Secure Ranging With Message Temporal Integrity

Nils Ole Tippenhauer, Kasper Bonne Rasmussen, and Srdjan Čapkun
Department of Computer Science, ETH Zurich
8092 Zurich, Switzerland
{tinils, kasperr, capkuns}@inf.ethz.ch

ABSTRACT

In this paper, we address the problem of delay attacks on radio frequency time of arrival (ToA) secure ranging. In secure ranging, two mutually trusted devices try to determine their distance in the presence of an attacker. A delay attack consists of delaying the ranging messages exchanged between the devices, resulting in an increase of the measured message arrival times and thus in an increase of the measured distance. In this work, we propose the first secure ranging protocol that enables the detection of delay attacks on ranging. This protocol therefore enables two trusted devices to obtain a secure estimate of their mutual distance; existing solutions enabled the devices only to obtain an upper bound on their mutual distance. We further discuss possible implementations of our secure ranging protocol using Ultra-Wide-Band radio technology. Finally, we introduce and formally define the notion of message temporal integrity, a message security property which relates to message delay and advancement.

1. INTRODUCTION

In this work, we address the problem of secure ranging between two trusted devices in time-of-arrival systems. Secure ranging enables devices to measure their mutual distance, such that the measured distance corresponds to their actual physical distance (within the ranging system accuracy) and is not affected by the attacker. This mutual distance can be used e.g. to compute one's correct position by using localization algorithms [17, 41]. This makes secure ranging relevant in systems and applications which critically rely on correct localization information and have no other conventional localization system such as GPS or WLAN localization; such conventional systems are also often susceptible to manipulation attacks [42, 35]. Examples of such systems include vehicular networks, sensor networks, and networks of mobile robots [13, 12, 4] which use locations for tasks such as navigation, routing, and data harvesting.

Over the last years, several protocols have emerged that solve different subproblems of secure ranging, depending on the trust between the participating parties. If the devices are not mutually trusting, devices assume different roles: one assumes the role of a verifier (trusted device) and the other device assumes the role of the prover (untrusted device). In this scenario, the verifier can have three security goals: to securely obtain the distance to the prover, an upper-bound on its distance to the prover or a lower-bound on its distance to the prover. In time-of-arrival systems, obtaining the actual distance to the prover or its lower-bound is inherently difficult since the prover can easily delay its ranging messages. However, it has been shown that it is possible for the verifier to obtain an upper-bound on its distance to the prover; the protocols that achieve this are commonly referred to as distance-bounding protocols [6, 7, 11, 32, 36, 25, 21, 40]. In the scenario in which the devices are mutually trusting, the devices again have the same three possible goals: to obtain a range, an upper-bound or a lower bound. In [41], a trusting distance bounding protocol is presented to obtain an upper-bound on the distance of the devices; this protocol is referred to as authenticated ranging protocol. Different secure ranging goals and scenarios are summarized in Table 1.

In this work, we address the following problem: *How can two mutually trusted devices obtain a correct estimate of their physical distance in the presence of an attacker?*

The solution that we will propose inherently also contains a solution to the distance lower-bounding problem. To the best of our knowledge, this is the first protocol that enables secure ranging (and not only distance bounding) of mutually trusted devices. Our protocol consists of two components: one based on randomized ranging messages, which prevents message advancement (distance reduction) attacks, and the other one based on special message encoding, which enables the detection of message delay (distance enlargement) attacks.

Following our secure ranging protocol, we introduce the notion of *Message Temporal Integrity* and formally define it as a message property which is preserved if the message is neither advanced nor delayed in its transmission over the communication channel.

The rest of the paper is organized as follows. In Section 2, we review some of the most relevant attacks on localization systems. In Section 3, we describe our secure ranging protocol, and analyze its security in 4. In Section 5, we define Message Temporal Integrity as a general notion and present the Message Temporal Integrity Protocol. Section 6 covers related work and we conclude the paper in Section 7.

Trust	Upper bound	Lower bound	Distance
Mutual	Authent. rang. [41]	Our proposal	
No trust	Distance bound. [6]	No known protocols	

Table 1: Secure ranging goals and protocols that achieve them. Distance bounding protocols yield an upper bound between untrusting devices and authenticated ranging protocols between mutually trusting devices. Until now, no protocols have been proposed which enable the measurement of a lower bound or of a secure distance between either trusting or untrusting devices. Our protocol will provide an upper and lower bound, and therefore a distance, for mutually trusting devices.

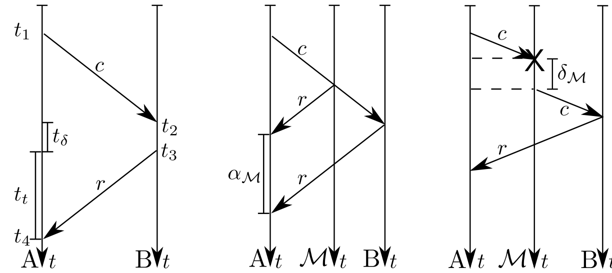
2. MESSAGE ADVANCEMENT AND DELAY ATTACKS

In this section, we discuss attacks whose aim is to advance or delay messages exchanged between two trusted devices and therefore to reduce or increase, respectively, the distance measured by these devices. We will show that current protocols can efficiently prevent message advancement attacks aiming to reduce distances, but do not prevent message delay attacks. In our analysis, we will therefore focus on the message delay attacks. We start by introducing our system and attacker model.

2.1 System Model

We address the following scenario: two (honest) entities A and B mutually trust each other and would like to determine their respective distance. Whenever we speak of the security of a given protocol, we implicitly assume that the entities involved in the protocol are not compromised and correctly follow the protocol. We do assume that the entities know public protocol parameters and either share a secret key or hold each others' authentic public keys. Finally, we assume that A and B do not know if they reside in each others' communication power range, i.e., if they can communicate *directly*.

To determine their mutual distance, A and B run a ranging protocol. A conventional non-secure ranging protocol, in which device A wants to measure the distance to the device B operates as follows (Figure 1(a)). To measure the distance, A starts a local clock at time t_1 and sends a challenge c to B; c contains the IDs of A and B, and the ranging command. Upon receipt of c, B starts a local high precision clock at t_2 and prepares a reply message r, which contains both IDs, the response command, and the the processing time $t_\delta = t_3 - t_2$ between receipt of c and transmission of r at t_3. When A gets r at t_4, it will stop its local clock to obtain the total runtime of the exchange $t_t = t_4 - t_1$. A is then able to compute the propagation time of one signal by subtracting the local processing time of B and dividing by two, $t_p = \frac{t_t - t_\delta}{2}$. Multiplying t_p with the propagation speed v (approximately speed of light) yields the distance between A and B: $d_{AB} = t_p v$.



(a) Time-of-Arrival ranging. (b) Advancement attack. (c) Delay attack.

Figure 1: (a) A simple ranging protocol: A sends c, B replies with r after t_δ. (b) The guessing-based advancement attack: The attacker $\mathcal{M}$ intercepts c sent from A to B. $\mathcal{M}$ now guesses r and replies earlier to A. The round trip time will be $\alpha_{\mathcal{M}}$ shorter than the expected propagation times. (c) The message delay attack: The attacker $\mathcal{M}$ intercepts a message (e.g. c sent from A to B) and replays it $\delta_{\mathcal{M}}$ time later. B's reply will then arrive later than expected at A.

2.2 Attacker Model

We adopt the following attacker model, which we will use throughout the paper. The goal of the attacker $\mathcal{M}$ is to delay or advance the delivery of messages. This attack can be useful in protocols in which the exact time of arrival is used to measure distances (e.g. secure ranging protocols) or to synchronize clocks. We assume that $\mathcal{M}$ cannot obtain the secret key shared between A and B. We do not specifically address side-channel information leaks in the analysis; trusted devices are assumed not to leak information. Finally, we consider the cryptographic primitives being used in this work to be secure, e.g. no attacks better than brute force exist to decrypt without the correct key, and the attacker is computationally bounded.

We assume that the attacker $\mathcal{M}$ controls the communication channel in the sense that she can eavesdrop messages and modify transmitted messages by adding her own messages to the channel. We further assume that the attacker cannot trivially disable the communication channel (e.g., by using a Faraday cage to block the propagation of radio signals) between A and B. The attacker can add signals to the channel (e.g., jam the transmission and in that way prevent the reception of the information contained in the original message). However, the receiver will still receive the message from the sender, superimposed by the attacker's signals. We assume that message relaying and insertion incurs a non-negligible processing delay δ for the attacker. As we will show in Section 4, the attacker is thus unable to reactively annihilate unpredictable signals, but can modify or annihilate the signals whose shape it can predict. In addition, we assume that the attacker cannot transmit messages at a speed higher than the speed of light.

2.3 Attacks on Distance Measurement Protocols

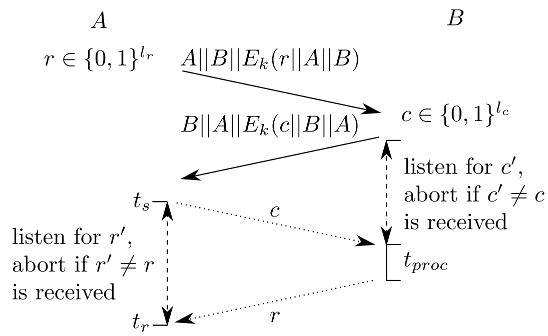
Message advancement attacks can be easily performed on the distance measurement protocol that is described in Section 2.1 if the reply message r is known to the attacker before the sender transmits it. If this is the case, in order to advance the message, the attacker simply needs to transmit it before the sender does (Figure 1(b)), and then jam the senders' message if the receiver is in the power range of the sender. If the sender and the receiver are not in each others' range, the attacker does not need to jam the sender's message since the message would never reach the destination. It might seem odd that the sender would try to send a message to the receiver if it does not know whether the receiver is in its range. However, the sender might believe that the receiver is in its range, while it is not, if the attacker created a wormhole between the two devices [15] and uses it to control the communication between the two devices. Another way to perform a message advancement attack is to speed up the transmission time of the message; this is not possible if the messages are transmitted using radio signals and travel at the speed of light.

The key to preventing message advancement attacks is to make both c and r at least partially unpredictable for the attacker. An example would be the inclusion of previously shared nonces N_c and N_r in both c and r for authentication. In such an authenticated scenario, the attacker has to guess the nonce to correctly advance either c or r, which she can do with a chance of $2^{-|N_c|}$ and $2^{-|N_r|}$, respectively. The authenticated ranging protocol in [41] is based on this approach.

Message delay attacks can be considered in two scenarios. In the first one, the communicating parties are in each others' range. In the second scenario, they are not (but e.g., believe that they are, as a result of a wormhole attack). The latter scenario is the most favorable for the attacker in the sense that she can simply record the message from one device and relay it with a delay to the other node. As we show later on, this attack can be prevented using secure ranging protocols, which will enable the devices to measure an upper-bound on their mutual distance and deduce if they are in each others' range. This attack as well as the notions of communication and physical neighborhood were previously discussed in [23]. The message delay attack is illustrated in Figure 1(c). To the best of our knowledge, except for the protocol that we propose in this work, there are no existing protocols that can successfully detect or prevent message delay attacks on ranging protocols.

3. SECURE RANGING

Until now, the goal of secure ranging protocols [6, 41] was to provide a secure upper distance bound between the mutually trusted verifier and prover. This is achieved by measuring the round trip time of challenges sent by one of the nodes and the corresponding replies by the other node. These protocols prevent an attacker from reducing the measured distance by relying on authentication of the ranging challenges and replies, but not from increasing the distance using message delays. In this section, we will present a Delay Evident Secure Ranging (DESR) protocol, which allows the devices to measure securely both the lower- and the upper-bound of their mutual distance. This protocol therefore enables the devices to measure securely their physical distances. This protocol can serve as a basis for secure localization and can equally be adapted for precise secure time synchronization (see Section 5).



$$d_{AB} = \frac{t_r - t_s - t_{proc}}{2} v$$

Figure 2: The Delay Evident Secure Ranging protocol: in the initialization phase, A sends a fresh nonce r to B, encrypted using a shared key k. B replies with an encrypted fresh nonce c. To transmit the subsequent messages c and r, A and B use our special modulation scheme as explained in Section 3.2. In the ranging phase, B starts listening on the channel, recording the first (even incomplete) message c' it receives. A sends c and starts itself listening for any reply r'. Upon reception of $c' = c$, B sends the reply r after a constant delay t_{proc}. If A receives $r' = r$, it computes the distance d_{AB} to B. The protocol was successful iff $c' = c$, $r' = r$, and $d_{AB} \leq$ the communication range of A and B.

Our protocol is shown in Figure 2, it consists of two phases: the initialization phase and the measurement phase.

3.1 Initialization Phase

During the initialization phase, device A contacts device B in order to initiate secure ranging. Here, we assume that A and B share a secret key, which they use to authenticate and encrypt their communication when needed.

Using this key, A generates and encrypts a nonce r of length l_r and sends it to B. B replies to A with a fresh nonce c of length l_c, again encrypted using k. Both nonces are used to ensure message freshness in the following ranging phase.

3.2 Ranging Phase

All communication in the ranging phase is based on a special modulation, which will now be described in detail (see Figure 3). The modulation scheme resembles traditional On-off keying with single very short UWB pulses (e.g. 2ns pulse length and 10^6 symbols per second on devices like [33]). To send a binary sequence, for every One, a pulse is sent, and for every Zero no pulse is sent. To detect the message, the receiver will integrate signals in the communication frequency band over a time window with length equal to the pulse rate. If the total energy in this integration window is above a certain threshold, the signal is decoded as one, otherwise as zero. This threshold depends only on the expected signal strength of the UWB pulses used to communicate and not on the noise level on the channel. On-Off keying schemes are commonly being used in existing Ultra-Wide-Band ranging devices, which also provide the nanosecond precision, that is required by RF-ranging protocols [33]. To protect against delay attacks on our system, the receiver of a message does not only attempt to detect correct packets on the channel, but also incomplete messages, single pulses and any signals within its operating frequency band, regardless of their shape. For this, the receiver of a message in the ranging phase will listen for the message continuously, until either (1) the expected message was successfully received, or (2) any other sequence of pulses was detected.

We will now explain the ranging phase step-by-step:

1. In the initialization phase, A sends its initial message to B. B responds to A, and starts listening on the channel immediately afterwards, awaiting the challenge c from A.

2. A sends c using our modulation scheme and starts a local high precision clock.

3. Immediately after sending c, A also starts to listen on the channel itself for r.

4. B receives c'

5. If $c' = c$, B replies with r after t_{proc}, otherwise B aborts the protocol.

6. A receives r' and stops its local clock.

7. If $r' = r$, A computes the round trip time passed between the transmission of c and the reception of r. Based on this time, the distance between A and B can be computed. If this distance is smaller than the maximal (conservative) communication range of A and B, both parties consider the distance measurement successful.

8. In any other case, A and B consider this distance measurement as failed.

In detail, the distance between both parties can be computed as $d_{AB} = \frac{t_r - t_s - t_{proc}}{2} \cdot v$, where t_s and t_r are the sending time of c and the reception time of r, respectively; t_{proc} is the constant or known processing time at B and v is the constant propagation time of radio signals (approximately the speed of light). Only if the measured distance d_{AB} is less than the maximum communication range of both parties, will they accept the result, otherwise they will either restart the protocol or assume the presence of an attacker. If the ranging phase was aborted during its execution, both parties can agree to repeat it using a new challenge c and reply r, or simply assume the presence of an attacker and abort the protocol.

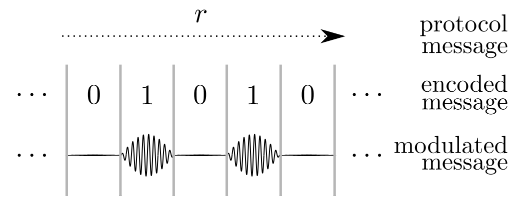


Figure 3: The challenges and replies in the secure ranging protocol are transmitted using a variant of on-off keying.

4. SECURITY ANALYSIS

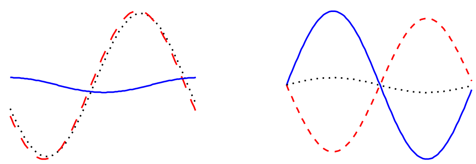
In this section, we analyze the security of our protocol. In particular, we will discuss attacks aiming to delay the challenge or response on the channel. We will show that message advancement attacks on our protocol are prevented using cryptographic primitives and that message delay attacks are effectively prevented due to the message encoding and modulation used to transmit protocol messages. Some of these attacks were already discussed in other contexts [39, 1, 2]. We will also show how the prerequisite of the communicating partner's proximity (presence in each others' range) is verified in our protocol.

This analysis uses the attacker model as defined in Section 2.2. In particular, the goal of the attacker is to shorten or lengthen the measured distance between A and B.

4.1 Message Advancement

We will now show how we can ensure that neither the challenge nor the reply message can be advanced by the attacker. Given that A and B are mutually trusted, the messages have not been sent too early by the devices. To advance the challenge or reply on the channel, the attacker could try to guess the content of the messages, and send the message earlier. Therefore, advancing either the challenge or the reply by guessing can be done with probability $2^{-|c|}$ and $2^{-|r|}$, respectively. In the case the message was guessed correctly, we assume that the erasure of the following original message is automatically successful for the attacker, if this should be necessary.

Another way of advancing a message on the communication channel would be to use a faster communication channel and use a wormhole to get the message to the receiver faster than intended. This attack was demonstrated on ultrasonic ranging systems using a radio wormhole [29]. This attack is not possible in our scheme, since we assume that the devices use radio frequency technology to communicate. As radio waves propagate with a constant speed which is close to the speed of light v, the attacker cannot transmit the messages



(a) Overshadowing (b) Destructive interference

Figure 4: (a) Overshadowing of two sine waves: the original signal (solid blue) is overshadowed by another signal (dashed red) with different phase. At the receiver, the original message looks like noise on a strong signal (dotted black). (b) Destructive interference of two sine waves: the original signal (solid blue) is superposed by an echo signal (dashed red), shifted by π radians and 90% of its amplitude. The resulting received signal is attenuated to 10% of the original signal (dotted black).

faster.

4.2 Message Delay

We will now show how we can ensure that the c and r have not been delayed on the channel on the way from the sender to the receiver. This analysis is valid only if A and B are in each other's communication range (see Section 4.3); if they are not, the protocol is aborted. This proximity forces the attacker to prevent the reception of the original message at the receiver in order to replay it later for a delay attack.

We will now discuss different methods to prevent the reception of the original message, and conclude with a discussion of message manipulation attacks.

4.2.1 Effects of Noisy Jamming and Overshadowing

A simple attack to prevent a receiver from hearing a message on the channel is to send a second, stronger signal. This stronger signal will add to the legitimate signal from the original sender and overshadow it (Figure 4(a)). The content of the added signal can range from random noise in the target band to fully modulated and coded messages. Typical radio hardware will drop malformed messages and ignore messages which are not intended for that recipient, higher level protocols will therefore not be able to detect this attack but will decode the message contained in the stronger signal.

In our protocol, using the proposed on/off keying based modulation scheme, an attack using high levels of noise or other signals can be trivially detected. This follows from the observation that such a jamming attack will lead to constant high signal values, which will be decoded as ones at the receiver. Therefore, the data content of the message is changed, which will be detected. This forces the attacker to perform more subtle jamming attacks, such as signal annihilation, which will be discussed in Section 4.2.2. For this reason, we will only consider these more advanced jamming attacks in the remainder of the paper.

The same is true for naïve overshadowing attacks, in which the attacker simply replays the original message with a small delay and at a higher gain is shown in Figure 5. In our mod-

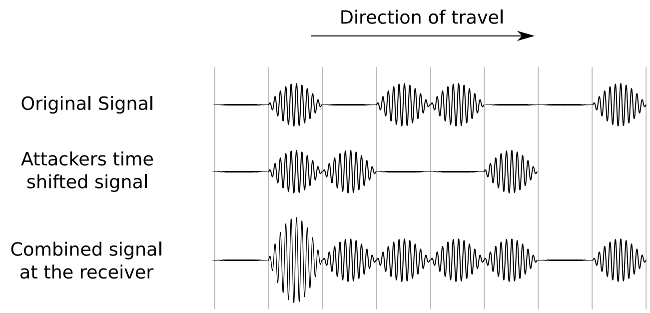


Figure 5: Naive overshadowing attacks: The upper signals are 10011010 encoded using On-Off Keying. An attacker trying to replay this message while it is still being transmitted will cause a collision, yielding an invalid message.

ulation scheme, this attempt will result in a decoding error at the receiver. Therefore, the attacker has to erase the signals for the original message first in order to replay the original message, just like in the jamming case as discussed above.

4.2.2 Signal Annihilation

A more subtle attack is to prevent the reception of the original message is to construct an annihilation (canceling) signal based on destructive signal interference. This is a well-known problem in communications and it sometimes occurs unintentionally in multipath environments [24]: objects in the environment of the sender or receiver reflect radio waves, which in effect add as correlated noise to the received signal. Due to the increased propagation path, a phase shift of the original signal occurs. In some cases, this will lead to a serious signal degradation at the receiver, e.g. if the phase shift is around 180 degrees or π radians, the reflected signal will superimpose the original signal and significantly attenuate the strength of the original signal (Figure 4(b)).

Following the same principle, the attacker can artificially create such interference to attenuate or erase signals from the channel at the receiver's location. In our security model, we consider a strong attacker who is always able to erase signals from the channel, if she knows or can predict these signals in advance. Nevertheless, in the case of our DESR protocol, both c and r are nonces unknown to the attacker. We will now discuss in more detail attacks attempting to erase unknown messages from the channel, and why we consider them infeasible.

To erase a message of unknown content, in addition to matching the signal baseband's frequency and phase, most important the destructive signal sent by the attacker has to match the carrier's frequency and phase. This can be done in two ways: by analyzing the received signal and sending out an own countersignal (demodulating attacker), or by simply delaying and re-sending the received signal (repeating attacker). We will now discuss each of these and then discuss the impact of the attacker's position.

Demodulating attacker: An attacker who demodulates the signal will try to obtain the current symbol's information

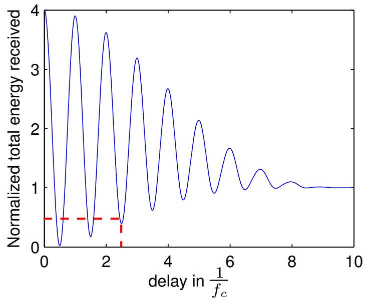


Figure 6: Remaining energy at the receiver in a destructive interference attack: if the original signal is replayed with the correct delay, the signal at the receiver is attenuated. To achieve an attenuation to less than 50% of the original received squared signal strength, the delay can be maximal a quarter of the symbol duration, which is equivalent to 2.5 carrier frequency periods in this example.

before it is completely transmitted, and then start a corresponding counter signal to prevent the remaining part from arriving at the receiver. Assuming that the attacker's goal is to attenuate the signal by at least 50%, she will have at most half of the symbol's duration to demodulate the incoming signal and start her own transmission. This assumes that the attacker is positioned colinear with both sender and receiver. We will discuss the influence of non-optimal positioning later.

Repeating attacker: Another attack is the following: the attacker $\mathcal{M}$ picks up all signals in the target band of the channel and replays them with a certain delay. In order to maximise the attenuation, this delay can be either half the wavelength λ_c of the carrier, or $(n+0.5)\lambda_c, n \in \mathbb{N}$, if the attacker is positioned colinear with the sender and the receiver. If the attacker is not colinear with the sender and the receiver, the additional propagation delay of the cancelling signal has to be taken into account. We have simulated such signal cancellations and found that the maximum delay available to $\mathcal{M}$ in this scenario is approximately a quarter of the symbol length d_p. This result can be obtained by computing the total energy of the original and cancelling signal with respect to the phase shift, Figure 6 shows an example of such a result for a baseband wavelength $\lambda_b = 2000$ and a carrier wavelength $\lambda_c = 100$. In this case, a maximal phase shift of 25% symbol length will still attenuate the total signal power to 40%, any more delay will leave more than 50% of the energy on the channel.

Attacker's Position: In the case that $\mathcal{M}$ is not colinear with A and B, the signals by the attacker will be delayed additionally . This delay is directly reducing the attacker's time to react to a signal on the channel. For both demodulating and replaying attackers, the following inequality must hold:

$$\frac{d_{A\mathcal{M}} + d_{\mathcal{M}B}}{v} + \delta \leq \frac{d_{AB}}{v} + \delta_{max}$$

In this inequality, v is the signal propagation speed, d_{AB} is the distance between A and B, and δ is the attacker's processing time to react to the signal. The maximal processing delay depends on the target attenuation and the attacker type, e.g. 50% attenuation by a demodulating attacker yields $\delta_{max} = \frac{d_p}{2}$, while for a repeating attacker $\delta_{max} \approx \frac{d_p}{4}$ (d_p is the length of the symbol).

Even if we assume an ideal attacker with $\delta = 0$, the attacker is still restricted to an area very close to the signal path. For example, if $d_p = 2ns$, then the following has to be true for the attacker's location:

$$d_{A\mathcal{M}} + d_{\mathcal{M}B} \leq d_{AB} + 15\text{cm}$$

Summary on message erasure: We conclude from our analysis that the erasure of signals predictable for the attacker is possible using the methods discussed. Nevertheless, these attacks are much more difficult if the signals cannot be predicted reliably, especially if only short symbols are used in the communication. Therefore, to be able to attack our secure ranging system, the attacker has to guess the content of c or r in advance in order to successfully erase them, with the chance of a successful guess as discussed in Section 4.1. Only then, the attacker will be able to match the legitimate signal's carrier frequency and phase.

4.3 Communication Proximity Verification

As mentioned in Section 4.2, in order to make sure that the signals between them were not delayed, the devices need to be able to verify if they reside in each others' direct communication range. Only in this scenario unsuccessful message erasures at the destination will be detected at the receiver. We will now explain why the attacker cannot convince the devices that they are in direct communication range, if they are not. If both parties are not in direct communication range and the signals are instead forwarded by $\mathcal{M}$, she will be able to delay the communication, and thus enlarge the measured distance. However, the attacker is not able to reduce the measured distance. It therefore suffices that the devices compare their measured distance to their nominal communications range. Note that the devices are mutually trusted and can thus share information about their radio antennas etc., based on which they can estimate their mutual communications range. Only if the result of the range measurement shows that both devices can directly communicate, the result of the protocol will be used, and then the parties will be sure that the measured distance is their exact distance, and was not enlarged by the attacker. Notice that the protection against distance shortening attack (message advancement) does not rely on the devices being in each others' ranges.

4.4 Replay Attacks

Our protocol protects B from replay attacks by $\mathcal{M}$ by relying on the freshness of c and r. If $\mathcal{M}$ replays A's initialization message, B will respond with an encrypted nonce, which $\mathcal{M}$ cannot decipher. Therefore, $\mathcal{M}$ cannot send the correct c to initiate the ranging. Likewise, $\mathcal{M}$ cannot impersonate B by replaying a previous response to the initialization message, because $\mathcal{M}$ does not know the current r to reply in the ranging phase.

4.5 Discussion on Modulation and Other Physical Layer Attacks

The modulation scheme we proposed in Section 3.2 was chosen because it resembles UWB based on-off keying already used in the context of insecure range measurements [33]. Therefore, our proposed changes do not require fundamental changes to the radio hardware. We expect that other modulation schemes can be found which would also be suited for a secure ranging system. The only basic requirement for such a modulation scheme is the prevention of full message erasure by the attacker. In particular, the area of jamming resistant communication promises to provide such solutions.

Several of the attacks proposed on wireless distance measurements so far assume an untrusted, potentially malicious prover. In [7], guessing-based early-reply attacks are discussed in which the malicious prover tries to reply prematurely to challenges by the verifier, or starts sending the preamble before the complete challenge has been received. Both attacks are not applicable in our scenario, in which we trust the prover to behave correctly. In addition, our use of very short UWB signals mitigates the impact of single symbol advancement attacks by a third party. As the typical gain of these attacks is a fraction of the symbol length, in our case this attack would have a maximal gain in the order of a few nanoseconds.

4.6 Analysis Summary

As detailed in Section 2, the only chance for an attacker to delay a message unknown to the attacker is to erase it from the channel using destructive interference and to replay it later. According to our earlier analysis in Section 4.2.2, this erasure will only be successful with a probability of 2^{-l_c} and 2^{-l_r}, respectively, because the whole message has to be guessed correctly to erase it. Furthermore, if the attacker plans to only modify the content of a message unpredictable to her in advance, e.g. by using signal superposition, this problem is as hard as message erasure (which is, in fact, a modification of all Ones in the original message to Zero).

In summary, we can conclude that the attacker has at best a chance of $2^{-\min(l_c,l_r)}$ to manipulate the outcome of a single round of ranging.

5. MESSAGE TEMPORAL INTEGRITY

Unlike data communication systems, localization and ranging systems are focused on time measurements, not on data exchange. Ranging systems base their operation primarily on the measurements of the propagation of signals (i.e., on signal time-of-flight). To operate in adversarial settings, localization and ranging protocols therefore require that both the data integrity and physical characteristics such as the temporal integrity of their protocol messages (signals) are protected [16, 41]. Although the notion of time is important and can be crucial in a number of other, more traditional security protocols (e.g., authentication and key establishment [5]), what is particular to localization and ranging protocols, is the requirement that the attacker cannot manipulate the message reception time. Whereas many security protocols can tolerate arbitrary message delays, and are proven to be secure against such attacks under the Dolev-Yao model [8], ranging protocols cannot tolerate message delays. If the attacker introduces a one nanosecond delay in the delivery of a message, this delay will change the measured distance by 30 cm. Due to the importance of device locations when nanosecond message delay attacks are considered, the detection of these attacks requires a more complex approach than the detection of longer delays ($\geq \mu$seconds).

In this section we generalize the solution proposed in Section 3 and define the notion of *Message Temporal Integrity*. We say that the temporal integrity of the message exchanged between two parties is preserved, if the message is neither advanced nor delayed in its transmission over the communication channel. Existing research has addressed message temporal integrity only partially. In recent years a number of protocols were proposed that only protect the communication against message advancement attacks [15, 28, 29, 41]; however, there are no solutions in the open literature that fully protect the communication from message delay attacks. The solutions proposed so far enable detection of message delay attacks only if those delays are long (these solutions were discussed in the context of time synchronization in [10, 19, 34]). However, until now no solutions exist that enable the detection of delay attacks if those delays are in the order of nanoseconds.

5.1 Message Temporal Integrity

We define Message Temporal Integrity more precisely as follows.

DEFINITION 1. *The temporal integrity of a message m sent at time t_s is entirely preserved iff its propagation time $t'_p = t'_r - t_s$ from the sender to the receiver is equal, within a specified accuracy, to the propagation time $t_p = t_r - t_s$ of the same message, if the message would be sent at the same time t_s, and would propagate from the sender to the receiver unaffected by the adversary. Here, t_r is the time at which the message arrives at the receiver.*

This definition states that if the message temporal integrity is preserved, the message propagated on the channel unaffected by the attacker; i.e., the attacker did neither advance nor delay the message. Following Definition 1, we further define two additional notions: Upper-Bound and Lower-Bound Message Temporal Integrity. These notions are defined in the same way as Message Temporal Integrity, except that the condition that $t'_p = t_p$ in Definition 1 is modified, in the case of Upper-Bound Message Temporal Integrity to $t'_p \geq t_p$, and in the case of Lower-Bound Temporal Integrity to $t'_p \leq t_p$.

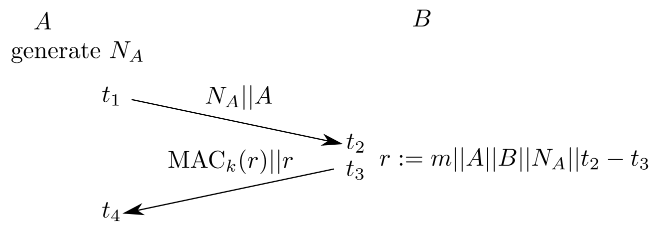
This simply means that the Upper-Bound Message Temporal Integrity is preserved if the message is not advanced, but could have been delayed. Similarly, the Lower-Bound Message Temporal Integrity is preserved if the message is not delayed but could have been advanced. Examples of protocols that achieve Upper-Bound Temporal Integrity are existing secure ranging [41, 3] and distance-bounding protocols [6, 7, 11, 22, 32, 36, 25, 21, 40], which prevent the attacker from advancing the message by simply making the message unpredictable for the attacker. However, in those protocols the attacker can delay the message once it has been transmitted.

The verification of the message temporal integrity assumes that the communicating devices are able to accurately measure propagation times of the messages that they exchange. However, these propagation times are very short, usually in the order of nanoseconds (e.g., if the devices are 100m distant, the propagation time of the message between the sender and the receiver will be approx. 330 ns). Today, a number of devices, mainly those designed for ranging and localization applications, are able to measure message (signal) arrival times with such precision; examples include devices that use Ultra-Wide-Band radios [33]. However, in a number of other platforms (e.g., those based on 802.11 standards), the applications do not have access to precise signal acquisition times, and they cannot measure the message propagation times. Instead, these devices will be only able to measure the time interval that passed from the time at which the operating system handed the message to the sender's radio, till the time at which the receiver's radio passed the message to the receiver's operating system. We call this time interval the message transmission time and we denote it by t_t. For 802.11 based platforms, t_t is usually in the order of microseconds.

Given this, we refine our definition of message temporal integrity and we introduce a notion of a *Loose and Tight Message Temporal Integrity*, respectively. We say that the message temporal integrity is tight, if it refers to the message propagation time t_p. We say that the message temporal integrity is loose, if it refers to the message transmission time t_t, which consists of t_p and other contributing factors such as the medium based access and message transmission times. Tight and loose upper and lower bound message integrity is defined analogously.

5.2 Preserving Loose Message Temporal Integrity

In this section we show how the devices can, using a simple challenge-response protocol, verify the loose temporal integrity of their messages. This can be achieved by a challenge-response protocol that we show in Figure 7; this protocol is a variant of the protocol proposed in the context of secure time synchronization in [10, 19, 34]. The protocol enables the device A to verify the loose temporal integrity of the message m sent from another entity B. This is achieved by the following: (1) message advancement attacks are prevented by making parts of the exchanged messages unpredictable to the attacker; and (2) message delay attacks are detected by comparing the measured round-trip transmission time t'_t with the expected round-trip transmission time t_t.



$t'_t = \frac{(t_2-t_1)+(t_4-t_3)}{2}$

If $t'_t \leq t_t$ then *loose temporal integrity was preserved*,
else *abort*

Figure 7: The Loose Message Temporal Integrity Protocol: A requests a loose temporal integrity verifiable message from B by sending a nonce N_A and its name. B then replies with a MAC-secured message containing the data m, the participating parties, the nonce N_A, and the processing time $t_3 - t_2$. A can now verify if the transmission time t'_t was as expected t_t with μs precision.

We assume that before the protocol execution, the parties share a secret key k. The protocol is started by A, which transmits a freshly generated nonce; this nonce is unpredictable for the attacker. Upon receiving the nonce, B computes a message authentication code (MAC) over this nonce and over the message that it intends to send, using the key k, and transmits the computed MAC, along with the message back to A. Upon reception of the reply from B, A measures the round-trip message transmission time $(t_2-t_1)+(t_4-t_3)$, as a difference between the time at which it received the reply from B and the time at which it transmitted N_A. Based on this time, A computes the measured message transmission time t'_t. Finally, to verify the temporal integrity of m, A verifies the received MAC, and compares the measured transmission time t'_t with the expected (maximal) message transmission time t_t. Here, t_t can be generously defined, to allow for the measurement variability of t'_t. If $t'_t \leq t_t$, A concludes that the temporal integrity of m was not violated.

5.3 Problem Statement: Preserving Tight Message Temporal Integrity

In the previous section, we showed that, using a simple protocol, devices can easily verify the loose temporal integrity of the messages that they exchange. In this section, we show why a similar approach cannot be used to verify the tight message temporal integrity.

Figure 7 shows that the loose temporal integrity of messages can be easily verified when the message transmission time can be measured or estimated. The same protocols can be used to verify tight message temporal integrity, assuming that the expected message propagation time t_p is known.

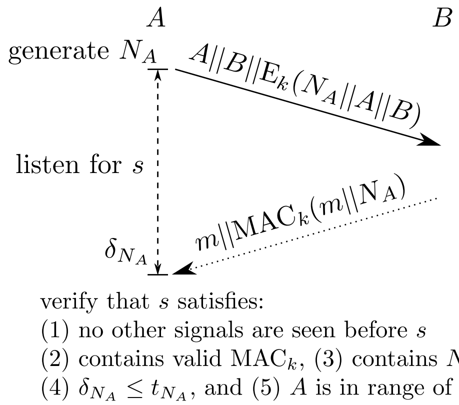


Figure 8: The Message Temporal Integrity Protocol: A requests m, the message to be secured, by sending an encrypted nonce N_A to B. After sending this request, A starts to listen to any signal s on the channel. Upon reception of the nonce, B replies with a message constructed using N_A, m and k. When A receives a message s, she verifies the temporal integrity by checking the listed 5 conditions. In this check, t_{N_A} is the maximum round trip time for the message exchange, and δ_{N_A} the actual round trip time measured by A.

However, unlike in the case of message transmission time t_t, which can be estimated independently of the distance between the nodes, the expected message propagation time t_p cannot be estimated without knowing the distance between the nodes. The reason for this is simple: the processing delays in the radios can make t_t at least an order of magnitude larger than the propagation time. The distance between the nodes d_{AB}, which is directly proportional to the propagation time ($d_{AB} = v \cdot t'_p$, where v is the speed of light), therefore, for most considered distances (e.g., ≤ 100m) does not matter in the measurement of t_t and t'_t. However, d_{AB} is crucial in the verification of the tight temporal integrity since it directly determines the value of t'_p.

This is why the LMTI protocol in Figure 7 cannot be used to verify the tight temporal integrity of the messages unless the distance between the communicating parties is known. However, in most scenarios, the tight temporal integrity of the messages is verified in order to secure ranging and localization, where the goal of the application is to measure the distance between the nodes or their locations. The LMTI protocol therefore cannot be used in such applications.

In the remainder of this paper, we propose a protocol to verify the tight temporal integrity for messages between parties that do not know their mutual distance.

5.4 MTI Protocol Description

In this section, we propose our message temporal integrity protocol (MTIP). Using this protocol, the communicating parties can verify that all correctly received messages were neither advanced nor delayed by the attacker. We first present our protocol; we then analyze its security in detail.

The goal of our message temporal integrity protocol (MTIP) is to verify the tight temporal integrity of a message m sent from B to A. The protocol unfolds as follows (Figure 8): A requests a message m from B by transmitting a nonce N_A encrypted with a shared key k. Upon sending the nonce, A actively listens on the channel for a predefined time t_{N_A}. If within that time, A does not receive the reply from B, it aborts the protocol (or restarts it with a fresh nonce).

Upon receiving the nonce, B sends back a reply containing the same nonce N_A, the message m and a message authentication code (MAC) constructed using k to protect message data integrity. Like the challenges and replies of the secure ranging protocol (DESR), this reply is specially encoded on the physical layer using on-off keying.

Upon reception of this reply, A verifies that (1) during t_{N_A} and prior to the arrival of the message, there were no other signals (messages) on the channel, decodeable or garbled (2) the MAC corresponds to the message and to N_A,(3) the transmission contains N_A which corresponds with this protocol run, (4) the transmission was received before t_{N_A} elapsed, and (5) A knows that it is in the power range of B. If all five conditions are met, A concludes that the tight temporal integrity of m was not violated.

The main intuition why the temporal integrity of the message can be verified is the same as in the secure ranging case (Section 4), and will be discussed in more detail in the following security analysis.

The MTIP works under the assumption of presence awareness, that is, that A will accept that B's message preserved its temporal integrity only if it knows that during the protocol execution, A was in B's power range. This condition can be validated right before the MTIP is run, by having A and B run a secure ranging protocol.

5.5 Security Analysis

We will now show that the proposed MTIP enables the verification of the tight message temporal integrity (upper and lower). In this analysis, we will use the attacker model defined in Section 2.2, and discuss lower and upper bound temporal integrity separately.

5.5.1 *Upper Bound Message Temporal Integrity*

Upper bound message temporal integrity guarantees that the messages have not been advanced on the communication channel. In order to advance a message, the attacker $\mathcal{M}$ needs to know its content before it will be sent, which would enable her to send it earlier. In the context of our protocol, this means that the attacker needs to guess either the nonce N_A that A will send, and/or the $\mathrm{MAC}_k(m, N_A)$ that B transmits to A (we assume that the message m is known to the attacker). Only then, the attacker would be able to violate the upper-bound MTI of the message m; however, if the nonce and the MAC are sufficiently long, the attacker can guess their content only with negligible probability. Given that the message from B to A is authenticated, this assures A that the message was indeed sent by B, and because A

trusts B, it trusts that the message was sent at the expected time.

5.5.2 Lower Bound Message Temporal Integrity

We already argued about the protection against message delay in the case of unpredictable messages in Section 4. Messages with predictable content are assumed to be easily delayable by the attacker in our attacker model. Unfortunately, even the prepending of predictable data to unpredictable data can weaken the security against delay attacks. Consider the following example: a nonce N_A is appended to a message m of content known to the attacker a priori, and the message is longer than the nonce (i.e., $|m| > |N_A|$). The receiver, however, has no way to verify the integrity of m — it could be any data. This could be the case, for example, in a monitoring sensor network, in which a special sensor reading is caused inevitably by the attacker, who then wants to delay the reporting of this message. To delay the reporting message, the attacker can delete the message from the channel, let the nonce pass unchanged and then transmit any data she wants followed by a replay of the nonce. In that case the message will appear in the following sequence to the receiver: $a||b||N_A$ where a is the unchanged original nonce, b is the data the attacker inserted and N_A is the replayed nonce. Unless the receiver can detect that the message has changed, it will now believe that the temporal integrity of the message was not violated since the nonce is correct, when in fact the attacker delayed the transmission by up to the length of the original message $|m|$. This attack can be prevented in two ways: one is by transmitting the nonce before the message. In this case, the message data integrity is still not preserved, but its temporal integrity is. The second possibility is to authenticate the message using a message authentication code. We chose to use the latter case in our MTIP, and will analyze it now in more detail.

To prevent the above mentioned attack, we protect the message integrity by appending a MAC. This MAC is computed by using the key k, shared between A and B, and cannot be created for arbitrary messages by the attacker. Therefore, the data content of the shifted message must have the same data content of the original message, so that the attacker can re-use the original MAC. The attacker can still easily erase the data part m from the channel, but hiding the original MAC in the data section would require the computation of a new MAC for the changed data section – which is considered infeasible (as it requires guessing the MAC with a chance of $2^{-|\text{MAC}|}$). The other alternative is the attempt to change the MAC into the corresponding bits of data. But as the attacker does not know the MAC in advance, she essentially has to guess it in order to flip the right bits, with the same chances of success as before. We therefore conclude that the attackers chances are $2^{-|\text{MAC}|}$ to delay or erase the original message if a MAC is used, even if the content of m is known to the attacker in advance.

6. RELATED WORK

The problem of message temporal integrity is closely related to the problem of (secure) time synchronization, which has been studied in detail in the context of wireless networks [10, 9, 38, 19, 20, 26, 31, 34].

Related research has also been done in the context of protocols that upper bound the message round-trip time in ranging applications. Upper bounding of the round-trip time also gives an upper bound on the distance between the nodes — this family of protocols are therefore known as distance bounding protocols [6, 7, 11, 22, 32, 36, 25, 21, 40]. However, these protocols allow only to verify an upper bound to an untrusted prover, but do not address distance enlargement attacks. Distance bounding protocols represent a subset of secure localization protocols [43, 28, 30, 14, 27, 18, 37]. Distance bounding protocols are attractive because they do not need dedicated positioning hardware but instead rely on nano-second precision of the radio to localize.

To ensure message temporal integrity with nano-second granularity in this work, we make use of the properties of on-off keying. On-off keying was first put in the context of secure communication in [39] where it was used for broadcast authentication. In this work, we make use of on-off keying to achieve temporal integrity by making sure that an external attacker cannot erase legitimate transmissions from the channel and therefore is unable to violate the temporal integrity of the transmitted messages.

To the best of our knowledge, there are no solutions in the open literature that propose a ranging protocol that enables the detection of message delay (and thus also distance enlargement) attacks. One solution that detects range enlargement attacks was proposed in [41] in the context of secure localization; however, that solution assumes that the location of a device (and thus the distances to the device) are verified by an infrastructure of at least three verifiers.

7. CONCLUSION

In this paper, we discussed the problem of message advancement and delay attacks on wireless communication channels. Starting with well-known secure ranging protocols, we showed how to achieve protection not only against message advancement attacks but also against message delay attacks. The resulting new secure ranging protocol enables not only secure computation of an upper bound on the distance between two trusted parties but also the secure computation of the actual distance between the devices.

We further introduced notions of Loose and Tight *Message Temporal Integrity*, new message properties that define message temporal manipulations. We proposed and analyzed protocols that achieve message temporal integrity.

Our proposed UWB modulation scheme is well suited for implementation on existing and upcoming Ultra-Wide-Band communication platforms; this constitutes a part of our future work.

8. REFERENCES

[1] David Adamy. *EW 101: A First Course in Electronic Warfare*. Artech House, February 2001.

[2] David Adamy. *EW 102: A Second Course in Electronic Warfare*. Artech House, August 2004.

[3] A. Alkassar and C. Stuble. Towards secure IFF: preventing mafia fraud attacks. In *Proceedings of MILCOM*, 2002.

[4] Ljubica Blazevic, Jean-Yves Le Boudec, and Silvia Giordano. A location-based routing method for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 4(2), 2005.

[5] Colin Boyd and Anish Mathuria. Protocols for authentication and key establishment, 2003.

[6] Stefan Brands and David Chaum. Distance-bounding protocols. In *Proceedings of EUROCRYPT*, 1994.

[7] Jolyon Clulow, Gerhard P. Hancke, Markus G. Kuhn, and Tyler Moore. So near and yet so far: Distance-bounding attacks in wireless networks. In *Proceedings of ESAS*, 2006.

[8] D Dolev and A.C. Yao. On the security of public key protocols. In *Proceedings of the IEEE 22nd Annual Symposium on Foundations of Computer Science*, 1981.

[9] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. *SIGOPS Operating System Review*, 2002.

[10] Saurabh Ganeriwal, Srdjan Čapkun, Chih-Chieh Han, and Mani B. Srivastava. Secure time synchronization service for sensor networks. In *WiSe '05: Proceedings of the 4th ACM workshop on Wireless security*, 2005.

[11] Gerhard P. Hancke and Markus G. Kuhn. An RFID Distance Bounding Protocol. In *Proceedings of IEEE SecureComm*, 2005.

[12] M. Hazas and A. Ward. A novel broadband ultrasonic location system. In *Proceedings of Ubicomp*, September 2002.

[13] J. Hightower and G. Borriello. A survey and taxonomy of location systems for ubiquitous computing, 2001.

[14] Lingxuan Hu and David Evans. Localization for mobile sensor networks. In *Proceedings of ACM/IEEE MobiCom*, September 2004.

[15] Yoh-Chun Hu, Adrian Perrig, and David B. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *Proceedings of IEEE InfoCom*, 2003.

[16] Loukas Lazos and Radha Poovendran. Serloc: secure range-independent localization for wireless sensor networks. In *Proceedings of ACM WiSe*, 2004.

[17] Loukas Lazos, Radha Poovendran, and Srdjan Čapkun. Rope: robust position estimation in wireless sensor networks. In *Proceedings of IPSN*, 2005.

[18] Donggang Liu, Peng Ning, An Liu, Cliff Wang, and Wenliang Du. Attack-resistant location estimation in wireless sensor networks. In *ACM Transactions on Information and System Security*, 2008.

[19] M. Manzo, T. Roosta, and S. Sastry. Time synchronization attacks in sensor networks. In *Proc. of the ACM Workshop on Security of Ad Hoc and Sensor Networks*, 2005.

[20] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi. The flooding time synchronization protocol. In *Proceedings of ACM SenSys*, 2004.

[21] Catherine Meadows, Paul Syverson, and LiWu Chang. Towards more efficient distance bounding protocols for use in sensor networks. In *Proceedings of IEEE SecureComm*, 2006.

[22] Jorge Munilla, Andres Ortiz, and Alberto Peinado. Distance bounding protocols with void-challenges for RFID. Printed handout at RFIDSec, July 2006.

[23] Panos Papadimitratos, Marcin Poturalski, Patrick Schaller, Pascal Lafourcade, David Basin, Srdjan Čapkun, and Jean-Pierre Hubaux. Secure neighborhood discovery: A fundamental element for mobile ad hoc networking. *IEEE Communications Magazine*, 2008.

[24] Theodore Rappaport. *Wireless Communications: Principles and Practice*. 2001.

[25] Kasper Bonne Rasmussen and Srdjan Čapkun. Location privacy of distance bounding protocols. In *Proceedings of ACM CCS*, 2008.

[26] Kasper Bonne Rasmussen, Srdjan Čapkun, and Mario Čagalj. Secnav: secure broadcast localization and time synchronization in wireless networks. In *Proceedings of ACM/IEEE MobiCom*, 2007.

[27] T. Ristenpart, G. Maganis, A. Krishnamurthy, and T. Kohno. Privacy-preserving location tracking of lost or stolen devices: Cryptographic techniques and replacing trusted third parties with DHTs. In *Proceedings of Usenix Security*, July 2008.

[28] Naveen Sastry, Umesh Shankar, and David Wagner. Secure verification of location claims. In *Proceedings of ACM WiSe*, 2003.

[29] Sarah Sedihpour, Srdjan Čapkun, Saurabh Ganeriwal, and Mani Srivastava. Implementation of Attacks on Ultrasonic Ranging Systems, demo at ACM SENSYS'05, 2005.

[30] Vitaly Shmatikov and Ming-Hsiu Wang. Secure verification of location claims with simultaneous distance modification. In *In Proc. of 12th Annual Asian Computing Science Conference (ASIAN)*, December 2007.

[31] M. Sichitiu and C. Veerarittiphan. Simple, accurate time synchronization for wireless sensor networks. In *Proc. of the IEEE Wireless Communications and Networking Conference*, 2003.

[32] Dave Singelée and Bart Preneel. Distance Bounding in Noisy Environments. In *Proceedings of ESAS*, 2007.

[33] Multispectral Solutions. UPS (Urban positioning system).
Multispectral Solutions, Inc; *www.multispectral.com*.

[34] Kun Sun, Peng Ning, and Cliff Wang. Tinysersync: secure and resilient time synchronization in wireless sensor networks. In *Proceedings of ACM CCS*, 2006.

[35] Nils Ole Tippenhauer, Kasper Bonne Rasmussen, Christina Pöpper, and Srdjan Čapkun. Attacks on public wlan-based positioning. In *Proceedings of ACM MobiSys*, 2009.

[36] Nils Ole Tippenhauer and Srdjan Čapkun. Id-based secure distance bounding and localization. In *Proceedings of the European Symposium on Research in Computer Security*, 2009.

[37] Patrick Traynor, Joshua Schiffman, Thomas La Porta, Patrick McDaniel, Abhrajit Ghosh, and Farooq Anjum. Constructing secure localization systems with adjustable granularity. Technical Report NAS-TR-0084-2007, December 2007.

[38] J. van Greunen and J. Rabaey. Lightweight time synchronization for sensor networks. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, 2003.

[39] Mario Čagalj, Srdjan Čapkun, Ramkumar Rengaswamy, Ilias Tsigkogiannis, Mani Srivastava, and Jean-Pierre Hubaux. Integrity (i) codes: Message integrity protection and authentication over insecure channels. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, 2006.

[40] Srdjan Čapkun, Leventé Buttyan, and Jean-Pierre Hubaux. Sector: Secure tracking of node encounters in multi-hop wireless networks. In *Proceedings of ACM SASN*, 2003.

[41] Srdjan Čapkun and Jean-Pierre Hubaux. Secure positioning in wireless networks. *IEEE Journal on Selected Areas in Communications*, 2006.

[42] J. S. Warner and R. G. Johnston. Think GPS cargo tracking = high security? think again. *Technical report, Los Alamos National Laboratory*, 2003.

[43] Brent R. Waters and Edward W. Felten. Secure, private proofs of location. Technical Report TR-667-03, January 2003.