Integrity Regions: Authentication Through Presence in Wireless Networks

Srdjan Čapkun[*]
Department of Computer Science
ETH Zurich
8092 Zurich, Switzerland
capkuns@inf.ethz.ch

Mario Čagalj[*]
Faculty of Electrical Engineering, Mechanical
Engineering, and Naval Architecture (FESB)
University of Split
21000 Split, Croatia
mario.cagalj@fesb.hr

ABSTRACT

We introduce *Integrity (I) regions*, a novel security primitive that enables message authentication in wireless networks without the use of pre-established or pre-certified keys. Integrity regions are based on the verification of entity proximity through time-of-arrival ranging techniques. We demonstrate how I-regions can be efficiently implemented with ultrasonic ranging, in spite of the fact that ultrasound ranging techniques are vulnerable to distance enlargement and reduction attacks. We further discuss how I-regions can be used in key establishment applications in peer-to-peer wireless networks.

Categories and Subject Descriptors

C.0 [**Computer-Communication Networks**]: [Security and protection]

General Terms

Security, Wireless Networks

Keywords

Authentication, Key Establishment, Distance Bounding

1. INTRODUCTION

Wireless technology enables users to communicate, regardless of their micro-location, provided that their devices are located within each others' radio range. The extent of this range depends on devices' transmitting power, (transmitting and receiving) antenna gains, receiver sensitivity, and on various environmental factors [27]. Existing radio propagation models can estimate the reach of radio signals, but generally with low precision and only in specific environmental conditions; for most radio technologies, randomness introduced by the environmental conditions often makes the accurate estimation of the full extent of a radio range difficult if not impossible [27].

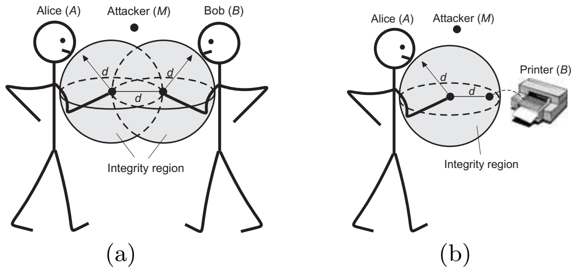
In an adversarial setting, this means that users can hardly predict the maximal distance from which an adversary can eavesdrop on their communication. Furthermore, adversary's receiver sensitivity and antenna gain can be much higher than the ones of the users; in this way, he/she can eavesdrop communication from a distance which is much larger the than users' communication range [27]. Moreover, an adversary can use devices with high transmission power; this enables it to insert, modify, and jam messages exchanged between users from large distances.

For these reasons, the communication between wireless devices that do not share any pre-established or pre-certified keys is subject to man-in-the-middle attacks (MITM) by adversaries that are located far from the devices. The reduction of devices' transmission ranges might seem like an appealing solution to alleviate this problem, but adversaries using high-gain antennas will still be able to perform MITM attacks (e.g. an attacker with a 30 dB gain antenna will boost a reception range of a signal by a factor of 30; this means that if devices communicate using a 1 m range, the attacker will still be able to perform MITM attacks from a 30 m distance). Recent attacks on Bluetooth [2] using such eavesdropping demonstrate in practice that the privacy of communication cannot be preserved through power range reduction. Given this, two wireless devices trying to establish a shared secret key (e.g. using the Diffie-Hellman (DH) protocol [10]) would not be able to establish the key securely and would be subject to the know MITM attacks on a non-authenticated DH protocol [7].

In this work, we introduce *Integrity regions (I-regions)*, a new security primitive that prevents MITM attacks on wireless communication. I-regions rely on range measurements to prevent distant attackers from inserting forged messages into the communication between the devices. In this approach, for each received message, devices verify if it came from an expected (typically short < 1 m) distance (i.e. from their integrity regions). If the message came from the expected (safe) distance, it is accepted as authentic. With I-regions, we force the attacker to be present in the close proximity of the devices in order to perform a man-in-the-middle attack. If the users can verify (i.e., visually) that the attacker (or an adversarial device) is not present in their immediate vicinity, they will be able to verify the integrity and the authenticity of the exchanged messages and therefore

[*] Equally contributing authors.

WiSe'06, September 29, 2006, Los Angeles, California, USA.
Copyright 2006 ACM 1-59593-557-6/06/0009 ...$5.00.



(a) (b)

Figure 1: Examples of applications of integrity regions. (a) Key establishment (e.g., exchange of the Diffie-Hellman public keys - bidirectional message authentication and integrity verification); (b) Device authentication (user receives an authentic public-key of a device - unidirectional message authentication and integrity verification).

prevent MITM attacks. Even in scenarios in which there is no visual verification of the space by users (e.g., in sensor networks), integrity regions force the attacker to come close to the devices (i.e., within devices' mutual distance) in order to successfully perform MITM attacks. As a result, the cost of mounting the attack can exceed by far the expected payoff (gain) for the attacker, thus rendering the attack uninteresting.

I-regions can be implemented with ultrasonic or radio time-of-arrival ranging techniques; due to its lower hardware requirements, in this work, we demonstrate the feasibility of I-regions using ultrasonic ranging (in face of its vulnerability to distance enlargement and reduction attacks).

The most straightforward application of I-regions is in key establishment in peer-to-peer wireless networks; using I-regions, users can establish a shared secret key by simply getting physically close and by verifying that the attacker is absent from their close proximity (the joint integrity region). This is illustrated in Fig. 1. The figure shows the integrity regions of users A and B (whose sizes are determined by their mutual distance d), and the location of attacker (M) placed outside of these integrity regions. In this setting, users A and B can, using I-regions, securely establish their shared secret key.

The rest of 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-regions and we provide details about their properties. In Section 4, we show the application of I-regions in the context of key establishment in peer-to-peer wireless networks. In Section 5 we describe the related work. Finally, we conclude the paper in Section 6.

2. PROBLEM STATEMENT

We consider the following problem. Two users, equipped with personal devices capable of communicating over a radio link, get together and want to establish a shared key (alternatively, a user approaches a device and wants to establish a key with this device). Although they can visually recognize each other, we assume that they do not share any authenticated cryptographic information (e.g., public keys or a shared secret) prior to this meeting. In addition, the users can communicate only over a radio channel (no infrared or physical ports are available). The challenge is the following: *How can the users establish a shared key in a secure way?*

Before discussing possible solutions, we first introduce our attacker model.

2.1 Attacker model

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. The attacker can further jam the transmission and in that way prevent the transmission of the information contained in the message. Finally, we assume M to be computationally bounded.

We classify attackers according to their antenna gain and transmitting power. Consequently, the attacker M with the highest antenna gain and transmitting power will be able to control the channel from the furthest distances. We do note however, that this space is finite given that the device's transmitting power and its receiver sensitivity are finite. Still, in our analysis, we will assume that the distance from which the attacker can control the channel is large (i.e., much larger than the devices' communication range).

2.2 Diffie-Hellman key agreement

The Diffie-Hellman (DH) key agreement protocol [10] seems to be appropriate for the problem (and the set of assumptions) at hand; the DH key agreement protocol is believed to be secure against a passive adversary[1] (e.g., eavesdropping on a wireless link). Let us briefly review how the DH key agreement protocol works. To agree on a shared key, two users, Alice (A) and Bob (B) proceed as follows. A picks a random secret exponent X_A, and calculates the DH public parameter g^{X_A}, where g is a generator of a group of large order. B does the same, that is, he calculates g^{X_B}. Finally, A and B exchange the public parameters g^{X_A} and g^{X_B} and calculate the shared DH key as $K = g^{X_A X_B} = (g^{X_A})^{X_B} = (g^{X_B})^{X_A}$.

2.3 Attacks

It is well known that the basic version of the DH protocol is vulnerable to man-in-the-middle attacks. At first glance, it may seem that mounting the MITM attack against wireless devices that communicate over a radio link and are located within the radio communication range of each other can be perpetrated only by a sophisticated attacker. But this is not the case; recently, a number of MITM attacks have been demonstrated against wireless networks. Techniques used in these attacks include ARP spoofing in 802.11 networks to jamming-based attacks in wireless sensor networks [36]. Here we describe these attacks in more detail.

The Address Resolution Protocol (ARP) [25] is used by the Internet Protocol (IP) to map IP network addresses to the hardware addresses used by a data link protocol. An attacker can send spoofed ARP-replies to the victim, who will

[1] This is true if the Computational Diffie-Hellman problem [22] is intractable.

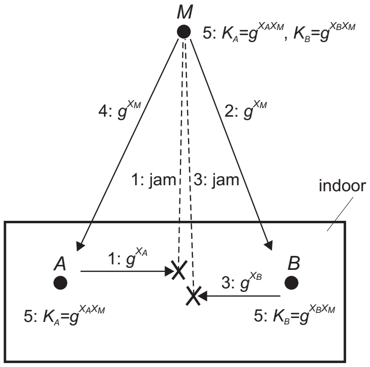
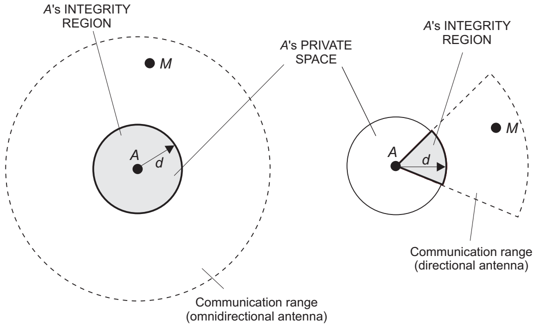


Figure 2: Man-in-the-middle attack on the Diffie-Hellman key agreement over wireless communication channel. The attacker is located far from the devices running the Diffie-Hellman protocol.


A's INTEGRITY REGION
M
A's PRIVATE SPACE
A
d
Communication range (omnidirectional antenna)
A's INTEGRITY REGION
M
A
d
Communication range (directional antenna)

(a) (b)

Figure 3: Examples of integrity regions for user A with (a) omnidirectional and (b) directional antennas.

consequently send all its packets to the attacking machine. In an experiment we conducted, we were able to redirect the traffic between two "legal" machines through an attacking machine, despite the fact that the two legal machines were in radio-communication range of each other. In this way, the attacker could perpetrate the MITM attack (by altering the DH parameters). For this attack we used a collection of publicly available tools for network auditing and penetration testing, called *dsniff* [29].

While ARP spoofing might be an attack somewhat specific to IP-based networks (although it is just an instance of a more general address spoofing attack), jamming-based attacks are more general and can be mounted on almost any wireless network (i.e., except for those networks applying anti-jamming communication techniques e.g., frequency hopping). We illustrate jamming-based MITM attacks by an example of a MITM attack on DH key agreement protocol; this attack is shown on Fig. 2. This figure emphasizes an important aspect of jamming-based MITM attacks: to successfully launch an attack, an attacker does not have to be located close to the devices running the protocol. This is due to two reasons: (i) an attacker can use high-gain antenna to eavesdrop on devices' communication and (ii) an attacker can jam devices' communication from large distances, using high-power antennas. With these two "tools" in hand, an attacker's *attack space* is large and he effectively controls the channel; thus, the attacker can schedule, modify and insert messages in the communication between the devices. As shown on Fig. 2, an attacker can then modify devices' contributions (g^{X_A}, g^{X_B}) to the common DH key and establish different keys with A and B, while they believe that they have established a common key.

Examples of more involved MITM attacks against Bluetooth [2] equipped devices can be found in [15] and [17].

In conclusion of this analysis, in this work, our goal is *to devise mechanisms that prevent the attacker from modifying the DH parameters on a radio channel without being noticed.*

2.4 Primitives

In this paper, we will present our solution over the multiplicative group $\mathbb{G}$ with the generator g. Here, we take $\mathbb{G}$ to be a subgroup of $\mathbb{Z}_p^*$ of the prime order q, where $\mathbb{Z}_p^*$ is the multiplicative group of non-zero integers modulo a large prime p. However, the whole treatment here applies to any group in which the Diffie-Hellman problem is hard. These are all groups in which it is infeasible to distinguish between quadruples of the form (g, g^x, g^y, g^{xy}) and quadruples (g, g^x, g^y, g^z) where x, y, z are random exponents. Furthermore, we assume that p and a generator g of $\mathbb{Z}_p^*$, $(2 \leq g \leq p-2)$ are selected and published. All devices are preloaded with these values[2].

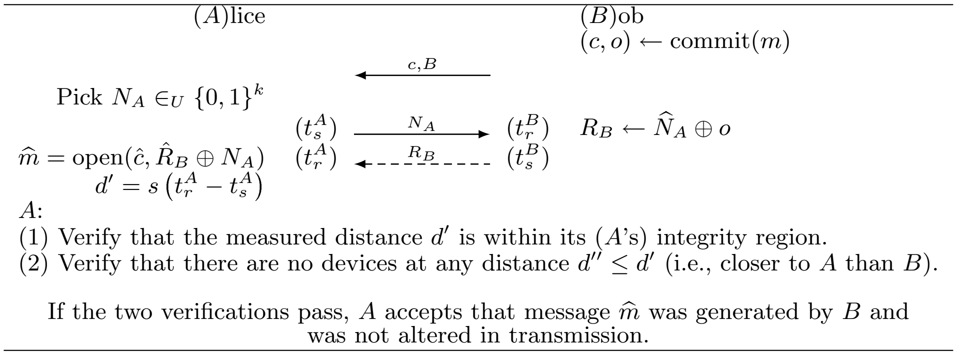
Commitment schemes are important cryptographic building blocks that we will be using in our protocol. In this subsection, we provide only an informal treatment of commitment schemes. The semantics of a commitment scheme are the following: (i) a user who commits to a certain value cannot change this value afterwards (we say that the scheme is binding), (ii) the commitment is hidden from its receiver until the sender opens it (we say that the scheme is hiding). A commitment scheme transforms a value m into a commitment/opening pair (c, o), where c reveals no information about m, but (c, o) together reveal m, and it is infeasible to find $\hat{o}$ such that $(c, \hat{o})$ reveals $\hat{m} \neq m$. Now, if Bob wants to commit a value m to Alice, he first generates the commitment/opening pair $(c, o) \leftarrow \text{commit}(m)$, and sends c to Bob. To open m, Alice simply sends $\hat{o}$ (and m if necessary) to Bob, who runs $\hat{m} \leftarrow \text{open}(\hat{c}, \hat{o})$; we denote with $\hat{m}$ the message at the receivers side when message m is sent over a public (unauthentic) channel. If the employed commitment scheme is correct, at the end of the protocol we must have $m = \hat{m}$. In our security analysis, we assume the usage of an ideal, non-malleable [19] commitment scheme.

3. INTEGRITY REGIONS

In this section, we formally introduce the concept of Integrity regions. First, we introduce a notion of the *private space.* We define it as follows:

Definition 1. *User's (device's) private space $PS(A)$ is the largest spherical space centered at user's (device's) location, within which the user can establish (e.g., visually)*

[2] We stress here that we could let users select and communicate to each other their own parameters p and g. However, this would come at the expense of the number (and size) of messages to be exchanged between the users, and our goal is to keep key exchange protocols as simple as possible.



(A)lice

(B)ob

$(c, o) \leftarrow \text{commit}(m)$

$\xleftarrow{c,B}$

Pick $N_A \in_U \{0,1\}^k$

$(t_s^A) \xrightarrow{N_A} (t_r^B)$ $\quad R_B \leftarrow \widehat{N}_A \oplus o$

$\widehat{m} = \text{open}(\hat{c}, \hat{R}_B \oplus N_A)$ $\quad (t_r^A) \xleftarrow{R_B} (t_s^B)$

$d' = s\left(t_r^A - t_s^A\right)$

A:

(1) Verify that the measured distance d' is within its (A's) integrity region.

(2) Verify that there are no devices at any distance $d'' \leq d'$ (i.e., closer to A than B).

If the two verifications pass, A accepts that message $\widehat{m}$ was generated by B and was not altered in transmission.

Figure 4: Message Transfer authenticator based on Integrity Regions (MT-IR). Device A verifies if the message was received from device B and if it's integrity was preserved. (the full and the dashed arcs represent radio and an ultrasonic channel, respectively). Note that N_A and R_B are exchanged bit-by-bit.

the presence (or absence) of other wireless (trusted and untrusted) devices, or within which the user (device) can assume the absence of adversaries.

User's (device's) private space is a space that the user (device) controls or knows. This space is typically very small (i.e., < 1 m in radius).

We now define integrity regions as follows.

Definition 2. *An integrity region $IR(A)$ of a user A is the subspace of its private space $PS(A)$, bounded by the device's communication channel directionality.*

The size of an integrity region is determined by the radius d of user's private space and depends on the directionality of device's antenna. Fig. 3 shows two examples of integrity regions, namely, devices with directional and omnidirectional antennas. Note here that the size of the integrity region is controlled by the user and is upper-bounded by the size of user's private space.

An integrity region is a user controlled space, in which the user (or a device) has sufficient confidence to establish keys. In what follows, we will show how we can use integrity regions for message authentication and integrity verification.

3.1 Message authentication and integrity verification in integrity regions

Having defined integrity regions, we now construct a *Message Transfer authenticator based on Integrity Regions* (MT-IR). With this protocol, a device can verify the integrity and the authenticity of messages received from other devices within its integrity regions.

In our description, we use the notation $\hat{x}$ to denote the message at the receiver's side when message x is sent over a public (unauthentic and insecure) channel. Before describing the protocol, we first describe our scenario. We observe two devices A and B. Device A controls its integrity region $IR(A)$ (i.e., is aware of the presence or absence of trusted and untrusted devices). We assume that device B is in A's integrity region $IR(A)$ and that B is the device (physically) closest to A. If this is initially not the case, we assume that A will approach B such that the above will hold (e.g., the user approaches a device or two users approach each other). For simplicity, and without any loss of generality, we will assume that B is the only device in the integrity region of A. A and B do not share any secrets or hold each-others' public keys. The devices are equipped with speakers and microphones and are able to perform ultrasonic ranging; an example of such devices are Cricket sensor motes [26].

In order to enable A to verify the integrity and authenticity of messages received from B, we construct our Message Transfer authenticator based on Integrity Regions (MT-IR). This protocol is shown on Fig. 4. In this protocol, node B first commits to the message that it wants to send to A, and sends a commitment c of the message to A. As already noted in Section 2.4, this commitment is both binding and hiding, and non-malleable [19]; it binds the sender to the message and hides the value of the message from the receiver. The verifier (A) then issues a fresh, randomly generated challenge N_A and measures the time until it received the response from B. This response is computed by B as a function of the challenge N_A and of the message decommitment o; it is sent over an ultrasonic communication channel and is received by A on its ultrasonic interface. After receiving this ultrasonic signal, device A then calculates the distance $d' = s(t_r^A - t_s^A)$ which corresponds to the measured time of flight of the ultrasonic signal $(t_r^A - t_s^A)$; here, s is the speed of sound, and the time of flight of the radio signal is neglected. Note here that N_A and R_B are exchanged bit-by-bit [8]. We do acknowledge that there some time is required for processing at the node B, but this time (< 100 μs) is negligible compared to the time of flight of the ultrasonic signal (approximately 2 ms for the distance of 70 cm). Node A therefore does not account for B's processing time and considers this time to be 0 s. By fixing the processing time to 0 s, we essentially turn this ranging protocol into a distance-bounding protocol on B [8, 35, 28].

Finalizing the protocol, A (i) verifies that the measured distance d' is within its (A's) integrity region; this verification can be done automatically by the device if the radius d of the integrity region is known and pre-defined in the device (in most application scenarios, this radius can be estimated to approximately 0.3-1 m), (ii) verifies (e.g., visually) that, except for B, there are no other devices within the integrity region at any distance $d'' \leq d'$ (i.e., closer to A than B).

If both verifications pass, A accepts that the message $\hat{m}$ A received was generated by device B and was not altered in transmission. In Section 3.2, we will show that the vulnerability of ultrasound ranging to distance enlargement and reduction attacks, does not give an adversary any advantage

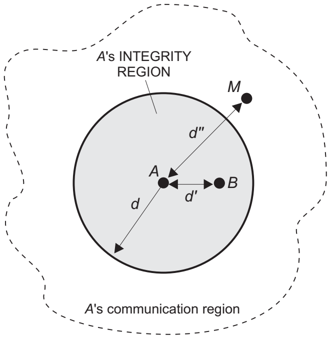


Figure 5: Message authentication and integrity verification with MT-IR. A accepts that the message $\hat{m}$ was generated by device B and was not altered in transmission only if the measured distance d' is within the integrity region ($d' \leq d$) and if there are no other (adversarial) devices closer to A than B is.

with respect to the MT-IR protocol. Therefore, a message is accepted if and only if it comes from a location within the integrity region and if there are no other (untrusted) devices closer to A than B. Note that the way that A identifies B is visually, and therefore associates the received message with the person (or the device) that it sees. This is illustrated on Fig. 1.

To summarize, in this protocol, message (m) is associated to a device (B), based on its distance to the verifying device (A). This association is conditional on the length of the measured distance (needs to fit within an integrity region), and on the presence of other devices in the region (other devices should not be closer to the verifier A than B is). This is illustrated on Fig. 5. MT-IR essentially forces an attacker to come close to the devices on which it wants to launch the attack (i.e. to enter its integrity region); if the attacker is outside of the integrity region, the attack will be detected.

In its construction, MT-IR protocol is similar to distance bounding protocols [8, 35, 28]. Distance-bounding protocols are designed to upper-bound distances between mutually authenticated devices. MT-IR differs from distance-bounding in that it uses range measurements within integrity regions to provide authentication and message integrity verification between devices that do not share any authentication material.

3.2 Security analysis

The presented MT-IR protocol achieves two goals: message authentication and integrity verification. In this protocol, message authentication and integrity verification is performed using measured ranges (i.e., the message is associated with a range, and the range with a device) and through commitment schemes that guarantee that the authenticated message was not modified in transmission. Given this, the security of the protocol depends on the inability of the attacker to manipulate measured distances and to break commitment schemes. In this analysis, we assume that used commitment scheme is ideal and we focus on ranging mechanisms.

In our analysis, we observe the scenario illustrated on Fig. 5. We assume that A controls its integrity region $IR(A)$ (i.e., is aware of the presence or absence of trusted and untrusted devices). We further assume that B is the only device in the integrity region of A and that the adversary M is located outside A's integrity region.

Given the location of the adversary, which is further from A than the legitimate node B is, we can conclude that M cannot send a message to A, pretending that the message came from B. This is because the attacker cannot show that he is at distance d' where B is located; the attacker cannot respond to the randomly generated challenge N_A such that the reply message reaches A within time which corresponds to d'. Such an attack would be possible only if the attacker would have a helper node located close to A, which would then create a radio wormhole, through which the attacker could fake its distance to A. Since A controls its integrity region and can visually verify the absence of such attacker nodes in its proximity, this attack will fail.

In order to fake its location, the attacker can attempt to send a "reply" message $\hat{R}_B$ to A ahead of time, anticipating the transmission of N_A. This attack is prevented by the use of commitments. The attacker is forced to commit to a message, or to wait until B sends the commitment. If the attacker tries to send an early reply $\hat{R}_B$ anticipating N_A, device A will extract the commitment opening value $\hat{o} = \hat{R}_B \oplus N_A$ which will not correspond to the commitment $\hat{c}$ as received by A, with a high probability (e.g., $1-O(2^{-k})$, $k = 128$). Early reply attacks will, therefore, be detected by A. Here, we note that the attacker's probability to successfully perform this attack lies in the probability that it guesses the nonce N_A and/or breaks the commitment scheme [32].

Finally, the attacker can modify transmitted messages by adding messages to the communication channel. These attacks are detected by commitments, as the attacker cannot modify messages such that $\hat{c}$ corresponds to $\hat{o}$.

3.3 Implementation

We implemented MT-IR protocol using Cricket mote platform [26], running TinyOS. Cricket mote is a Mica-based platform [1] enhanced with an ultrasonic channel.

In our implementation, we used a simple signal coding scheme for the transmission of messages over an ultrasonic channel (i.e. for the transmission of message r in the MT-IR protocol): each bit "1" was encoded as a presence of the ultrasonic signal of duration 200μs within a transmission window of 60ms, and bit "0" was transmitted as the absence of signal of the same duration within the same window (60ms) (the last 200μs are considered as the start of the bit). To avoid attacks on distance-bounding due to different speeds of radio and ultrasonic signals, the nonce N_A and the decommitment response R_B (Figure 4) are computed and exchanged bit-by-bit, as in the original Brands and Chaum distance-bounding protocol [8].

The commitment scheme was implemented using TinySec [16] message authentication code (MAC) implementation with Skipjack. The commit c to the message is therefore the output of the MAC function, and the decommit value o is the message itself and the secret key that was used to generate the commitment. In order to speed-up the execution of the protocol, in our implementation, only the key is sent over the ultrasonic channel, whereas the message is transferred using the radio channel.

Besides implementing our protocol, we further tested its

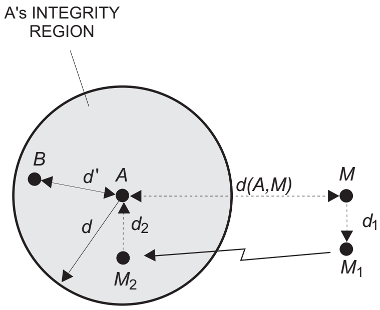


Figure 6: Wormhole attack on ultrasonic ranging.

resilience to range manipulations. Ultrasonic ranging techniques are vulnerable to distance enlargement and reduction attacks [34]. Distance enlargement consists in delaying ultrasonic signals that devices exchange, whereas distance reduction consists in creating radio wormholes that speed-up the transmission of the signals.

To prevent these attacks, in our protocol, we have assumed that the attacker cannot be located closer to the verifier A than the prover B (Fig. 5). If an attacker can place one malicious device close to A and another close to its own location, it will be able to perform a wormhole attack and pretend that it is in the integrity region of A. By doing this, the attacker would violate the MT-IR protocol and would be able violate the integrity and the authenticity of the messages exchanged between A and B. We implemented this attack using Cricket motes in a scenario shown on Fig. 6. Here, an attacker places node M_2 at distance $d_2 < d' < d$ from A and another node M_1 at distance d_1 from its own location.

Adversarial nodes M_1 and M_2 communicate mutually using a radio signal, through which they speed-up ultrasonic signals between M and A. In this way, M_1 and M_2 create an impression at A that M is closer to A than it actually is.

The reduction of distance between A and M by M_1 and M_2 is, however, bounded by the their distance to M and A. Since the distance from M_1 to M can be arbitrary small (both nodes are controlled by the attacker), the only limitation that these nodes have in performing the attack is in M_2's distance to A. This can be observed in our measurement results presented in Fig. 7, which show that the attackers can successfully reduce the distance between A and M up to the value $(d_1 + d_2)$, that is, the sum of distances between M_1 and M_2 to M and A, respectively.

These measurement results demonstrate the importance of the concept of integrity regions. They show that an appropriate control of the immediate vicinity of the devices can bring high security gain as the attackers which are located outside the integrity region cannot violate the integrity and the authenticity of the messages sent by devices that are located within the integrity region to the node controlling the region.

Another possible implementation of MT-IR is in using radio (RF) ranging. Radio ranging is, however, more demanding and it requires devices with a high (nanosecond) precision-of-time measurement. To the best of our knowledge, the only commercial technique that achieves such precision, and therefore achieves a high precision of distance measurement, is Ultra Wide Band (UWB) [11].

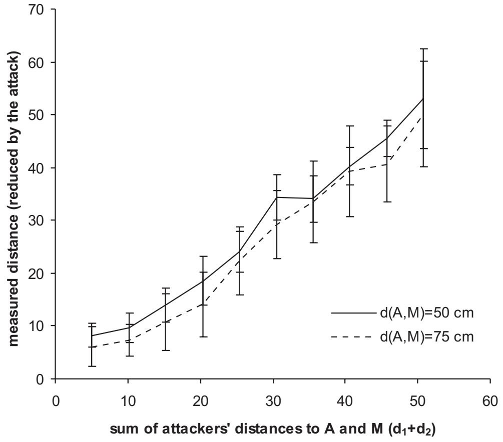


Figure 7: Wormhole attack on ultrasonic ranging. The attacker can reduce the distance to the device A only up to the sum of the distances d_1 and d_2.

With current off-the-self components, ultrasonic ranging seems the most viable implementation of MT-IR.

4. USER FRIENDLY KEY AGREEMENT WITH INTEGRITY REGIONS

In this section, we describe a key agreement protocol that allows two parties that share no secret (key) or certificates in advance to agree on a shared (secret) key. Our protocol is based on the Diffie-Hellman (DH) key agreement protocol (described in Section 2.2). As we already noted, the DH key agreement protocol is secure against passive attacks (under certain assumptions). Thus, in order to enable two parties A and B to securely agree on a shared secret using the DH protocol, it is sufficient to ensure the integrity of the DH public keys g^{X_A} and g^{X_B}. Traditionally, this is achieved by means of digital signatures and shared secrets. In this work, we show how to make use of the notion of integrity regions to preserve the integrity of the DH public keys in a flexible and user-friendly way (as opposed to some recent solutions that involve a physical contact between the parties' devices, exchange of pre-authentication data over an infrared link, cameras - see Section 5).

4.1 From Secure Message Transfer Authenticator to Secure Key Agreement

In Section 3.1, we presented the message transfer authentication based based on integrity regions (Fig. 4) that ensures that the message m accepted by Alice is the same message that has been sent by Bob.

In [6], Bellare, Canetti, and Krawczyk propose a very intuitive modular approach to security analysis and construction of secure protocols. This approach assumes two adversarial models: the authenticated link model (AM) and the un-authenticated links model (UM). Roughly speaking, the AM model is an ideal-world model in which the attacker is passive (e.g., he/she only eavesdrops communication). On the contrary, the UM model is a real-world model, in which

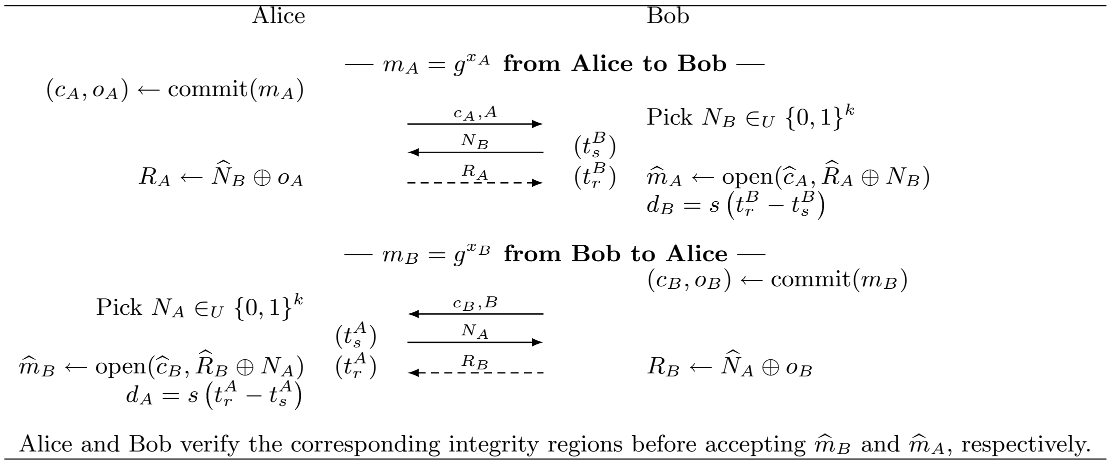

Alice
Bob
— $m_A = g^{x_A}$ from Alice to Bob —
$(c_A, o_A) \leftarrow \text{commit}(m_A)$
c_A, A
Pick $N_B \in_U \{0,1\}^k$
N_B
(t_s^B)
$R_A \leftarrow \widehat{N}_B \oplus o_A$
R_A
(t_r^B)
$\widehat{m}_A \leftarrow \text{open}(\widehat{c}_A, \widehat{R}_A \oplus N_B)$
$d_B = s\left(t_r^B - t_s^B\right)$
— $m_B = g^{x_B}$ from Bob to Alice —
$(c_B, o_B) \leftarrow \text{commit}(m_B)$
Pick $N_A \in_U \{0,1\}^k$
c_B, B
(t_s^A)
N_A
$\widehat{m}_B \leftarrow \text{open}(\widehat{c}_B, \widehat{R}_B \oplus N_A)$
(t_r^A)
R_B
$R_B \leftarrow \widehat{N}_A \oplus o_B$
$d_A = s\left(t_r^A - t_s^A\right)$
Alice and Bob verify the corresponding integrity regions before accepting $\widehat{m}_B$ and $\widehat{m}_A$, respectively.

Figure 8: DH-IR protocol: Straightforward application of the MT-IR authenticator to the basic DH key agreement protocol (the full and the dashed arcs represent radio and an ultrasonic channel, respectively).

the attacker is active; he can replay messages and try to fabricate messages. The security of the protocol is first proven in the AM model, assuming (as assumed by the model itself) that all the communication between the parties is authenticated. If the protocol is proven to be secure in the AM model, then it can be shown to be secure in the UM model, provided that each message transmitted between the parties is authenticated by a MT-authenticator [6].

The main idea of our approach to securing the basic DH key agreement protocol is to apply the MT-IR authenticator to the DH public keys g^{X_A} and g^{X_B}, as shown in Fig. 8. In other words, Alice and Bob will run the MT-IR protocol (Fig. 4) two times; that is, (1) by setting $m = g^{X_B}$ and running the protocol in Fig. 4, Alice will verify the integrity of g^{X_B}, and (2) by setting $m = g^{X_A}$ and running the protocol in Fig. 4 (where Alice and Bob reverse the roles), Bob will verify the integrity of g^{X_A}. We denote the resulting DH key agreement protocol ("compiled" with the MT-IR authenticator) with DH-IR (*Diffie-Hellman key agreement based on Integrity Regions*).

Then, based on [6], we have the following result for the DH-IR protocol:

PROPOSITION 1. *The DH-IR protocol, conditioned on the security of the MT-IR authenticator, is secure (i.e., a computationally bounded adversary cannot trick the parties into accepting modified DH public keys).*

While being secure, the DH-IR protocol is still somewhat suboptimal, as it requires two messages to be exchanged over an unreliable ultrasonic communication channel. Moreover, the size of de-commitment message o that is transmitted over the ultrasonic channel (see Fig. 8) increases over time, in order to compensate for fast (daily) advance in computational technology and computational power available to the adversary [20]; for example, the minimum required size for a target collision-resistant hash function increases linearly over time due to fast technological advances [20]. This is clearly not desirable in our context, since the reliability of the ultrasonic channel quickly deteriorates with the size of messages to be transmitted over it.

Consequently, we say that the DH-IR protocol is *time-variant*, that is, the number of bits to be transmitted using an unreliable ultrasonic communication channel increases over time. A similar problem has been reported recently by McCune, Perrig and Reiter in [23], where the resolution of a cellular phone screen was not sufficient to represent all the bits of a hash value. A simple solution to this problem is to increase permanently (due to increase in the hash function size) the resolution of displays. A better approach, however, would be to develop *time-invariant* protocols, where the size of messages to be transmitted over unreliable channels (or to be displayed on low-resolution cellular phone screens) would not vary over time.

In the following section, we present an optimized version of the DH-IR protocol, which involves only one message over the unreliable ultrasonic channel.

4.2 Optimal DH-IR Protocol

The optimized (and time-invariant) protocol unfolds as shown in Fig. 9. Both Alice and Bob calculate the commitment/openning pairs ((c_A, o_A) and (c_B, o_B)) for messages $m_A \leftarrow 0 \| g^{X_A} \| N_A$ and $m_B \leftarrow 1 \| g^{X_B} \| N_B$, respectively. Here, N_A and N_B are k bit long random strings and "0" and "1" are two public (and fixed) values that are used to break the symmetry and thus prevent a *reflection attack* [22].

In the first two messages, Alice and Bob exchange the commitments c_A and c_B. Then, in the following two messages they open the commitments by sending out o_A and o_B, respectively. It is important to stress that a given party opens his/her commitment only after having received the commitment value from the other party. The first four messages are exchanged over a radio link. Having received the commitment/openning pairs (c_A, o_A) and (c_B, o_B), Alice and Bob open the corresponding commitments and verify that "1" and "0" appear at the beginning of $\widehat{m}_B$ and $\widehat{m}_A$, respectively. If this verification is successful, Alice and Bob generate the authentication strings s_A and s_B, respectively, as shown in Fig. 9. Note that the length of each of these strings is k.

We will show later in this section that if there exist a means for Alice and Bob to verify, in a secure way, that $s_A = s_B$, then the probability that a computationally bounded adversary succeeds in forging the messages m_A and m_B can be made satisfactorily small. The main purpose of the last two

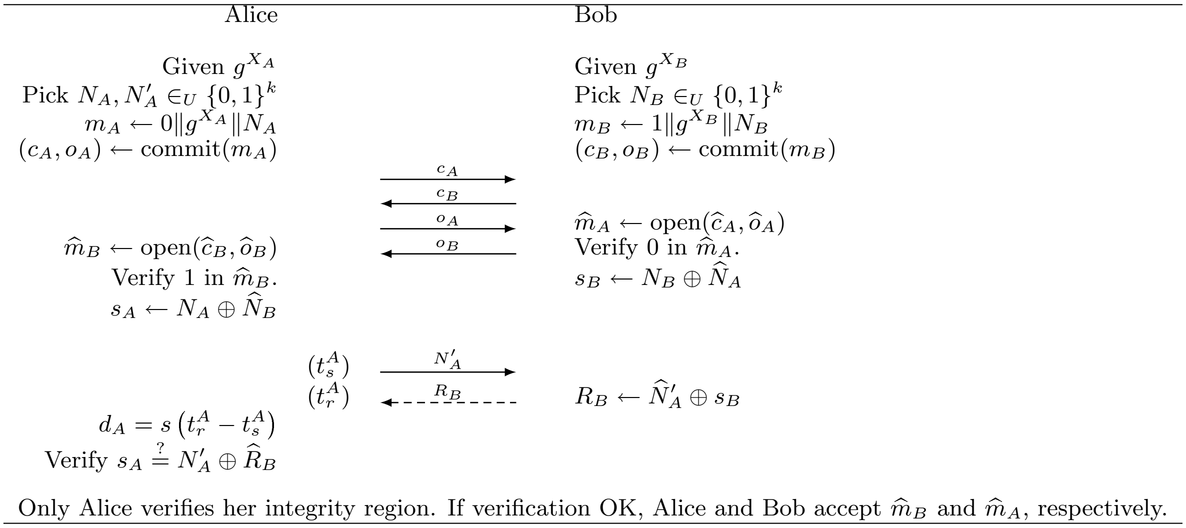

Alice
Bob
Given g^{X_A}
Pick $N_A, N'_A \in_U \{0,1\}^k$
$m_A \leftarrow 0 \| g^{X_A} \| N_A$
$(c_A, o_A) \leftarrow \text{commit}(m_A)$
Given g^{X_B}
Pick $N_B \in_U \{0,1\}^k$
$m_B \leftarrow 1 \| g^{X_B} \| N_B$
$(c_B, o_B) \leftarrow \text{commit}(m_B)$
c_A
c_B
o_A
o_B
$\widehat{m}_A \leftarrow \text{open}(\widehat{c}_A, \widehat{o}_A)$
Verify 0 in $\widehat{m}_A$.
$s_B \leftarrow N_B \oplus \widehat{N}_A$
$\widehat{m}_B \leftarrow \text{open}(\widehat{c}_B, \widehat{o}_B)$
Verify 1 in $\widehat{m}_B$.
$s_A \leftarrow N_A \oplus \widehat{N}_B$
(t_s^A)
N'_A
(t_r^A)
R_B
$R_B \leftarrow \widehat{N}'_A \oplus s_B$
$d_A = s\left(t_r^A - t_s^A\right)$
Verify $s_A \overset{?}{=} N'_A \oplus \widehat{R}_B$
Only Alice verifies her integrity region. If verification OK, Alice and Bob accept $\widehat{m}_B$ and $\widehat{m}_A$, respectively.

Figure 9: Optimal DH-IR key agreement protocol.

messages in the DH-IR protocol (Fig. 9) is to allow Alice to compare s_A against the authentication string s_B generated by Bob, in a secure way. Thus, Alice sends a k-bit long random string N'_A to Bob and measures the time until she received the response from Bob. Bob responds with $R_B \leftarrow \widehat{N}'_A \oplus s_B$, where the sign hat denotes that the N'_A as transmitted by Alice may have been altered by the adversary. Alice receives $\widehat{R}_B$, where again the sign hat denotes that R_B as transmitted by Bob may have been altered by the adversary. At the same time, Alice calculates the distance d_A as shown in Fig. 9 and verifies the corresponding integrity region for the presence of devices other than Bob's device (see Section 3.1). If this verification is successful, Alice knows that (with a high probability) Bob must have transmitted $\widehat{R}_B$, that is, $\widehat{R}_B = R_B$. Finally, if s_A equals $\widehat{R}_B \oplus N'_A$, Alice notifies Bob and they both accept the messages $\widehat{m}_A$ and $\widehat{m}_B$ (i.e., the corresponding DH public keys) as being authentic. Note that $\widehat{R}_B \oplus N'_A = s_B$ in case no attack takes place.

Optimality and Time-invariance. Compared to the DH-IR protocol from Fig. 8, the optimal DH-IR protocol involves only one k-bit long transmission over an unreliable ultrasonic communication channel. In addition, all the k bits to be communicated over the ultrasonic channel contribute to the uncertainty of the attacker to forge the DH public parameters. In this sense, the DH-IR protocol from Fig. 9 is optimal.

Concerning the time-invariance property, the number of bits k to be transmitted over the ultrasonic channel does not increase over time, even if the security parameters of the used commitment scheme commit$(\cdot)$ do increase. The commitment scheme is independent of the random strings N_A, N'_A and N_B. In this sense, the optimal DH-IR protocol from Fig. 9 is time-invariant.

Security of the Optimal DH-IR Protocol. It is out of the scope of this paper to give a full-fledged security proof for the optimal DH-IR protocol. We will rather give the main intuition behind different protocol stages. The optimal DH-IR protocol (Fig. 9) can be divided into two stages. The first stage finishes with the calculation of the authentication strings s_A and s_B. The second stage includes the remaining two messages and the corresponding verifications.

The main idea behind the first stage is the following. Assuming that Alice and Bob have a means to compare the authentication strings s_A and s_B in a secure way, then the odds of a computationally bounded adversary to forge m_A and m_B (and therefore g^{X_A} and g^{X_B}) can be made satisfactorily small. The second stage then simply implements the procedure to compare s_A and s_B in a secure way. We next give an informal proof of these two claims[3].

Claim 1. *Assuming that Alice and Bob can compare s_A and s_B in a secure way, the probability that a computationally bounded adversary forges m_A and/or m_B is bounded by $O(2^{-k})$.*

Proof Sketch. In this proof we will assume that the adversary cannot break the used commitment scheme commit$(\cdot)$; in practice this is possible only with a negligible probability. Let us first focus on the single run of the optimal DH-IR protocol. Assume that the adversary tries to submit either a forged message $\widehat{m}_B \leftarrow 1 \| g^{\widehat{X}_B} \| \widehat{N}_B$ to Alice or a forged message $\widehat{m}_A \leftarrow 0 \| g^{\widehat{X}_A} \| \widehat{N}_A$ to Bob. Note that this is the only way for the adversary to be successful against the observed protocol run; any attacking attempt against the commitment scheme will fail (by assumption, the commit$(\cdot)$ cannot be broken). We claim that in this case

$$Pr\big[s_A = s_B\big] = Pr\big[N_A \oplus \widehat{N}_B = N_B \oplus \widehat{N}_A\big] \leq 2^{-k} \ .$$

Indeed, from the unfolding of the optimal DH-IR protocol (Fig. 9), due to the binding, hiding and non-malleability properties of the commit$(\cdot)$, the adversary has to generate and commit to strings $\widehat{N}_B$ and $\widehat{N}_A$ before actually learning

[3] For simplicity, in this short analysis, we consider the commitment scheme to be perfect, with attacker's probability of breaking the scheme equal to $\epsilon \approx 0$. For an analysis of the impact of imperfect commitment schemes on the security of this authenticator, see [19].

messages m_A (i.e., N_A) and m_B (i.e., N_B), respectively. In other words, one among N_A and N_B will be disclosed as the last value in this protocol run (i.e., the adversary will learn at least one of them after having generated and committed to $\widehat{N}_A$ and $\widehat{N}_B$). Let us assume that it is the N_A. Since N_A is a random (uniformly distributed) k-bit string, we have

$$\begin{aligned} &Pr\big[N_A \oplus \widehat{N}_B = N_B \oplus \widehat{N}_A\big] = \\ &Pr\big[N_A = \widehat{N}_B \oplus N_B \oplus \widehat{N}_A\big] \leq 2^{-k} \ . \end{aligned}$$

Finally, for multiple runs of the protocol, the probability $Pr\big[s_A = s_B\big]$, assuming that the adversary is active, is bounded by $O(2^{-k})$.

Assuming that k is sufficiently large (i.e., 2^k is greater than the number of the protocol runs), $Pr\big[s_A = s_B\big] = 1$ will hold only when the adversary is not active (does not try to forge messages m_A and m_B). But this implies that the integrity of the DH public keys g^{X_A} and g^{X_B} will be preserved. □

Claim 2. *The probability that a computationally bounded adversary tricks Alice into accepting $\widehat{s}_B \neq s_B$, in the second stage of the optimal DH-IR protocol, is bounded by $O(2^{-k})$.*

Proof Sketch. Let us first consider a single run of the DH-IR protocol, or more precisely, a single run of its second stage. Referring back to Fig. 9, the goal of the adversary is to achieve that s_A is equal to $N'_A \oplus \widehat{R}_B$. Note that at this stage the adversary knows s_A. Since Alice verifies her integrity region for the presence of adversarial devices, the only hope for the adversary is to try to send an appropriate value $\widehat{R}_B$ from locations outside of Alice's integrity region. However, this implies that the adversary cannot wait for the random challenge N'_A before sending back $\widehat{R}_B$. Otherwise, Alice can detect, using the measured distance d_A, that $\widehat{R}_B$ does not come from her integrity region (see Section 3). Therefore, the only hope for the adversary is try to guess N'_A and set $\widehat{s}_B = s_A \oplus N'_{A,\text{guess}}$, where $N'_{A,\text{guess}}$ denotes the guess by the adversary. Then, we can write

$$Pr\big[s_A = N'_A \oplus s_A \oplus N'_{A,\text{guess}}\big] = Pr\big[N'_A = N'_{A,\text{guess}}\big] \leq 2^{-k} \ ,$$

since N'_A is a random (uniformly distributed) k-bit string.

Finally, for multiple runs of the protocol, the probability $Pr\big[N'_A = N'_{A,\text{guess}}\big]$, assuming that the adversary is active, is bounded by $O(2^{-k})$. □

By combining Claim 1 and Claim 2, we have the following result:

Proposition 2. *The probability that a computationally bounded adversary tricks Alice and/or Bob into accepting fake DH public keys $g^{\widehat{X}_B} \neq g^{X_B}$ and/or $g^{\widehat{X}_A} \neq g^{X_A}$ is bounded by $O(2^{-k})$.*

Therefore, by setting k to an appropriately large value, the odds of the adversary against the optimal DH-IR protocol can be made satisfactorily small.

5. RELATED WORK

Providing integrity and authentication over insecure (radio) channels is a very active area of research. In this context, Stajano and Anderson propose the *resurrecting duckling* security policy model, [31] and [30], in which key establishment is based on the physical contact between communicating parties (their PDAs). An approach inspired by the resurrecting duckling security policy model is proposed by Balfanz et al. [5]. 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). Asokan and Ginzboorg propose another solution based on a shared password [4]. They consider the problem of setting up a session key between a group of people (i.e., their computers) based on a fresh password chosen and shared among the parties present in the room; the shared password is then used to derive a strong shared session key. Users might be unreliable when dealing with meaningless strings, and have difficulties remembering strong passwords. To counter this problem, in [24], Perrig and Song suggest using hash visualization to improve the security of such systems.

In US patent no. 5,450,493 [21], Maher presents several methods to verify DH public parameters exchanged between users. This technique had a flaw, discovered by Jakobsson. Motivated by the flaw, Jakobsson and Larsson [18] proposed two solutions based on a temporary secret shared between the two users. In [12] and [13], 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 [32], Čagalj, Čapkun and Hubaux propose an optimal message authenticator, a more efficient protocol that enables provably secure authentication through the transfer of a short bit sequence over specifically constructed secure channels. In [19], Laur, Asokan and Nyberg show that such message authenticators require non-malleable commitment schemes.

In [9], Castelluccia and Mutaf propose an interesting device pairing protocol for CPU-constrained devices which relies on device indistinguishability; this protocol is an extension of the protocol proposed by Alpern and Schneider [3].

In [33], Čagalj et al. proposed Integrity 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 integrity codes enables a sender to encode any message such that if its integrity is violated in transmission, the receiver is able to detect it. This scheme enables broadcast authentication over insecure channels.

In [23] McCune, Perrig and Reiter present a system that utilizes barcodes and camera-telephones to implement a visual channel for authentication and demonstrative identification of devices. In [14], Goodrich et al. present a similar solution, based on voice channels.

6. CONCLUSION

In this paper, we introduced *integrity (I) regions*, a novel security primitive 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). Integrity regions make use of lightweight ranging techniques and of visual verification within a small physical space. The main application of integrity regions is in key establishment. The proposed scheme effectively enables *authentication through presence* and therefore protects key establishment from the man-in-the-middle attacks. We demonstrated that integrity regions can be efficiently implemented using off-the-shelf components such as ultrasonic ranging hardware.

7. ACKNOWLEDGMENTS

The authors acknowledge the help of Sahar Sedighpour and Kai Feng in the implementation of presented protocols.

8. REFERENCES

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

[2] Specification of the Bluetooth System (Core). Version 1.1. http://www.bluetooth.org, 2001.

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

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

[5] D. Balfanz, D. Smetters, P. Stewart, and H. Wong. Talking to Strangers: Authentication in Ad-Hoc Wireless Networks. In *Proceedings of NDSS*, 2002.

[6] M. Bellare, R. Canetti, and H. Krawczyk. A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols. In *Proceedings of the 30th Annual Symposium on the Theory of Computing*. ACM, 1998.

[7] C. Boyd and A. Mathuria. *Protocols for Authentication and Key Establishment*. Springer Verlag, 2003.

[8] S. Brands and D. Chaum. Distance-bounding protocols. In *Proceedings of the Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, pages 344–359. Springer-Verlag New York, Inc., 1994.

[9] C. Castelluccia and P. Mutaf. Shake Them Up! A movement-based pairing protocol for CPU-constrained devices. In *Proceedings of MobiSys*, 2005.

[10] W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 1976.

[11] R.J. Fontana. Experimental Results from an Ultra Wideband Precision Geolocation System. *Ultra-Wideband, Short-Pulse Electromagnetics*, 2000.

[12] C. Gehrmann, C.J. Mitchell, and K. Nyberg. Manual Authentication for Wireless Devices, 2004. RSA Cryptobytes, Vol. 7, No. 1.

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

[14] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. Loud And Clear: Human Verifiable Authentication Based on Audio. In *Proceedings of the 26th International Conference on Distributed Computing Systems (ICDCS)*, 2006.

[15] M. Jacobsson and S. Wetzel. Security Weaknesses in Bluetooth. In *Progress in Cryptography - CT-RSA 2001*. Lecture Notes in Computer Science 2020, Springer-Verlag, February 2001.

[16] Chris Karlof, Naveen Sastry, and David Wagner. Tinysec: A link layer security architecture for wireless sensor networks. In *(Proceedings of SenSys)*, November 2004.

[17] D. Kügler. Man in the Middle Attacks on Bluetooth. In *Proceedings of Financial Cryptography*, 2003.

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

[19] S. Laur, N. Asokan, and K. Nyberg. Efficient mutual data authentication using manually authenticated strings. Cryptology ePrint Archive, Report 2005/424.

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

[21] D.P. Maher. United States Patent (No. 5,450,493): Secure communication method and apparatus, 1993.

[22] Wenbo Mao. *Modern Cryptography, Theory & Practice*. Prentice Hall PTR, 2004.

[23] J. M. McCune, A. Perrig, and M. K. Reiter. Seeing is Believing: Using Camera Phones for Human-Verifiable Authentication. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2005.

[24] 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.

[25] D. Plummer. An Ethernet Address Resolution Protocol, 1982. IETF Standards Track RFC 826.

[26] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *Proceedings of MobiCom*, pages 32–43. ACM Press, 2000.

[27] Theodore S. Rappaport. *Wireless Communications: Principles & Practice*. Prentice Hall, February 2002.

[28] N. Sastry, U. Shankar, and D. Wagner. Secure Verification of Location claims. In *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, 2003.

[29] D. Song. dsniff. http://naughty.monkey.org/∼dugsong/dsniff/.

[30] 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.

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

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

[33] M. Čagalj, S. Čapkun, R. Rengaswamy, I. Tsigkogiannis, M. Srivastava, and J.-P. Hubaux. Integrity (I) codes: Message Integrity Protection and Authentication Over Insecure Channels. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2006.

[34] S. Čapkun and J.-P. Hubaux. Secure positioning of wireless devices with application to sensor networks. In *Proceedings of InfoCom*, 2005.

[35] B. Waters and E. Felten. Proving the Location of Tamper-Resistant Devices. Technical report, Princeton University.

[36] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of MobiHoc*, 2005.