$

Case 2 – *Case* 5. The strings $(1000c_1c_2)$, $(000c_1c_2c_3)$, $(00c_1c_2c_3c_4)$ and $(0c_1c_2c_3c_4c_5)$ cannot be transformed into the string 111000 without flipping at least one bit "1", since $c_1 \oplus c_2 = 1$ and $c_3 \oplus c_4 = 1$ (by the complementary encoding).

Case 6. We showed at the beginning of the proof that the string 111000 satisfies the condition one in Definition 2. So no string $(c_{i-3}c_{i-2}c_{i-1}c_ic_{i+1}c_{i+2})$, $i \in [4, 5, \ldots, k-2]$, can be transformed into the string 111000 without flipping at least one bit "1".

Case 7 – *Case* 11. The strings $(c_{k-4}c_{k-3}c_{k-2}c_{k-1}c_k1)$, $(c_{k-3}c_{k-2}c_{k-1}c_k11)$, $(c_{k-2}c_{k-1}c_k111)$, $(c_{k-1}c_k1110)$ and (c_k11100) cannot be converted into the string 111000 without filliping at least one bit "1", since they all contain at least one bit "1" among the last three digits.

Case 1. The string $(11000c_1)$ can be transformed into the string 111000 by flipping the third bit to "1", conditioned on $c_1 = 0$. In this case, the bit $c_2 = 1$ becomes the first bit of the new codeword $\hat{c}$ (not necessarily valid). From *Case* 2 to *Case* 11 above we know that the ending of the codeword $\hat{c}$ must be denoted either by the original delimiter 111000 or by the delimiter obtained by joining the first bit "1" of the original delimiter to the new codeword $\hat{c}$. In the first case, the length of the resulting codeword $\hat{c}$ is $k-1$ (an odd number) and so $\hat{c}$ cannot be a valid codeword. In the second case, one bit "1" is added to the sequence that already has a deficit of bits "0" (i.e., the bit $c_1 = 0$ is not a part of $\hat{c}$) and so the resulting codeword $\hat{c}$ cannot be not valid.

We conclude the proof by observing that the string 111000 is the shortest string (i.e., 6 bits long) that satisfies all the conditions in Definition 2. ■

Remark 1 *It is interesting to observe that for the complementary encoding rule and the delimiter* 111000*, the first two conditions from Definition 2 imply the third one (they are sufficient). If this holds in general (for any I-code and an i-delimiter) is an interesting open problem.*

Referring back to the example (1), the sender can preserve the integrity of message 11010 (i.e., the codeword

$c = 1010011001$) by simply emitting (using the on-off keying) 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 decodes a codeword received between any two consecutive i-delimiters (after having verified that the peer sender is active). According to Theorem 2, any successfully demodulated codeword between two i-delimiters must have been emitted by the peer sender (the codeword is authentic). At this stage, the peer sender can stop transmitting the above repeated sequence. The important implication of the synchronization based on i-delimiters is that the receiver does not have to know in advance the length of the message to be transmitted by the sender.

In the following sections, we report on our experience with the real-life implementation of I-codes and we introduce the novel concept of *authentication through presence*.

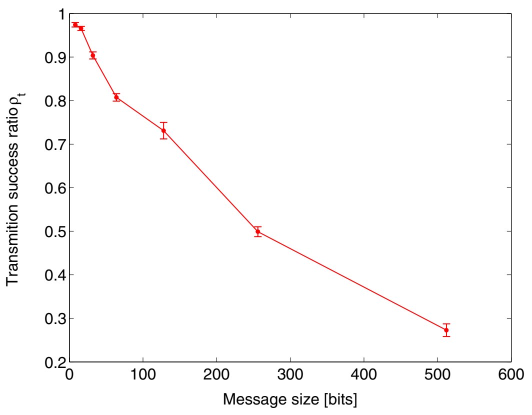

Figure 3. Robustness of I-codes. *The figure shows the message transmission success ratio ρ_t as a function of the size of transmitted messages. The results are obtained through measurements on Mica2 sensor motes.*

4 Implementation and Results

We implemented I-codes 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, 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 a 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 $T_s = 10$ ms). 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.

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 messages from 16 to 1024 bits). To transmit an ℓ-bit long message using I-codes, due to the complementary encoding, we actually transmit ℓ "0"s (10 ms of the absence of signal) and ℓ random packets (each 18 bytes long). We measured the message transmission success ρ_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. For each different message size, we perform 20 experiments as follows. We first generate 100 random messages of the given size. Next, we transmit these 100 messages and count the number of messages that have been successfully received. From this we calculate the success ratio ρ_t. Finally, we average the results obtained from 20 experiments and present them with 95% confidence interval.

The results of our measurements are shown in Figure 3. Quite expectedly, from Figure 3 we can observe that the transmission success ratio decreases quickly as the message size increases. These results further show that I-codes are best suited for reasonably short messages. For longer messages, we would need to transmit them multiple times in order for one of the messages to be transmitted correctly. For this purpose, we relay on the i-delimiters introduced in Section 3.4. 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 they are different from the messages sent by the sender. Moreover, with our implementation, no bit "1" sent by the transmitter was interpreted as a bit "0" on the receiver's side. This is important as it shows that the integrity of the messages transmitted with I-codes is preserved in our implementation.

From these measurements we conclude that I-codes pro-

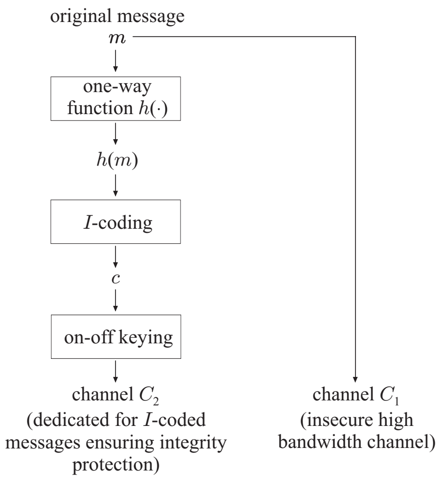
original message
m
one-way function $h(\cdot)$
$h(m)$
I-coding
c
on-off keying
channel C_2 (dedicated for I-coded messages ensuring integrity protection)
channel C_1 (insecure high bandwidth channel)

Figure 4. 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.*

vide 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.

5 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 introduce this concept and then we describe its use in two application

$$
\begin{array}{ll}
A \rightarrow B \text{ (on } C_1) : & m \\
A \rightarrow B \text{ (on } C_2) : & \ldots \underbrace{111000}_{i\text{-delimiter}} I\text{-code}\big(h(m)\big) \underbrace{111000}_{i\text{-delimiter}} \ldots
\end{array}
$$

B : Verify the integrity and the authenticity of $h(m)$ using I-codes.
Verify the integrity and the authenticity of m using $h(m)$.

Figure 5. Authentication through presence. *An example of the protocol that enables authentication through presence property; $h(\cdot)$ represents a one-way function.*

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 in Figure 4. 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 4 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 in Figure 5.

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-code$\big(h(m)\big)$ represents the I-coded message digest $h(m)$. The sequences preceding and following after I-code$\big(h(m)\big)$ are i-delimiters (Section 3.4), 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 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 legitime sender on the channel.

Remark 2 *The protocol shown in Figure 5 has the drawback that the size of the cryptographic hash function $h(\cdot)$ increases over time, in order to compensate for fast (daily) advance in computational technology and computational power available to an adversary. Today a "target collision-resistant" hash function implies the hash function size of at least 80 bits [14]. However, according to [14], the minimum required size increases linearly over time due to fast technological advances. It is not advisable to go below these minimum sizes in the adversarial model where the I-coded message in Figure 5 can be delayed sufficiently.*

Consequently, straightforward solutions similar to the one given in Figure 5 are said to be "time-variant", that is, the number of bits to be transmitted using I-codes increases over time. This is clearly not desirable in our context, since the reliability of I-codes drops quickly with the size of messages (see Figure 3). In Section 5.2, we will describe the protocol (in the context of user-friendly key establishment) that allows us to optimally trade-off the size of messages to be I-coded with the security, and to significantly decrease the size of I-coded messages.

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.

5.1 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 in Figure 6. 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 the user is located. 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

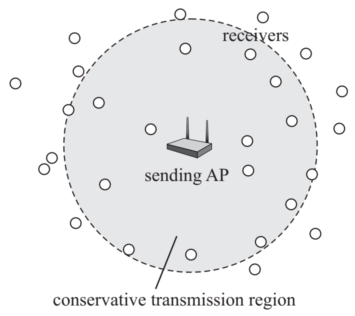

Figure 6. Broadcast integrity and authentication with an access point. *By the "conservative transmission region" we mean the region where the received power of a signal transmitted by the AP exceeds some predefined threshold level (which is a security parameter in our case).*

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 in Figure 4). 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).

5.2 Key Establishment Over Insecure Channels

In this section we show how authentication through presence can be used for user-friendly key establishment over an (insecure) radio link in peer-to-peer networks. We also show how it is possible to optimally trade-off the security of the key agreement protocol with the size of message to be transmitted using I-codes. This is particularly important, since I-codes are not developed with reliability in mind; as we saw in Figure 3, the reliability of I-codes drops quickly with the size of I-coded message.

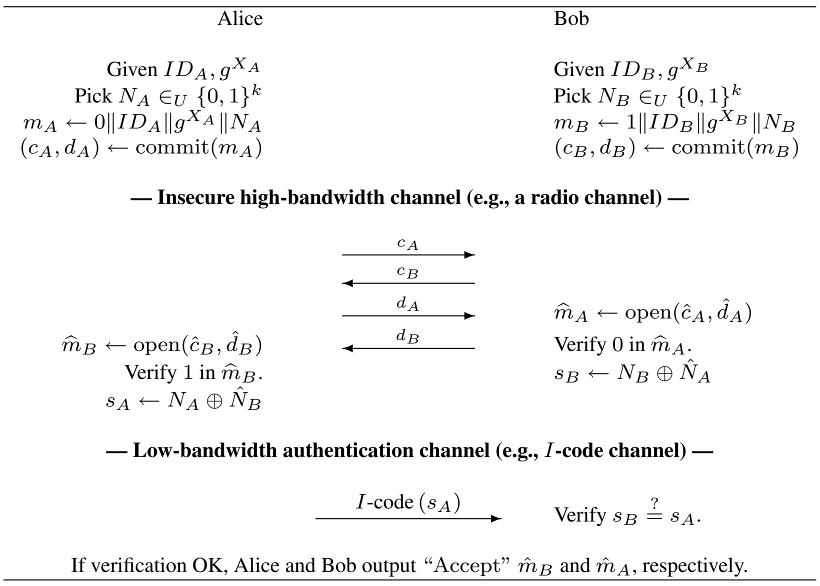
Alice
Bob
Given ID_A, g^{X_A}
Pick $N_A \in_U \{0,1\}^k$
$m_A \leftarrow 0\|ID_A\|g^{X_A}\|N_A$
$(c_A, d_A) \leftarrow \mathrm{commit}(m_A)$
Given ID_B, g^{X_B}
Pick $N_B \in_U \{0,1\}^k$
$m_B \leftarrow 1\|ID_B\|g^{X_B}\|N_B$
$(c_B, d_B) \leftarrow \mathrm{commit}(m_B)$
— Insecure high-bandwidth channel (e.g., a radio channel) —
c_A
c_B
d_A
d_B
$\widehat{m}_A \leftarrow \mathrm{open}(\hat{c}_A, \hat{d}_A)$
Verify 0 in $\widehat{m}_A$.
$s_B \leftarrow N_B \oplus \hat{N}_A$
$\widehat{m}_B \leftarrow \mathrm{open}(\hat{c}_B, \hat{d}_B)$
Verify 1 in $\widehat{m}_B$.
$s_A \leftarrow N_A \oplus \hat{N}_B$
— Low-bandwidth authentication channel (e.g., I-code channel) —
I-code (s_A)
Verify $s_B \overset{?}{=} s_A$.
If verification OK, Alice and Bob output "Accept" $\widehat{m}_B$ and $\widehat{m}_A$, respectively.

Figure 7. Diffie-Hellman key agreement protocol based on I-codes (DH^{IC})

In [23], we have developed a two-party key agreement protocol that is based on the Diffie-Hellman (DH) key agreement protocol. DH key agreement 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.

Our protocol unfolds as shown in Figure 7. Both Alice (A) and Bob (B) have selected their secret exponents X_A and X_B, respectively, randomly from the set $\{1, 2, \ldots, q\}$ (q being the order of an appropriate 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* [16]. 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 $\widehat{m}_A$. 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 $\widehat{m}_B$. If this verification is successful, A and B proceed to the final phase (Figure 7).

In the final phase, A and B first generate the authentication strings s_A and s_B, respectively, as shown in Figure 7 ($\oplus$ 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 DH^{IC} protocol can be found in [23]. Here we only state the result. We denote with γ 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.

Theorem 3 (cf. [23]) *The probability that an attacker succeeds against a targeted user of the DH^{IC} protocol is bounded by $\gamma 2^{-k}$.*

Our key agreement protocol exhibits two unique features: (i) it is optimal in the length of the message to be sent using I-codes, and (ii) it is *time-invariant*, i.e., in spite of the fact that the security parameters of the $\mathrm{commit}(\cdot)$ function increase over time, in order to compensate for fast (daily) advance in computational technology and computational power available to an adversary [14], the number of bits k to be transmitted using I-codes does not change over time (for the fixed security level). Note that that the protocol shown in Figure 5 is not time-invariant, since it relies on the cryptographic hash function $h(\cdot)$ whose security parameter increases over time [14].

Let us give an example of possible values for the parameters k and γ. Let us assume that the fixed user can participate in at most $\gamma = 2^{20}$ sessions (successful or abortive) in his/her lifetime; this corresponds to 32 sessions per day during approximately 89 years. Then, by choosing $k = 50$ (bits) we obtain that the highest probability of success by the adversary (having seen a huge number of 2^{20} DH^{IC} sessions by the fixed user) is $\gamma 2^{-k} = 2^{-30}$. Note that k also represents the length of the verification string s_A to be communicated through I-codes. From Figure 3, we can see that with I-codes and the complementary encoding rule, in normal circumstances, it will take on average around 2 repetitions of the 50 bit long verification string s_A, 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 and therefore the DH^{IC} is indeed user-friendly.

6 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 3.3, 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 2.

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.

6.1 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 $s(t)$; θ 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]:

$$\begin{aligned} E_r &= \int_0^{T_s} r^2(t)dt \\ &= \frac{1}{\omega_0}\sin^2\left(\frac{\theta}{2}\right)(2\omega_0 T_s - \sin(\theta) + \sin(\theta - 2\omega_0)) \\ &\overset{(1)}{\approx} 2T_s \sin^2\left(\frac{\theta}{2}\right) , \end{aligned} \tag{3}$$

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 $\sin(\cdot)/\omega_0 = \sin(\cdot)/(2\pi f_0) \to 0$.

We plot expression (3) in Figure 8; 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 θ 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 θ_0 is calculated as follows:

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

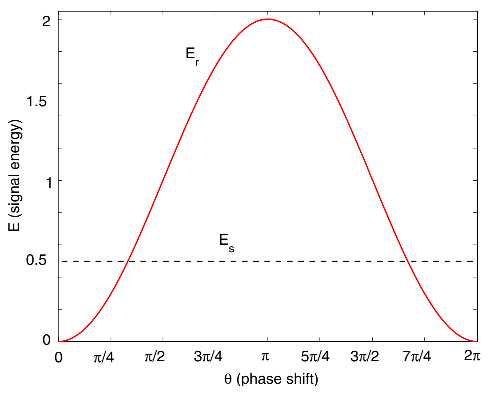

Figure 8. Signal energy. *The energy of the signal $r(t) \equiv \cos(\omega_0 t) - \cos(\omega_0 t - \theta)$ and the signal $s(t) = \cos(\omega_0 t)$ normalized with respect to T_s (the average power).*

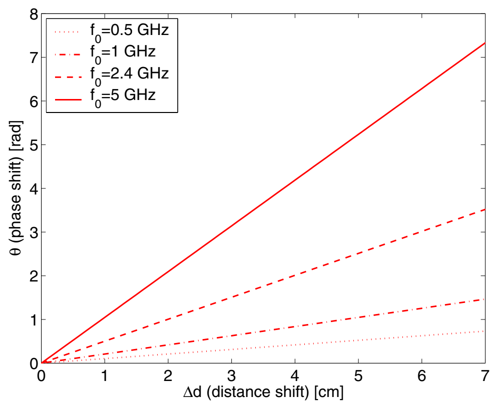

Figure 9. Phase shift. *The phase shift θ as a function of the distance shift Δd for different frequencies f_0.*

and therefore, $\theta_0 = 2\arcsin\left(\frac{1}{2}\right) = \frac{\pi}{3}$. Therefore, the attacker attenuates[3] $s(t)$ for $\theta \in \left[0, \frac{\pi}{3}\right) \cup \left(\frac{5\pi}{3}, 2\pi\right]$ (see Figure 8); 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 θ within the given bounds. We know that $\theta = \omega_0 \Delta t$, for a *time shift (delay)* Δt. In time Δt, the electromagnetic wave can travel the distance $\Delta d = \Delta t \cdot c$, where c is the propagation speed of the wave. We call Δd the *distance shift*. Combining these expressions we have:

$$\theta = \frac{2\pi f_0}{c} \Delta d \,. \tag{5}$$

In Figure 9 we plot expression (5) 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 Δd on the phase shift θ is. More importantly, for $f_0 = 5$ GHz (IEEE 802.11a), a Δd as small as 1 cm results in phase shift of $\frac{\pi}{3}$. As we discussed above, the adversary has to ensure that $\theta \in \left[0, \frac{\pi}{3}\right) \cup \left(\frac{5\pi}{3}, 2\pi\right]$, 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 Δd). 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 θ within the desired bounds.

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 received signals 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., 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 θ by a random variable with appropriate distribution.

6.2 Randomization at the Sender

We already saw in Figure 8 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 θ is a sample of a random variable Θ 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 show how to make this probability satisfactorily small.

The idea is to split the symbol interval T_s into K smaller

[3]Not necessarily causing sufficient signal attenuation.

and equal time slots T_m when the symbol "1" is to be 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 Φ 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 Θ. Let us assume Θ to be uniformly distributed on $[0, 2\pi)$. Let p_α 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 α-attenuated[4]. For Θ uniform random variable, i.e. $f_\Theta(\theta) = \frac{1}{2\pi}$, we have

$$\begin{aligned} p_\alpha &= P\left[\frac{E_r}{E_s} \leq \alpha\right] \\ &\overset{(1)}{=} P\left[\sin\left(\frac{\theta}{2}\right) \leq \pm\frac{\sqrt{\alpha}}{2}\right] \\ &= P\left[\theta \in [0, \theta_\alpha) \cup (2\pi - \theta_\alpha, 2\pi)\right] \\ &\overset{(2)}{=} \frac{\theta_\alpha}{\pi}\,, \end{aligned} \tag{6}$$

where $\theta_\alpha = 2\arcsin(\sqrt{\alpha}/2)$, the equality (1) follows from expression (4), and the equality (2) follows from the distribution of Θ.

We further note that Φ and Θ are independent random variables; indeed, Θ models the inability of the adversary to perfectly estimate the required distances and/or any delay that the adversary introduces. Therefore, p_α (as given in expression (6)), 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 α-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

$$P[K_a = k] = \binom{K}{k}\frac{1}{\pi^K}\theta_\alpha^k(\pi - \theta_\alpha)^{K-k}\,, \tag{7}$$

where $\theta_\alpha = 2\arcsin(\sqrt{\alpha}/2)$. For the binomial distribution (7), we can calculate the expected ratio K_a/K of the α-attenuated mini-slots as follows,

$$E\left[\frac{K_a}{K}\right] = \frac{E[K_a]}{K} = \frac{\theta_\alpha}{\pi} \leq \frac{1}{3}\,, \tag{8}$$

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

[4]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.

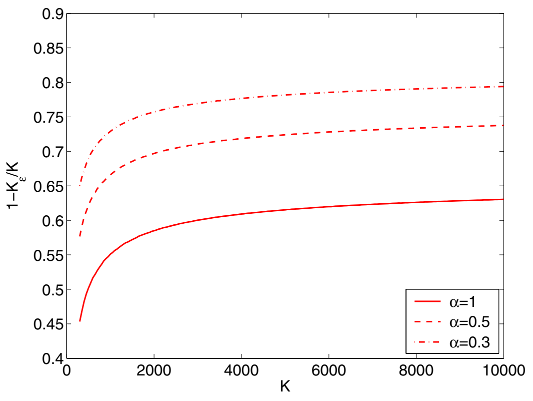

Figure 10. Attenuated mini-slot signals. *The ratio of mini-slot signals that are not α-attenuated as a function of K; $\epsilon = 10^{-14}$.*

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_ϵ ($K_\epsilon \leq K$) the *smallest* threshold for which the following holds

$$P[K_a \leq K_\epsilon] \geq 1 - \epsilon\,, \tag{9}$$

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 (7). 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 ϵ. For the given n, by choosing ϵ such that $e^{-n\epsilon}$ is reasonably close to 1, we essentially make K_ϵ an "upper bound" on the number of mini-slot signals that are α-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 α-attenuated.

In Figure 10, we plot the ratio $(1 - K_\epsilon/K)$ of the mini-slot signals that are not α-attenuated as a function of K, for $\epsilon = 10^{-14}$. For $n = 10^{10}$, we have $e^{-n\epsilon} \approx 0.9999$, 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_ϵ for which the bound (9) holds, is a reasonable upper bound on K_a. Coming back to Figure 10, we can see that if K is set too low, we cannot hope to achieve a very high ratio of non α-attenuated mini-slot signals for all the n transmissions of the symbol "1". Therefore, K should be chosen based on the expected α and the desired ratio $1 - K_\epsilon/K$.

6.3 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 when Θ has uniform distribution on $[0, 2\pi)$.

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]:

$$\mathcal{E}_R = E\left[\int_0^T R^2(t)dt\right] = \int_0^T E\left[R^2(t)\right] dt\,. \qquad (10)$$

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

$$\begin{aligned} E\left[R^2(t)\right] &= \int_0^{2\pi} r^2(t) f_\Theta(\theta) d\theta \\ &= \frac{1}{2\pi}\int_0^{2\pi} \left(\cos(\omega_0 t) - \cos(\omega_0 t - \theta)\right)^2 d\theta \\ &= 1 + \frac{1}{2}\cos(2\omega_0 t)\,. \end{aligned} \qquad (11)$$

Plugging this into the expression (10), we obtain:

$$\mathcal{E}_R = T + \frac{\sin(2\omega_0 T)}{4\omega_0} \overset{(1)}{\approx} T\,, \qquad (12)$$

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

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 $T/2$ (Figure 8)!

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 multipath fading effects, perfect estimate of signal amplitudes, etc.).

7 Related work

In this context, Stajano and Anderson propose the *resurrecting duckling* security policy model, [21] and [22], 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 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. [4]. In this work, the authors relax the requirement that the location limited channel has to be secure against passive eavesdropping; they introduce the notion of a *location-limited channel*. 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 ultrasound [4]. 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. 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 [3]. 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 except a fresh password.

In most IT security systems the weakest links are the users. 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 visualization 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 [15], Maher presents several methods to verify DH public parameters exchanged between users. This technique had a flaw, discovered by Jakobsson [12]. Motivated by the flaw, Larsson and Jakobsson [12] proposed two solutions based on a temporary secret shared between the two users.

In [10] and [11], Gehrmann 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 [23], we propose an optimal message authenticator, a more efficient protocol that enables provably secure authentication through the transfer of a short bit sequence over an authenticated 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 [2], 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. It is a pairing scheme that does not rely on public-key cryptography. As a follow-up, in [8], Castelluccia and Mutaf propose two movement-based pairing protocols for CPU-constrained devices. We 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. [24], where the authors show how to make use of users mobility to bootstrap secure communication in 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.

8 Conclusion

In this paper, we introduced *integrity (I) codes* for a radio channel, a novel mechanism 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.

Acknowledgments

The authors would like to thank the anonymous reviewers for the thorough reviews and many helpful suggestions.

References

[1] Mica sensor platform. http://www.xbow.com.

[2] B. Alpern and F. Schneider. Key exchange using Keyless Cryptography. *Information processing letters*, 16(2):79–82, 1983.

[3] N. Asokan and P. Ginzboorg. Key Agreement in Ad-hoc Networks. *Computer Communications*, 23(17):1627–1637, November 2000.

[4] 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.

[5] J. M. Berger. A Note on Error Detecting Codes for Asymmetric Channel. *Information and Control*, 4:68–73, 1961.

[6] M. Blaum and H. van Tilborg. On t-Error Correcting/All Unidirectional Error Detecting Codes. *IEEE Transactions on Computers*, pages 1493–1501, 1989.

[7] J. M. Borden. Optimal Asymmetric Error Detecting Codes. *Information and Control*, 53:66–73, 1982.

[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. Mitchell, and K. Nyberg. Manual Authentication for Wireless Devices, January 2004. RSA Cryptobytes, Vol. 7, No. 1.

[11] C. Gehrmann and K. Nyberg. Enhancements to Bluetooth Baseband Security. In *Proceedings of Nordsec*, Copenhagen, Denmark, November 2001.

[12] J.-O. Larsson and M. Jakobsson. SHAKE. Private communication with M. Jakobsson.

[13] E. L. Leiss. Data Integrity on Digital Optical Discs. *IEEE Transactions on Computers*, 33:818–827, 1984.

[14] A. K. Lenstra and E. R. Verheul. Selecting Cryptographic Key Sizes. *Journal of Cryptology*, 14(4):255–293, 2001.

[15] D. Maher. United States Patent (No. 5,450,493): Secure communication method and apparatus. http://www.uspto.gov, 1993.

[16] W. Mao. *Modern Cryptography, Theory & Practice*. Prentice Hall PTR, 2004.

[17] R. C. Merkle. A Digital Signature Based on a Conventional Encryption Function. In C. Pomerance, editor, *Lecture Notes in Computer Science (CRYPTO'87)*, volume 293, pages 369–378. Springer-Verlag, 1988.

[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] J. G. Proakis and M. Salehi. *Communication Systems Engineering – Second Edition*. Prentice Hall, 2002.

[21] F. Stajano. *Security for Ubiquitous Computing*. John Wiley & Sons, Ltd., 2002.

[22] 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.

[23] M. Čagalj, S. Čapkun, and J.-P. Hubaux. Key Agreement in Peer-to-Peer Wireless Networks. *Proceedings of the IEEE (Special Issue on Cryptography and Security)*, 94(2), 2006.

[24] S. Čapkun, J.-P. Hubaux, and L. Buttyán. Mobility Helps Peer-to-Peer Security. *IEEE Transactions on Mobile Computing*, 2006.