UNIVERSITÀ DI PISA

Scuola di Dottorato in Ingegneria "Leonardo da Vinci"

Corso di Dottorato di Ricerca in
INGEGNERIA DELL'INFORMAZIONE

Tesi di Dottorato di Ricerca

Secure and Private Localization in Wireless Networks

Autore:

Pericle Perazzo _____

Relatore:

Prof. Gianluca Dini _____

Anno 2014
SSD ING-INF/05

Sommario

Nell'era del mobile computing, la posizione di persone o cose è un'informazione importante per una vasta gamma di applicazioni. Questo tipo di dato è peculiare sotto molti aspetti, e pone nuove sfide dal punto di vista della sicurezza e della privacy. Riguardo alla sicurezza, la diffusa tecnologia GPS si è rivelata piuttosto fragile. Gli attacchi di spoofing della posizione sono facili da mettere in pratica contro ricevitori di segnale GPS civile. Un attaccante relativamente sofisticato può ingannare un ricevitore e portarlo a misurare qualsiasi posizione desiderata. Questo può avere effetti devastanti su sistemi "dependable" basati su misure GPS.

D'altra parte, una posizione è spesso riferita ad una persona. In questo caso, è un dato personale, e la sua divulgazione indiscriminata può costituire una violazione della privacy. Inoltre, la posizione di una persona può rivelare informazioni ancora più sensibili su di essa. L'enorme raccolta di posizioni di utenti da parte degli odierni fornitori di servizi sta diventando una seria preoccupazione. L'opinione pubblica si sta sensibilizzando sempre di più su questo problema. È facile immaginare che i fornitori di servizi del futuro dovranno essere fidati dal punto di vista della privacy. Le compagnie che non sono fidate dai loro stessi clienti incontreranno serie difficoltà sul mercato.

In questa tesi di dottorato, consideriamo una serie di problemi di sicurezza e di privacy riguardanti ai dati di posizione, e proponiamo soluzioni innovative. Prima di tutto, affrontiamo il problema della misura sicura (non GPS) di posizioni. Studiamo le limitazioni delle attuali tecnologie range-based di posizionamento sicuro, che utilizzano *protocolli distance-bounding* eseguiti da nodi àncora terrestri. Un protocollo distance-bounding permette di misurare un limite superiore sicuro alla distanza tra due dispositivi. Ci concentriamo su un nuovo tipo di attacco, generalmente considerato fattibile in letteratura: l'attacco di *enlargement*. Gli attacchi di enlargement mirano a far misurare al protocollo distance-bounding una distanza maggiore di quella reale. Investighiamo la loro fattibilità ed il loro effetto contro protolli distance-bounding implementati su standard IEEE 802.15.4a UWB. Basandoci sui risultati di questa analisi, proponiamo *EMCD-ML*, un algoritmo per il posizionamento sicuro che riduce sensibilmente il

numero di nodi àncora necessari rispetto ai metodi allo stato dell'arte.

In secondo luogo, affrontiamo il problema della protezione della privacy nella disseminazione e nell'uso dei dati di posizione. Proponiamo **-UniLO*, un insieme di operatori che offuscano delle posizioni per scopi di privacy. Gli operatori *-UniLO sono applicati dall'utente al dato di posizione prima di rilasciarlo al service provider. Essi impediscono al provider di inferire altre informazioni sensibili dal dato di posizione, e contemporaneamente mantengono la fruibilità del servizio. Affrontiamo anche i problemi correlati di: (i) gestire l'imprecisione nelle misure di posizione; (ii) offrire più livelli di privacy contemporanei; (iii) difendersi contro avversari conoscenti la mappa; e (iv) difendersi contro location server non fidati. Gli operatori *-UniLO offrono un livello di sicurezza più alto rispetto ai metodi di offuscamento allo stato dell'arte.

Abstract

Since computing has become mobile, the position of people or things is an important piece of information for a plethora of applications. This kind of data is peculiar in several ways, and poses new challenges from the point of view of security and privacy. Regarding security, the broadly-used GPS technology has shown to be quite fragile. Position spoofing attacks are easy to mount against civilian GPS receivers. A relatively sophisticate attacker can deceive a receiver into measuring any desired position. This can have devastating effects on "dependable" systems based on GPS measurements.

On the other hand, position is often referred to people. In this case, it is personal data, and its indiscriminate disclosure can constitute a privacy violation. Moreover, the position of a person can disclose even more sensitive information about her. The huge collection of users' positions by nowadays service providers is becoming a major concern. Public opinion is getting more and more aware of this problem. It is easily imaginable that future service providers will have to be trusted from a privacy standpoint. Companies which are not trusted by their own costumers will have a hard time on the market.

In this dissertation we consider a range of security and privacy problems related to position data, and we propose novel solutions to them. First of all, we approach the problem of secure (GPS-free) measurement of positions. We study the limits of current secure positioning technologies based on ground anchor nodes, in particular those based on the standard IEEE 802.15.4a UWB. We investigate some relevant security properties of such a standard, relative to the feasibility and the effect of *enlargement attacks*. Basing on the results of this analysis, we propose novel approaches for secure positioning able to significantly improve the scalability in terms of anchor nodes.

Secondly, we approach the problem of privacy preserving in the dissemination and the usage of position data. We propose **-UniLO*, a set of operators that obfuscate positions for privacy purposes. *-UniLO operators can be applied to position data by the user before its dissemination to the service providers. They do not permit the provider to infer other sensitive information from position data, while still permitting

the delivery of the service. We also approach the related problems of: (i) dealing with imprecise location measurements; (ii) offering multiple contemporaneous levels of privacy; (iii) defending against map-aware adversaries; and (iv) defending against untrusted location servers. *-UniLO operators offer a higher security level with respect to state-of-the-art obfuscation methods.

To my wife,
she helped me a lot

1

Introduction

Positions are everywhere in modern computing. The possible usages of position information are countless: from self-guidance of unmanned vehicles [31] to location-based services [11, 35, 39], from location-based access control [7] to tracking of people or goods [26]. The security and privacy problems posed by this kind of data are peculiar. In this dissertation we approach some of them. Namely, we focus on the topics of *secure positioning* [20, 62, 88, 100, 101] and *location privacy* [8, 37, 44, 48, 68].

Secure positioning aims at measuring the position of a device in presence of an adversary trying to deceive the measurement process. We focus on range-based secure positioning techniques [20, 100, 108], which are based on distance-bounding protocols. Distance bounding [15, 16, 50] allows us to determine a secure upper bound to the distance between two devices. We first approach the sub-problem of *enlargement attacks*. Enlargement attacks aim at deceiving the distance bounding into measuring a distance larger than the real one. They can follow a jam-and-replay strategy or an overshadow strategy. We propose *SecDEv*, a distance bounding protocol able to withstand jam-and-replay strategies. Then, we study the feasibility of overshadow strategies against distance bounding implemented on IEEE 802.15.4a UWB [55]. Basing on the results of this analysis, we propose *EMCD-ML*, a method for secure positioning based on the impossibility of the adversary to control the effect of an overshadow attack. EMCD-ML sensibly reduces the necessary number of anchor nodes with respect to state-of-the-art methods.

Location privacy aims at avoiding the disclosure of (precise) position data in location-based services (*LBSs*). Privacy-preserving mechanisms can be various and orthogonal to each other. We propose *LbSprint*, a software architecture to integrate different privacy-preserving mechanisms by means of the standard language XACML [73]. Then, we develop *UniLO*, a mathematical operator for location privacy. UniLO reduces the precision of a position before its disclosure, in such a way that an adversary cannot reconstruct original data. We also extend it to provide for multiple contemporaneous levels of privacy. We show that UniLO surpasses state-of-the-art obfuscation methods in terms of resistance against statistical attacks, while still per-

mitting the delivery of the service. Finally, we develop some advanced techniques that further improve the resistance of UniLO in case of untrusted location servers, map-aware adversaries and imprecise position measurements.

1.1 Structure of the dissertation

This dissertation is divided in a first part about distance bounding and secure positioning, and a second part about location privacy. It follows a brief description of each chapter.

1.1.1 SecDEv: secure distance evaluation in wireless networks

The contribution of this chapter is twofold. First, we propose SecDEv, a secure distance-bounding protocol for wireless channels that withstands enlargement attacks based on jam-and-replay. By leveraging on the characteristics of radio frequency signals, SecDEv establishes a *security horizon* within which a distance is correctly measured and a jam-and-replay attack is detected. Second, we show how SecDEv improves the scalability of secure positioning techniques. This chapter has been published as a conference paper [32].

1.1.2 Feasibility of overshadow enlargement attack on IEEE 802.15.4a distance bounding

In this chapter we analyze the feasibility of enlargement attacks through overshadow strategies on 802.15.4a UWB distance-bounding protocols. We show that the overshadow strategies, generally considered feasible by the existing literature, are actually difficult to carry out. Depending on the delay introduced by the adversary, there are cases in which they have no effect or their effect is not controllable. This chapter has been published as a journal paper [94].

1.1.3 EMCD-ML: secure positioning through enlargement miscontrol detection

In this chapter we propose EMCD-ML (Enlargement MisControl Detection MultiLateration), a secure positioning algorithm which leverages on the adversary's impossibility of controlling enlargement attacks. EMCD-ML guarantees low adversarial success probability, while improving the scalability in terms of anchor nodes, with respect to state-of-the-art solutions.

1.1.4 Integration of privacy protection mechanisms in location-based services

The contribution of this chapter is twofold. First we present LbSprint, a middleware architecture for location-based services which integrates different privacy mechanisms

by means of the standard XACML language. The system administrator can configure and extend the set of such mechanisms. Secondly, we present practical optimizations which considerably improves the performance of the XACML policy evaluation process. This chapter has been published as a conference paper [34].

1.1.5 UniLO: a uniformity-based approach to location privacy

In this chapter, we propose UniLO, an obfuscation operator which offers high assurances on obfuscation uniformity, even in case of imprecise location measurement. We also deal with service differentiation by proposing three UniLO-based obfuscation algorithms that offer multiple contemporaneous levels of privacy. Finally, we experimentally prove the superiority of the proposed algorithms compared to the state-of-the-art solutions, both in terms of utility and resistance against inference attacks. This chapter has been partially published as a conference paper [33].

1.1.6 Advanced techniques for obfuscation-based location privacy

In this chapter we present an advanced obfuscation approach capable of dealing with measurement imprecision, multiple levels of privacy, untrusted servers, and adversarial knowledge of the map. We estimate its resistance against statistical-based deobfuscation attacks, and we improve it by means of three advanced techniques, namely *extreme vectors*, *enlarge-and-scale*, and *hybrid vectors*.

2

SecDEv: secure distance evaluation in wireless networks

Nowadays, many critical systems depend on position measurements. Some examples include robot guidance, geographic routing, etc. The security of these systems is at risk if someone can cause the system to measure a false position (*spoofing attack*). Assuring security in the measurement of a position is not a trivial challenge. *Secure positioning systems* [19, 62, 88, 100, 101] aim at correctly determining a position in presence of a spoofer adversary. We focus on *range-based* secure positioning, which acts by directly measuring distances (*ranging*) from a set of anchor nodes whose positions are known. A promising approach in this sense is to measure distances by means of *wireless distance-bounding protocols* [15, 16, 50, 81]. A distance-bounding protocol determines a distance between a verifier and a prover by measuring the round-trip time between a request and an acknowledgment messages, both carrying cryptographic quantities. They are usually realized on impulse-radio ultra-wide band (IR-UWB) technology, which is capable of sub-meter precision in ranging operations. Distance-bounding protocols are designed to resist to *reduction attacks*, i.e., an adversary is not capable of causing the measurement of a shorter distance. On the other hand, *enlargement attacks* are generally considered feasible by the literature. As a result, current range-based secure positioning methods need additional tests and a large number of anchor nodes [19, 100].

In this chapter we propose SECure Distance EValuation (SecDEv), a distance-bounding protocol able to resist to enlargement attacks based on jam-and-replay tactics [58, 96, 100]. SecDEv exploits the characteristics of wireless signals to establish a *security horizon* within which a distance can be correctly evaluated (besides measurement errors) and any adversarial attempt to play a jam-and-replay attack is detected. We also show how SecDEv improves the scalability of secure positioning techniques in terms of number of anchor nodes. This chapter has been published as a conference paper [32].

The remainder of this chapter is organized as follows. In Section 2.1 we compare SecDEv with other state-of-the-art solutions. In Section 2.2 we introduce a reference distance-bounding protocol. In Section 2.3 we define the threat model. In Section 2.4

we introduce SecDEv as an improvement of the reference distance bounding. In Section 2.5 we show how SecDEv improves the performance of secure positioning techniques.

2.1 Comparison of SecDEv to the state of the art

Secure localization has a vast applicability in many technological scenarios, but it has showed to be a nontrivial problem. The silver bullet is yet to be found.

Brands and Chaum [15] proposed distance-bounding protocols, in which a *verifier* node measures the distance of a *prover* node. Distance-bounding protocols do not determine the actual distance, but rather a secure upper bound on it. In this way, the actual distance is assured to be shorter or equal to the measured one, even in presence of an adversary. These protocols were created to assure the physical proximity between two devices, and consequently to contrast *mafia fraud* attack [30].

Hancke and Kuhn [50] fitted distance bounding protocols for RFId tags. Their proposal deals with a variety of practical problems such scarce resources availability, channel noise and untrusted external clock source. Though extensions for RFId's are possible, we focus on more resourceful devices. We assume the clock source is internal and trusted and the channel noise is corrected by FEC techniques.

Clulow et al. [22] focused on a wide variety of low-level attacks, which leverage on packet latencies (e.g. preambles, trailers, etc.) and symbols' modulations. PHY-layer preambles are sent before the cryptographic quantities, in order to permit the receiver to synchronize itself to the sender's clock. The preamble of the response is fixed and does not depend on the content of the challenge. A dishonest prover could thus anticipate the transmission of the response preamble to reduce the measured distance. To deal with this problem, Rasmussen and Čapkun [83] proposed full-duplex distance bounding protocols, in which the challenge and the response are transmitted on separate channels. The prover receives the challenge and meanwhile transmits the response. In this way, a dishonest prover cannot anticipate the transmission of the response, without having to guess the payload. In the present chapter, we assume the prover to be honest. This permits us to simplify our reference distance-bounding protocol (cfr. Section 2.2). In particular we use a single channel in a half-duplex fashion.

Flury et al. [41] and, more in depth, Poturalski et al. [81] analyze the PHY-protocol attacks against impulse-radio ultra-wideband ranging protocols (IR-UWB), with particular attention to 802.15.4a [85], which is the *de facto* standard. These studies concentrate only on reduction attacks, and estimate their effectiveness in terms of meters of distance reduction. We instead focus on the opposite problem, distance enlargement, which requires different countermeasures.

Chiang et al. [19] proposed the first technique able to mitigate the enlargement attack in case of dishonest prover. The verifier makes two power measurements of the prover's signal on two collinear antennas. Subsequently, it computes the difference

of the two measurements. Given the standard path-loss model, if the difference is low, the signal source will be far away. Otherwise it will be near. The idea is that the adversary cannot modify the way the signal attenuates over the distance, thus the distance estimation is trusted. Obviously such proposal relies on the standard path-loss model, which is poorly reliable. The authors claim that if the path loss exponent varies between 2 and 4, an enlargement of more than twice the measured distance is impossible. In this chapter, we focus on external adversaries. The problem of distance enlargement in presence of internal ones is challenging as well, but falls outside our present scope.

2.2 Reference distance-bounding protocol

A distance-bounding protocol allows a *verifier* (V) to "measure" the distance of a *prover* (P). In its basic form, a distance-bounding protocol consists in a sequence of single-bit challenge-response rounds [15]. In each round, the verifier sends a challenge bit to the prover that replies immediately with a response bit. The round-trip time enables V to compute an upper-bound of the P distance. Then, the distance is averaged on all rounds. Many variants of distance-bounding protocols have been proposed in the literature [16, 50]. Here, we establish a *reference distance-bounding protocol*, similar to those described in [81] for external adversaries. It involves a *request* message (REQ) from the verifier, an *acknowledgment* message (ACK) from the prover, and a final *signature* message (SGN) from the prover. Such a reference protocol is vulnerable to jam-and-replay attacks, as we will show in Section 2.3, and SecDEv (cfr. Section 2.4) will overcome these vulnerabilities.

The request and the acknowledgement convey, respectively, a and b, which are two independent, random and unpredictable sequences of bits. Note that, differently from the original version of distance-bounding protocol, the request and the acknowledgement are frames, rather than single bits. In fact, it is hard to transmit single bits over an IR-UWB channel. This is due to TLC regulation, which poses strict limits to the transmission power. In 802.15.4a [85], for example, every packet is preceded by a multi-bit synchronization preamble. The signature authenticates the acknowledgement and the request by means of a *shared secret* S. What follows is a formal description of the protocol.

REQ $\mathrm{V} \longrightarrow \mathrm{P} : a$
ACK $\mathrm{P} \longrightarrow \mathrm{V} : b$
SGN $\mathrm{P} \longrightarrow \mathrm{V} : \mathrm{sign}_S(a, b)$

The quantities a, b and $\mathrm{sign}_S(\cdot)$ are k-bit long. Therefore, the probability for an adversary to successfully guess one of these quantities is 2^{-k}. Such a probability gets negligible for a sufficiently large value of k, which we call the *security parameter*.

The verifier measures the distance between itself and the prover, by measuring the round-trip time $\hat{T}$ between the request and the acknowledgement messages. With

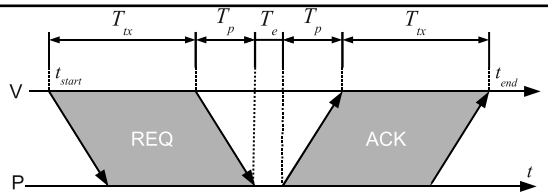


Figure 2.1. Round-trip time

reference to Fig. 2.1, we denote by t_{start} the instant when the transmission of REQ begins, and by t_{end} the instant when the reception of ACK ends. We denote by T_e the time interval from the end of REQ reception, to the beginning of ACK transmission. Since ACK does not depend on REQ, T_e does not include any elaboration time. It includes only the time for the antenna to switch from the receive mode to the transmit mode and the necessary hardware delays. We assume T_e to be small and known. Dedicated hardware can fulfill these requirements. We further denote by T_{pkt} the transmission time of the request and acknowledgement messages, and with T_p their propagation time in the medium. The round-trip time will be:

$$\hat{T} = 2T_p = (t_{end} - t_{start}) - 2T_{pkt} - T_e \tag{2.1}$$

Finally, we obtain a measure of the distance:

$$\hat{d} = \frac{c \cdot \hat{T}}{2} \tag{2.2}$$

where c is the speed of light.

The distance measurement precision depends on the capability of measuring the time interval with nanosecond precision. Localization systems based on IR-UWB can achieve nanosecond precision of measured time of flight, and consequently a distance estimation with an uncertainty of 30 cm. Also, this feature of time precision are available only with dedicated hardware.

IR-UWB protocols like 802.15.4a provides packets made up of two parts: a preamble and a payload. The preamble permits the receiver to synchronize to the transmitter and to precisely measure the time of arrival of the packet. The payload carries the information bits. In our protocol, a and b are transmitted in the payload part. We suppose the last part of the payload to carry a forward error correction code (FEC), for example some CRC bits.

In a non-adversarial scenario, the *actual distance* d will be equal to the *measured distance* $\hat{d}$. To deceive the measurement process, the adversary has to bring the verifier to measure a fake round-trip time. That is, she must act in a way that the verifier receives the acknowledgement at a different instant of time, while still receiving the correct signature. The basic idea of distance-bounding protocol is that an external adversary cannot deliver a copy of the legitimate acknowledgement *before* than the legitimate one.

On the other hand, she can deliver a copy of the acknowledgement *after* the legitimate one. In other words, she can only *enlarge* the measured distance, not *reduce* it. Thus, we are always sure that $d \leq \hat{d}$, i.e., the measured distance is a secure upper bound for the actual distance.

2.3 Threat model

We assume that the adversary (M) is an external agent, meaning that she does not know the shared secret (S) ant it cannot be stolen. Techniques like trusted hardware and remote attestation can help defending against these possibilities [53]. The objective of M is to deceive the verifier into measuring an enlarged round-trip time:

$$\hat{T} = 2T_p + \Delta T \tag{2.3}$$

in order to make it infer an enlarged measured distance:

$$\hat{d} = \frac{c \cdot \hat{T}}{2} = d + \frac{c \cdot \Delta T}{2} \tag{2.4}$$

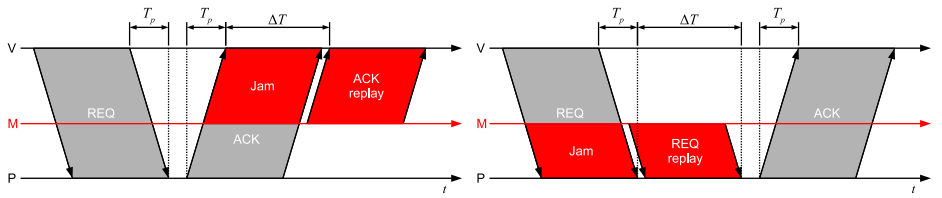
We do not deal with distance reduction attacks. Since our protocol is an enhancement of the reference distance-bounding protocol of Section 2.2, it offers the same guarantees against distance reduction attacks.

2.3.1 Adversary's capabilities

M can eavesdrop, transmit or jam any signal in the wireless channel. The principle of a jammer is to generate a radio noise at a power comparable or higher than the legitimate one. In case of IR-UWB channels, a jammer could send periodic UWB pulses, in such a way to disrupt the synchronization process [80]. Alternatively, she could simply send random pulses in the payload part, in such a way the receiver discards the packet as corrupted after the FEC test. In both cases, the goal of the jammer is to disrupt the reception of the message.

M can transmit or jam *selectively*, in such a way that only a target node receives. In the meanwhile, M can correctly eavesdrop other signals. To do this, she can place a transmitting device nearby the receiver, and a listening one nearby the transmitter. Alternatively, she can use a single device with two directional antennas. One of them transmits to the receiver, while the other listens to the transmitter.

Another possibility is the *overshadowing* attack. In this attack, M injects a fake signal with higher power than the original one. The original signal becomes entirely overshadowed by the attacker's signal. Ideally, original signal is treated as noise by the receiver. In this chapter, we do not deal with this attack, and we focus only with jam-and-replay attacks. The overshadowing attack is indeed interesting and deserves a full analysis, that we are planning to do in future work. Here we only points out that



(a) Jam-and-replay on ACK. (b) Jam-and-replay on REQ.

Figure 2.2. Jam-and-replay attack

it is not simple to be performed in a real-world IR-UWB protocol. In fact, the verifier does not receive only the fake signal, but the legitimate signal too. Even if the former is much stronger in power, the latter is still a valid IR-UWB signal, which interferes with the packet synchronization and reception. Sending an overshadowing signal is probably not enough. The adversary should also attenuate the legitimate signal with some complementary technique, such as electro-magnetic shields or similar.

We assume that M has no physical access to the prover or the verifier. This has two consequences: (i) she cannot tamper with the nodes and steal their secret material, and (ii) she cannot attenuate the wireless signals with electro-magnetic shields or Faraday cages.

2.3.2 Jam-and-replay attacks

In the distance-bounding protocol of Section 2.2, the adversary can enlarge the measured round-trip time in the following way (Fig. 2.2a).

1. M listens to the radio channel, until she hears a REQ signal.
2. M waits for the ACK signal.
3. M jams the ACK signal and eavesdrop it in the meanwhile.
4. After a time ΔT, M replays it.

The adversary must replay the ACK signal selectively, in such a way that only the verifier receives it. Otherwise, the prover will also receive the replayed signal, and could infer that the protocol is under attack.

It is important to highlight that M has to wait for the legitimate ACK to end, before starting the transmission. This is because she must avoid signal collision.

The adversary can perform a similar attack on the REQ signal (Fig. 2.2b). Even in this case, M has to wait for the end of the legitimate REQ before starting her transmission.

We state the following:

Proposition 1. *In a jam-and-replay attack on REQ/ACK, the adversary must enlarge the round-trip time of a quantity ΔT not smaller than T_{pkt}, i.e., $\Delta T \geq T_{pkt}$.*

Proposition 1 represents the fundamental limitation of the jam-and-replay attacks. SecDEv will leverage on this to withstand them. Note that this limitation comes from the properties of the radio-frequency channel, and does not depend on how many devices the adversary controls. For the sake of simplicity, Figg. 2.2a and 2.2b show a single adversary.

2.4 SecDEv protocol

SecDEv is a distance-bounding protocol, which measures the correct distance between a verifier V and a prover P in presence of an adversary M performing a jam-and-replay attack. It is similar to the reference distance-bounding protocol (cfr. Section 2.2), except that the length of REQ and ACK do not depend only on the security parameter, but also on a *security horizon*.

Let us consider the Equation 2.3 for a general enlargement attack and apply the Proposition 1, we obtain the constraint $\hat{T} \geq 2T_p + T_{pkt}$. Hence:

$$\hat{T} \geq T_{pkt} \tag{2.5}$$

Equation 2.5 assures us that a measured round-trip time smaller than T_{pkt} has not been affected by any jam-and-replay attack. We can translate T_{pkt} in a distance d_M, that we call *security horizon*:

$$d_M \triangleq \frac{cT_{pkt}}{2} \tag{2.6}$$

In terms of distances, Equation 2.5 becomes:

$$\hat{d} \geq d_M \tag{2.7}$$

Equation 2.7 is our test to distinguish between trusted and untrusted distance measurements. V can extend the packet transmission time to enlarge the security horizon (cfr. Eq. 2.6), in order to securely measure longer distances. T_{pkt} is enlarged by introducing padding bits after the nounce. Padding bits have not to be unpredictable. They can have a well-known value (e.g. all zeroes), since they serves only to prolong the packet transmission time. V decides on the length of the REQ padding, and P has to respond with the same padding length in the ACK. Therefore, both messages have the same length, to withstand both jam-and-replay on REQ and on ACK.

Let us explain the protocol in detail. We assume that the wireless channel is characterized by the parameter tuple: $\{T_{pre}, R_{pld}, T_e\}$. T_{pre} is the transmission time of the preamble part. R_{pld} is the bit rate of the payload part. T_e is the reaction time of the prover node. In addition, we define the following triplet of protocol parameters: $\{k, S, d_M\}$. k is the security parameter. A higher value for k implies a higher security

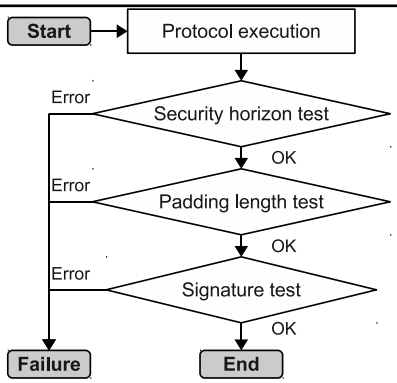


Figure 2.3. SecDEv algorithm

level, but has an impact on power consumption, as we will see in the following. S is a secret bit sequence shared between V and P. Its length is longer than or equal to k. d_M is the security horizon that distinguishes between trusted and untrusted measured distances. If the actual distance d is longer than d_M, the measured distance cannot be trusted because it may be affected by a jam-and-replay attack. In such a case, the protocol can be executed again with a longer d_M. Alternatively, the distance d can be first estimated in an insecure manner, and then securely confirmed with $d_M > d$. A higher value for d_M allows us to measure longer distances, but has an impact on power consumption.

We further define the following quantities. N_{pad} and N_{fec} are respectively the number of bits of the padding and the FEC code. Since the number of bits of a and b is k, the total transmission time will be:

$$T_{pkt} = T_{pre} + (k + N_{pad} + N_{fec})/R_{pld} \tag{2.8}$$

If with $N_{pad} = 0$, the T_{pkt} identifies the minimum value of d_M. Thus, if the actual distance is smaller than this value, there is not need of padding bits. Otherwise, we determine N_{pad} with the following formula:

$$N_{pad} = \left\lceil \left(\frac{2d_M}{c} - T_{pre} \right) \cdot R_{pld} \right\rceil - k - N_{fec} \tag{2.9}$$

Using the Equation 2.9, we can set every value of d_M. Note that T_{pkt} grows with d_M. A larger security horizon causes longer messages, accordingly higher energy consumptions per protocol execution. An implementer must choose d_M as a trade-off between ranging capabilities and power consumption.

Fig. 2.3 shows the algorithm executed by V. After the protocol execution, V tests whether the measured distance is within the security horizon, that is, if $\hat{d} < d_M$. If this test fails, the measured distance is discarded as untrusted. Then, V tests the length of the ACK padding. If it contains less bits than the REQ one, the measured distance is discarded as untrusted. This is to avoid a jam-and-replay attack on REQ

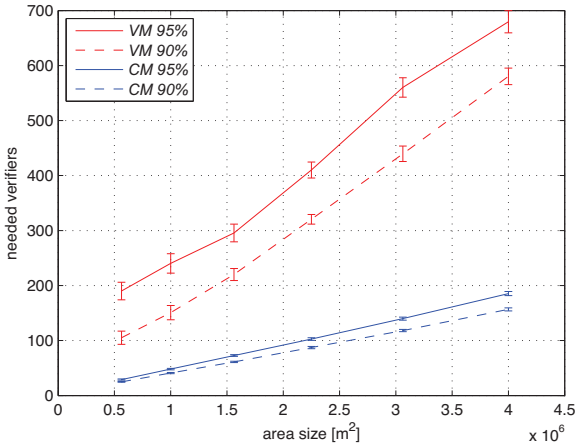


Figure 2.4. Verifiers required to cover an area

(cfr. Fig. 2.2b), in which M tries to lower ΔT by replaying REQ with a smaller padding. In such a case, P will respond with an ACK with a smaller padding too, and the attack will not pass the padding length test. Finally, V tests the validity of the cryptographic signature.

2.5 Experimental results

We combined SecDEv with multilateration technique to securely localize the prover. We analyzed the efficiency of this solution in terms of covered area and we compared it with *verifiable multilateration* [100], which is the state-of-the-art technique for secure positioning in wireless networks. Verifiable multilateration involves at least three distance measurements from different verifiers. The distance measurements are performed by means of distance bounding protocols, which are supposed to withstand reduction attacks. Verifiable multilateration deals with possible enlargement attacks by forcing an additional check to the final position estimation. In order to be trusted, the position must be inside the polygon formed by the verifiers, otherwise it is discarded as untrusted. Intuitively, this reduces the coverage area of the positioning technique.

In other words, classic multilateration is more scalable in terms of number of verifiers needed to cover a specific area. To quantify this, we have tested the performance of classic multilateration in terms of number of verifiers needed to cover a working area, and we have compared our results with those of verifiable multilateration, taken from [100]. We supposed that every verifier covers a circular area with radius $250\,\mathrm{m}$.

We neglect planned distributions [100], because in a real deployment, environment may impose constraints on the verifier positioning. Thus, we consider that the verifiers are uniformly distributed over the area of interest.

In order to evaluate the two techniques under the same conditions, our simulation were performed on areas of variable sizes. The verifiers were uniformly distributed in the area and in a boundary region outside the area, whose width was 10% of the area width. We use the boundary region to avoid the boundary effects [100] in the verifiable multilateration.

Fig. 2.4 shows how many verifiers are required to cover 95% and 90% of the working area. VM and CM curves are respectively verifiable multilateration with distance bounding and classic multilateration with SecDEv. The number of verifiers is the average of 100 simulations with confidence intervals of 95% calculated for different values of working area from $0.5km^2$ to $4km^2$. The chart shows that classic trilateration needs far less verifiers, because it has not the limitation of the verification triangles. This gives strong motivation to fight distance enlargement attacks.

3

Feasibility of overshadow enlargement attack on IEEE 802.15.4a distance bounding

In this chapter we make a preliminary analysis of the feasibility of enlargement attacks through *overshadow* strategy against indoor 802.15.4a-based distance-bounding protocols. In an overshadow strategy, the adversary receives and retransmits a legitimate packet with a certain delay and a stronger power. The legitimate packet gets thus "overshadowed" by a delayed copy of it. In this way, the adversary tries to delay the entire process of round-trip time measurement. In general, overshadow strategies are considered feasible in the literature [100]. Instead, we show that they are not easy to carry out and, depending on the delay introduced by the adversary, there are cases in which they have no effect or their effect is not controllable. This chapter has been published as a journal paper [94].

The remainder of this chapter is organized as follows. In Section 3.1 we describe the classic two-way ranging technique and we introduce our reference distance-bounding protocol. In Section 3.2 we describe the 802.15.4a UWB ranging technique and the receiver technology. In Section 3.3 we define the threat model. In Section 3.4 we analyze the overshadow attack strategies against 802.15.4a and their actual feasibility.

3.1 Two-way ranging and distance bounding

Two-way ranging (TWR) is the most widely used procedure to estimate the distance between two devices, i.e., a *verifier* (V) and a *prover* (P) in an asynchronous wireless network [29]. The TWR procedure works as follows (cfr. Fig. 3.1). First, V sends a request packet to P at time t_0. P receives it at time t_1, after a time of flight $T_{of} = d_{\mathrm{V,P}}/c$, where $d_{\mathrm{V,P}}$ is the distance between V and P and c is the speed of light. After some delay T_d, P replies with an acknowledgement packet at time t_2. The reply arrives at V at time t_3 after T_{of}. The verifier can estimate $T_{of} = (t_3 - t_0 - T_d)/2$, since the value of T_d is assumed known to V as well. Finally, the V-P distance is obtained by $d_{\mathrm{V,P}} = T_{of} \cdot c$.

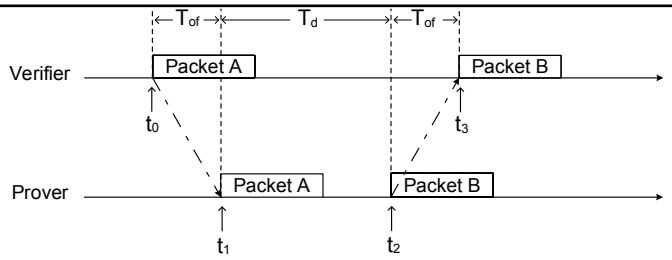


Figure 3.1. TWR procedure

In order for the TWR procedure to give an accurate distance estimate, P and V must precisely measure the arrival times of the packets they receive. Wireless UWB protocols provide nanosecond precision, leading to a sub-meter accuracy on distance estimation.

The TWR procedure is not secure by itself. An external adversary can indeed impersonate a legitimate prover and transmit a fake acknowledgment packet, thus deceiving the verifier into measuring a false distance. A proposed solution is to implement a *distance-bounding protocol* [15] on the top of it. A simple example, proposed in [81] for 802.15.4a-based systems, is the following:

REQ $\mathrm{V} \longrightarrow \mathrm{P} : a$
ACK $\mathrm{P} \longrightarrow \mathrm{V} : b$
SGN $\mathrm{P} \longrightarrow \mathrm{V} : \mathrm{sign}_S(a, b)$

The request packet (REQ) and the acknowledgment packet (ACK) convey, respectively, a and b, which are two independent and unpredictable sequences of bits. The signature packet (SGN) authenticates the request and the acknowledgment by means of a shared secret S. The verifier estimates the distance between itself and the prover, by measuring the round-trip time between REQ and ACK packets. We use such a protocol as our *reference distance-bounding protocol*. The considerations we make about the overshadow attack hold for more complex distance-bounding protocols as well.

3.2 IEEE 802.15.4a physical layer

We focus on the IEEE 802.15.4a standard [55] for TWR operations. IEEE 802.15.4a introduces an impulse radio ultra-wide band (IR-UWB) PHY protocol capable of sub-meter precision in TWR operations in indoor or urban environments. It has been the first standardized UWB protocol for precise ranging, and it is one of the most probable choices for future implementations of wireless distance-bounding protocols [81].

On the contrary, there is no such requirement for the SGN. So we are free to map it into another UWB packet, as well as into a packet of a different protocol, e.g. “vanilla” 802.15.4. To better analyze the feasibility of overshadow-based enlargement attacks against 802.15.4a UWB, it is necessary to explore the effects of such attacks from a

physical-layer point of view. Thus, in the following we give some more details on the structure of the transmitted signal prescribed by the 802.15.4a IEEE standard [55] and on a characteristic ranging algorithm suited for it. The UWB packets are made up of three major segments: a synchronization header (SHR), a physical-layer header (PHR), and a data field. We begin by describing the SHR, which is used for the time-of-arrival (TOA) estimation. The SHR consists of two blocks: a synchronization preamble (SYNC) and a start-of-frame delimiter (SFD). The mathematical model of the signal transmitted during the SHR is:

$$s(t) = \sum_{i=0}^{N_{SHR}-1} a_i \psi(t - iT_{sym}) \tag{3.1}$$

where $N_{SHR} = N_{SYNC} + N_{SFD}$, N_{SYNC} and N_{SFD} are the number of symbols in the SYNC and SFD, respectively, and T_{sym} is the symbol duration. Symbols a_i are all equal to 1 during the SYNC while they take values $\{-1, 0, +1\}$ during the SFD. Finally, $\psi(t)$ is expressed as:

$$\psi(t) \triangleq \sum_{k=0}^{K_{pbs}-1} d_k p(t - kT_{pr}) \tag{3.2}$$

where $\{d_k\}_{k=0}^{K_{pbs}-1}$ is a *perfectly balanced sequence* with elements $\{-1, 0, +1\}$, $p(t)$ is an ultra-short causal pulse (*monocycle*) and $T_{pr} \triangleq T_{sym}/K_{pbs}$ is the pulse repetition period.

The transmitted signal $s(t)$ arrives at the receiver through multiple propagation paths (*multipath channel*), characterized by different attenuations and delays. Denoting by $h(t)$ the *channel response* (CR) to $p(t)$[1], the received signal can be written as

$$r(t) = \sum_{i=0}^{N_{SHR}-1} \sum_{k=0}^{K_{pbs}-1} a_i d_k h(t - kT_{pr} - iT_{sym} - t_{TOA}) + w(t) \tag{3.3}$$

where $w(t)$ is thermal noise. In the above equation, t_{TOA} is the time-of-arrival instant of the signal at the receiver and represents the parameter to be measured. It coincides with t_1 or t_3 in the verifier-prover and prover-verifier channels, respectively, according to the TWR procedure depicted in Fig. 3.1.

We consider a simple non-coherent energy-based receiver which guarantees high ranging precision with low cost and low power consumption. Here, $r(t)$ is first passed through a band-pass filter (BPF), to remove the extra-band noise, and then is demodulated in a square-law device followed by a low-pass filter (LPF).

The ranging operation is concerned with the estimation of the position, t_{PHR}, of the first peak of the first pulse of the PHR [55]. Such a peak represents the arrival

[1] Without loss of generality, it is assumed that $h(t)$ starts at $t = 0$.

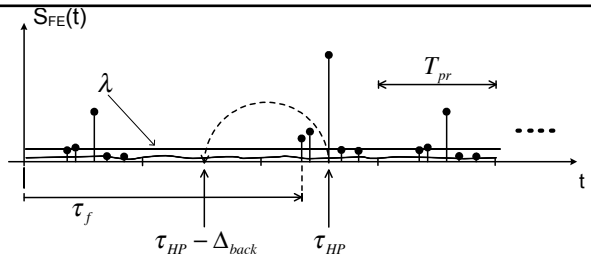


Figure 3.2. Fine timing acquisition procedure

of the *ranging marker* and is conventionally taken as the time of arrival of the entire signal packet [55]. In fact, estimating t_{PHR} is equivalent to estimate t_{TOA}, since $t_{TOA} = t_{PHR} - N_{SHR}T_{sym}$.

We consider the TOA-estimation procedure described in [29], but the conclusions we draw are valid also with other threshold-based TOA estimation algorithms. In particular, TOA estimation is performed in the following three steps. The *frame detection* step decides through energy measurements whether a packet is present or not. The *fine timing acquisition* step produces a fine estimate of the arrival time t_{PHR} with an ambiguity of multiples of T_{sym}. Finally, the *SFD detection* step disambiguates the estimate of t_{PHR} through a correlation mechanism.

We write t_{PHR} as a multiple of T_{sym} plus a fractional part $\tau_f \in [0, T_{sym})$, i.e., $t_{PHR} = \tau_f + NT_{sym}$. The *fine timing acquisition* phase and the *SFD detection* phase deal with the estimation of τ_f and N, respectively.

We now focus on the *fine timing acquisition* procedure. Indeed, as we show later, this is the only step of the ranging operation that the adversary can attack. The *fine timing acquisition* scheme we analyze is described in detail in [29] and essentially consists in the correlation of the signal $y(t)$ at the output of LPF with K_{pbs} cyclic-shifted versions of the sequence $\{d_k^2\}_{k=0}^{K_{pbs}-1}$. This produces a T_{sym}-long signal, say $S_{FE}(t)$ [2], whose support is in the interval $[0, T_{sym})$, which is used for the estimation of τ_f. Specifically, the estimation of τ_f is performed in two steps. In the first step (*highest-peak search*) the position τ_{HP} of the maximum of $S_{FE}(t)$ is sought for. In the second step (*leading-peak search*), starting from τ_{HP} we jump back by Δ_{back} seconds and proceed forward looking for the first time $S_{FE}(t)$ crosses a given threshold λ whose value depends on the thermal noise. The distance of the crossing time from the beginning of $S_{FE}(t)$ provides an estimate of τ_f. The *fine timing acquisition* procedure is described in Fig. 3.2.

[2] For $t = \tilde{m}T_{pr} + \tilde{\varepsilon}$, with $0 \le \tilde{m} \le K_{pbs} - 1$ and $\tilde{\varepsilon} \in [0, T_{pr})$, $S_{FE}(t)$ coincides with $S'(\tilde{m}, \tilde{\varepsilon})$ defined in [29].

3.3 Adversary model

We consider an adversary (M) who wants to deceive the verifier into accepting a specific enlarged distance measurement. Since the distance measurement is obtained from a round-trip time measurement at V, the adversary's aim is to enlarge such a round-trip time measurement, by introducing a controlled delay. She tries to obtain this by means of an *overshadow* strategy. Following this strategy, the adversary eavesdrops and retransmits a legitimate UWB packet with a certain delay and a stronger power. The legitimate signal and the adversarial one get thus overlapped at the victim's receiver. The idea at the basis of the attack is that the victim receiving two signals, both characterized by the expected structure, will hook to the stronger (malicious) one, thus obtaining an enlarged measurement of the round-trip time. Note that the adversary must transmit its signal in such a way that only the victim receiver is able to hear it. Otherwise, the presence of a malicious transmitter would be easily detected. This attack is considered feasible by the literature [100].

The adversary can attack the prover (by overshadowing the REQ), as well as the verifier (by overshadowing the ACK), as well as both. Without loss of generality, we assume overshadowing of the REQ signal but the analysis holds also for the ACK-overshadowing attack.

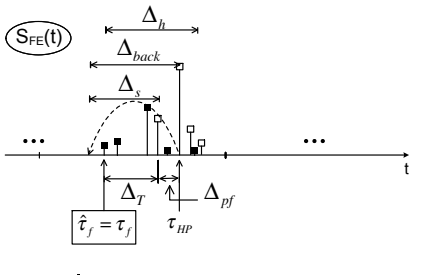
Finally, we observe that our adversary has no interest in jamming the legitimate signal or a part of it. In fact, jamming would not avoid the prover from starting the TOA estimation procedure, which is triggered by an energy threshold (cfr. Section 3.2). It would only disturb the TOA measurement in a random way, causing delays which are not controllable by the adversary.

3.4 Feasibility of the overshadow attack

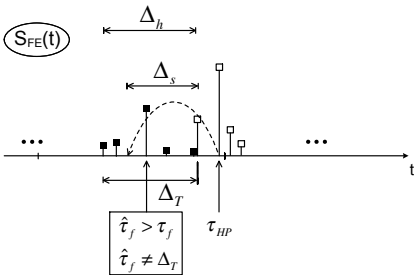
First of all, we observe that an overshadow attack has not a harmful effect on the *frame detection* procedure. It only produces the positive effect of increasing the energy measured by the receiver thus anticipating the estimation of the presence of the packet.

The overshadow attack may have a harmful effect on the *SFD detection*. However, it would result in a delay multiple of $T_{sym} = 3968\,\mathrm{ns}$ [55]. Such a delay corresponds to an enlargement of $595\,\mathrm{m}$, which is unrealistic for an indoor scenario. We assume that the application layer employs threshold mechanisms to exclude enlargements longer than $595\,\mathrm{m}$.

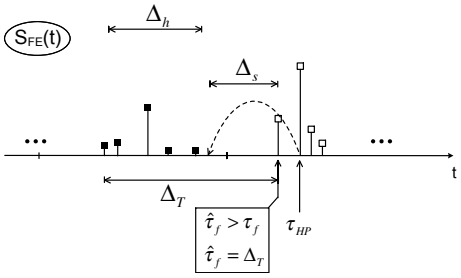
Now, we analyze the effects of the overshadow attack on the *fine timing acquisition* procedure. We make the pessimistic hypothesis that M is synchronized with V and has a perfect knowledge of the position of both P and V. Under these assumptions, M can make its message to arrive at P with a controlled delay Δ_T relative to the message sent by V. Therefore, the signal received by P is:



(a) Case 1 – No effect.



(b) Case 2 – Uncontrollable effect.



(c) Case 3 – Controllable effect.

Figure 3.3. Overshadow attack. Full and empty marks represent the components of $S_{FE}(t)$ associated to the signal transmitted by V and M, respectively.

$$r(t) = r^V(t) + r^M(t - \Delta_T) \tag{3.4}$$

where $r^V(t)$ and $r^M(t)$ are the signals associated to V and M, respectively. The signal $S_{FE}(t)$ used by the *fine timing acquisition* algorithm (see Section 3.2) has the shape shown in Fig. 3.3a. We have represented only the pulses above the threshold to ease the drawing. In Fig. 3.3a we have introduced two new parameters: Δ_{pf} and Δ_h. Specifically, Δ_{pf} represents the delay between the highest pulse and the first pulse in $r^M(t)$, while Δ_h represents the time dispersion of the propagation channel between verifier and prover. For the following discussion it is useful to define $\Delta_S \triangleq \Delta_{back} - \Delta_{pf}$. For $\Delta_T < T_{pr}$, three different cases are possible depending on the value of Δ_T.

1. $\Delta_T \in [0, \Delta_S]$ (Fig. 3.3a). In this case, the first pulse of the legitimate signal is correctly identified by the prover. The *fine timing acquisition* gives a correct estimate of the TOA, i.e., $\hat{\tau}_f = \tau_f$, where $\hat{\tau}_f$ represents the estimate of τ_f. The overshadow attack is ineffective.
2. $\Delta_T \in (\Delta_S, \Delta_S + \Delta_h]$ (Fig. 3.3b). In this case, a non-first pulse of the legitimate signal is identified as the first pulse, and thus the overshadow attack produces a timing enlargement. However, this enlargement is not controllable by the adversary since it depends on the propagation channels between V and P, and M and P. Thus, we have $\hat{\tau}_f > \tau_f$ but $\hat{\tau}_f \neq \tau_f + \Delta_T$.

Table 3.1. Mean Absolute Error of the Enlargement

	case 1	case 2	case 3
regular adversary	(no effect)	$7.79\,\mathrm{m}$	$0.15\,\mathrm{m}$
close adversary	(no effect)	$8.52\,\mathrm{m}$	$0.14\,\mathrm{m}$

3. $\Delta_T \in (\Delta_S + \Delta_h, T_{pr})$ (Fig. 3.3c). In this case, the first pulse of the malicious signal is identified as the first pulse, i.e., $\hat{\tau}_f = \tau_f + \Delta_T$. This is the only situation in which M is able to introduce a timing enlargement equal to Δ_T.

The case $\Delta_T \geq T_{pr}$ can be dealt with in a similar manner. Note that Δ_S and Δ_h depend on the channel, which is not deterministic. So, for a fixed Δ_T, the occurrence of each of the three cases will be expressed as a probability.

We simulated overshadow attacks to test their feasibility in a standard residential scenario (CM1) [70]. The signal parameters are set as done in [29]. The performance of the attacks has been assessed by measuring the mean absolute error (MAE) of the enlargement, i.e. the difference between the achieved enlargement and the target enlargement. We simulated both a *regular adversary*, which experiences an M-P channel following the CM1 model, and a *close adversary,* for which the M-P link can essentially be characterized by a single, line-of-sight, component. We assumed a signal-to-noise ratio $E_s/N_0 = 30\,\mathrm{dB}$, where E_s is the energy of a symbol, and N_0 is the noise spectral power density.

Table 3.1 shows the values of MAE in cases 2 and 3. As expected, the overshadow attack is effective and controllable only when case 3 occurs, both for regular and for close adversary. Observe that an attack with an uncontrollable effect could also be useful for an adversary. However, this is not the case in trilateration-based positioning in which the enlargement must be controllable in order the position to be spoofed in a coherent manner.

Figs. 3.4 and 3.5 illustrate the probability of the above three cases as a function of the target distance enlargement $\Delta_D = \Delta_T \cdot c/2$, with the regular and the close adversary, respectively. Experiments confirmed that a controllable attack (i.e., occurrence of case 3) is impossible for many values of Δ_D, and reaches the maximum probability of 18% at $\Delta_D = T_{pr} \cdot c/2 = 19.2\,\mathrm{m}$. Such a probability does not increase in the case of an adversary with a strong line-of-sight component.

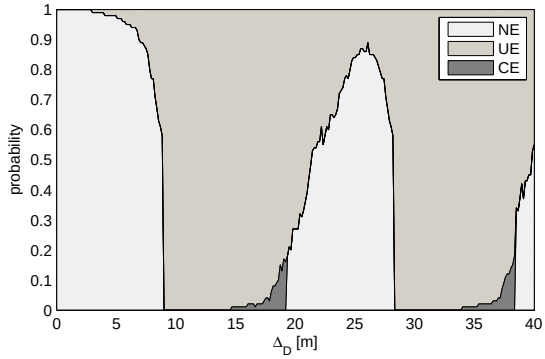


Figure 3.4. Probability of No Effect (NE), Uncontrollable Effect (UE) and Controllable Effect (CE) cases with a regular adversary.

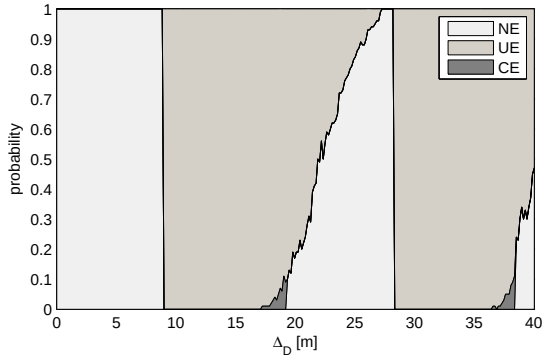


Figure 3.5. Probability of No Effect (NE), Uncontrollable Effect (UE) and Controllable Effect (CE) cases with a close adversary.

4

EMCD-ML: secure positioning through enlargement miscontrol detection

In this chapter, we estimate the controllability of an enlargement attack by PHY-level simulations of the IEEE 802.15.4a UWB protocol in an indoor scenario. We determine the adversary's best tactics to enlarge a ranging operation and estimate her success probability. Then, we propose *EMCD-ML* (*Enlargement MisControl Detection MultiLateration*), a secure positioning algorithm based on the difficulty of the adversary to control the effect of enlargement attacks. EMCD-ML is based on the fact that the adversary has a low probability to enlarge the distance measurements in such a way to coherently spoof the position. Such a probability gets even lower in case of more precise or more redundant multilateration systems. We show that it is possible to establish a trade-off between security, namely probability of adversary success, and performance, namely number of anchors, ranging operations, and coverage area. In practice we can achieve a high level of security saving 93% of the anchor nodes with respect to the state-of-the-art solutions.

With respect to Chapter 4, we consider a more powerful adversary, that improves her overall success probability (i) by knowing the statistical channel characteristics, and (ii) by attacking both the challenge and the response messages.

The remainder of the chapter is organized as follows. In Section 4.1 we compare EMCD-ML with other state-of-the-art solutions. In Section 4.2 we introduce our system model. In Section 4.3 we define the adversary and her capabilities. In Section 4.4 we analyze the effects of enlargement attacks against IEEE 802.15.4a UWB, and estimate their success probabilities. In Section 4.5 we describe EMCD-ML. In Section 4.6 we evaluate EMCD-ML in terms of security and anchor-node scalability.

4.1 Comparison of EMCD-ML to the state of the art

Secure positioning systems are traditionally classified in *range-based* [19, 100, 108] and *range-free* [62, 101]. Range-based systems rest on the *ranging* operation, i.e. the measurement of the distance between two devices, typically the prover and an anchor node whose position is known. On the contrary, range-free systems are not

based on the (direct) measurement of geometric quantities. They deduce the position from other, high-level information. Typical example is the hearing of beacon messages from the prover node. Range-free systems are typically cheaper as they do not require special hardware for distance measurements. However, they allow for a worse precision in the position estimation.

Our system is range-based, as it measures distances by means of IEEE 802.15.4a UWB ranging. As a consequence, it requires specialized hardware, but it allows for a more precise position estimation.

Hu et al. [54] proposed *packet leashes*, which is the first attempt to employ distance-bounding-like techniques for secure location verification. Sastry et al. [88] proposed the *Echo* protocol for secure location verification based on ultra-sound ranging. Čapkun and Hubaux [100] and, independently, Zhang et al. [108] proposed *verifiable multilateration*. The system measures the distances from a set of trusted verifier nodes by means of distance-bounding protocols. The position is computed by means of multilateration, and it is considered secure if it lies inside the polygon formed by the involved verifiers (*in-polygon check*). Indeed, if the measured position has been falsified, at least one of the distance-bounding protocols must have been exposed to a distance-reduction attack, which is infeasible. Verifiable multilateration is capable of resisting both to external adversaries and to dishonest provers, but fails in presence of more-than-one colluding provers.

Chiang et al. [19] improves its resistance against such a threat by introducing the concept of *simultaneous multilateration*. Simultaneous multilateration requires perfect synchronization between the verifiers. The request messages are sent by the verifiers in such a way to reach the prover simultaneously, and the prover sends back a single broadcast acknowledgment. This solution increases the necessary number of colluders to mount the attack, but does not eliminate the threat. Chandran et al. [17] give a general impossibility result on this, stating that if the number of colluders grows linearly with the number of verifiers, no range-based secure positioning methods are possible. In this chapter, we refer only to external adversaries. In presence of dishonest provers, it is easy to mount and control enlargement attacks, so our countermeasure is ineffective.

Another research track aims at securing the existing civilian GPS technology, either by modifying the receivers [71], or by extending the protocol [103]. GPS can cover only outdoor scenarios, and usually has a precision of 5–10 meters. We focus on localization based on UWB ranging, that offers outdoor as well as indoor coverage, and can reach sub-meter precision.

On the related topic of distance-bounding protocols, several solutions have been proposed [1, 15, 16, 25, 50], offering different properties in terms of security, efficiency, and noise tolerance. Clulow et al. [22] first showed the theoretic possibility of PHY-level attacks against distance bounding. Among others, they presented *early detection* and *late commit* attacks. Both leverage on the fact that, in real-life PHY protocols, cryptographic symbols are transmitted as long RF signals in order to decrease the

probability of noise errors. The adversary can thus leverage on these idle times in order to anticipate the request-acknowledgment mechanism, and to cause a reduction on the measured distance.

Poturalski et al. [81, 41] conducted a deep study on PHY-level reduction attacks against IEEE 802.15.4a UWB distance bounding. They evaluated the impact in terms of reduction meters, and proposed a set of countermeasures to limit such an impact. In the present chapter, we consider a distance-bounding protocol *immune* to reduction attacks, and we focus on enlargement attacks.

The same authors [82, 80] studied also the feasibility and the impact of *interfering attacks* against the preamble, and proposed as a countermeasure a novel time-of-arrival (ToA) estimation algorithm called *PIDH* (*power independent detection with Hamming distance*). In the present chapter, we consider classic ToA-estimation algorithms, namely jump-back-search-forward and search-back [29, 49]. We leave the security analysis of EMCD-ML with non-classic ToA estimators for future works.

Chiang et al. [19] proposed the first method to mitigate the effects of enlargement attacks in wireless distance bounding. Their method is based on signal strength difference, and requires that the verifier is equipped with a pair of antennas collinear with the position of the prover. Our approach does not require multiple antennas, and it is based on the difficulty to control an enlargement attack. Actually, the two countermeasures are orthogonal. They could be applied together to improve the overall security of the system.

In Chapter 2 we presented a PHY-independent wireless distance-bounding protocol resisting to enlargement attacks based on a *jam-replay* strategy. In the IEEE 802.15.4a PHY protocol, a jam-replay attack would produce unrealistic enlargements (hundreds of kilometers). This is due to the transmission time of a packet, which is on the order of milliseconds. We focus on *overshadow* attacks, which are more feasible to obtain a distance enlargement against IEEE 802.15.4a. In Chapter 3 we showed that, in the IEEE 802.15.4a UWB ranging standard, an overshadow-based enlargement attack is poorly controllable by the adversary. In this chapter, we consider a more powerful adversary, which improves her probability to control the attack (i) by knowing the statistical characteristics of the channel, and (ii) by attacking both the challenge and the response messages.

4.2 System model

Our system is a multilateration algorithm that determines the position X of a mobile node in the 2-dimensional plane by measuring $N \geq 3$ distances $d_1, \ldots, d_N$ of the node from N anchor nodes, whose positions $\mathrm{V}_1, \ldots, \mathrm{V}_N$ are known. Following the terminology of distance bounding, we will call *prover* the mobile node, and *verifiers* the anchor nodes. The multilateration algorithm finds X as the intersection of the circumferences with centers V_i and radii d_i (*ranging circumferences*). In presence of some imprecision, the measured distances $\widehat{d}_i$ will be affected by an error e_i:

$$\widehat{d_i} = d_i + e_i \tag{4.1}$$

In such a case, the ranging circumferences will not intersect in a point. The *measured position* $\widehat{X}$ will thus be the pseudo-solution in the least-squared-error sense:

$$\widehat{X} = \arg\min \sum_i \epsilon_i^2 \tag{4.2}$$

$$\|\widehat{X} - \mathrm{V}_i\| = \widehat{d_i} - \epsilon_i \tag{4.3}$$

The multilateration process will give as output the measured position and a set of N *residuals* $\epsilon_1, \ldots, \epsilon_N$. The residuals are an indirect estimation of the measurement errors e_i. High values of the residuals generally imply high errors. The opposite is not true, because high errors could geometrically "compensate" each other in a single point, and thus give low residuals. Table 4.1 shows the symbols used in this chapter and their meaning.

4.2.1 Ranging operation

Each verifier measures the distance by means of a *two-way ranging* (*TWR*) operation. The verifier estimates the distance by measuring the round-trip time (T_{RTT}) between the transmission of a *request packet* and the reception of an *acknowledgement packet*. If the *response time* (T_{resp}) of the prover is known, the distance can be estimated by:

$$\widehat{d_i} = \frac{T_{RTT} - T_{resp}}{2} \cdot c \tag{4.4}$$

where c is the speed of light. The precision of the distance estimation depends on the precision with which provers and verifiers estimate the packets' *time of arrival* (*ToA*). In multipath environments, this in turn highly depends on the bandwidth of the employed radio signals. Ultra-wide band PHY protocols like IEEE 802.15.4a can reach sub-meter precisions on the distance estimation.

The classic TWR procedure is not secure by itself. An adversary can indeed impersonate a legitimate prover and transmit a fake acknowledgement packet, thus deceiving the verifier into measuring a false distance (*impersonation attack*). A way to make it more secure is to implement a *distance-bounding protocol* [15] on the top of it. A simple example of it, proposed in [81] for IEEE 802.15.4a and for external adversaries only, is the following:

REQ $\mathrm{V} \longrightarrow \mathrm{P} : a$
ACK $\mathrm{P} \longrightarrow \mathrm{V} : b$
SGN $\mathrm{P} \longrightarrow \mathrm{V} : \mathrm{sign}_S(a, b)$

The request packet (REQ) and the acknowledgment packet (ACK) convey respectively a and b, which are two externally unpredictable sequences of bits. The *signature packet* (SGN) authenticates the request and the acknowledgment by means of a shared secret S.

Symbol:	Description:
a	Unpredictable bits generated by V
b	Unpredictable bits generated by P
S	Secret shared between V and P
$\mathrm{sign}_S(\cdot)$	Signature operation
d	Real distance between V and P
$\widehat{d}$	Measured distance between V and P
e	Error on the measured distance
e_{max}	Maximum absolute error in the honest case
d'	Adversary's objective distance
e_{ctrl}	Attack control error
P_{ctrl}	Attack control probability
c	Speed of light
T_{RTT}	Round-trip time
T_{resp}	Prover's response time
T_o	Overshadow delay
$\widehat{T}_o$	Best-tactic overshadow delay
T_e	Total delay obtained by the adversary
τ_{LP}	Real time of arrival
$\widehat{\tau}_{LP}$	Estimated time of arrival
τ_{HP}	Time of arrival of the highest peak
T_{JB}	Time jump of JBSF algorithm
T_{SB}	Time window of SB algorithm
λ	Noise threshold
d_{safe}	Maximum distance of sure reception
d_{ms}	Minimal spoofing distance
V_i	Position of the i-th verifier
X	Position of the prover
X'	Adversary's objective position
$\widehat{X}$	Prover's position measured by multilateration
ϵ_i	Residuals of the multilateration
ϵ_{max}	Acceptance threshold on the residuals
N_{min}	Minimal number of verifiers

Table 4.1. Summary of the notation

This protocol avoids the impersonation attack, because the adversary cannot forge the final signature unless she knows S. Moreover, it avoids reduction attacks, because an adversary cannot predict and anticipate the transmission of a and b.

4.2.2 Basic security assumptions

We state two fundamental assumptions that should hold in order to detect enlargement attacks. The first one is that the prover and the verifier must mutually trust each other in keeping secrets and behaving according to specifications. If this assumption does not hold, we cannot be sure (for example) that the prover does not purpose-

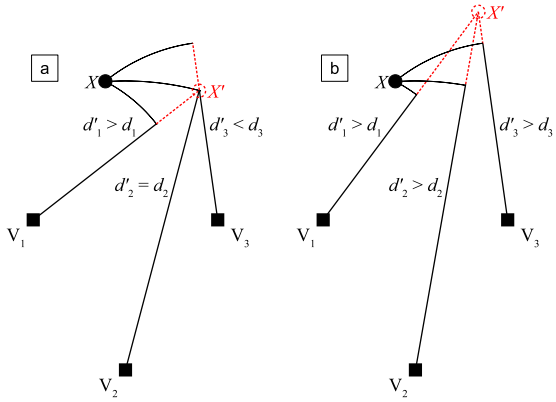


Figure 4.1. Multilateration spoofing

fully introduce delays in transmitting its acknowledgment, thus enlarging the distance measurement.

The second assumption is that the prover and the verifier must hear each other. Better stated, it must be possible to determine a distance of "safe reception" (d_{safe}), such that, if the prover and the verifier stay within this distance, they surely receive each others' signals. If this assumption does not hold, it is easy for the adversary to obtain and control a distance enlargement. She can simply replay with a controlled delay a packet unheard by the legitimate receiver. This assumption also excludes that the adversary is capable of artificially insulating a device, e.g. by means of a Faraday cage.

A possible application that meets the basic assumptions is the automatic guidance of transport robots for industrial scenarios [84]. The guidance algorithms rely on the knowledge of the position of the robots. If an adversary is able to spoof a position, she can deceive the guidance algorithm and cause severe damages to the system (collisions, etc.). In this scenario, the robots are the provers whose positions are measured by the anchor nodes. It is reasonable to suppose that the robot honestly execute the protocol, and does not leak the shared secret S. It is also reasonable to suppose that an adversary cannot shut a transport robot inside a Faraday cage, given its cumbersome size.

4.3 Adversary model

We suppose the adversary is equipped with one or more devices, that are able to communicate with each another. They can eavesdrop and transmit any signal in the wireless channel. The adversary does not have any limitation on the transmission power. The objective of the adversary is to spoof the position measured by the multilateration system. We denote by X' the *false position*, that is the position that the adversary wants to make the system believe the prover is in. In order to do this, the

adversary has to attack a number of ranging operations and make them measure *false distances* d'_i. For each verifier V_i, the adversary chooses d'_i in such a way the multilateration process gives as output X' (Fig. 4.1a). Three cases are possible:

1. $d'_i < d_i$. The distance measurement has to be reduced. The adversary has to perform a *reduction attack*.
2. $d'_i = d_i$. The distance measurement does not need to be attacked.
3. $d'_i > d_i$. The distance measurement has to be enlarged. The adversary has to perform an *enlargement attack*.

In this chapter, we consider the distance-bounding protocol as *immune* to reduction attacks. Poturalski et al. [41, 81] showed that, theoretically, the impact of reduction attacks on IEEE 802.15.4a can be indefinitely limited by tolerating decoding errors and by enlarging the unpredictable sequences conveyed by the REQ and ACK payloads. Given that reduction attacks are impossible, only false position requiring enlargement attacks are feasible (Fig. 4.1b).

From now on, we will focus on the single enlargement attack against the single ranging operation. We will thus omit the "i" subscripts on symbols to ease the notation. We call *objective enlargement* ($e' \geq 0$) the distance enlargement that the adversary wants to obtain:

$$e' = d' - d \tag{4.5}$$

With her attack, the adversary tries to cause an anomalous measurement error, which is as close as possible to her objective enlargement. In doing this, the adversary could introduce an undesired *attack control error* (e_{ctrl}):

$$e_{ctrl} = e - e' \tag{4.6}$$

We call *attack control probability* (P_{ctrl}) the probability that the attack is "controlled", i.e. that the attack control error is indistinguishable from an ordinary honest-case error. More precisely, we assume a *honest-case error limit* (e_{max}), such that, in the honest case:

$$|e| \leq e_{max} \tag{4.7}$$

The attack control probability is thus defined as:

$$P_{ctrl} = \Pr\left[|e_{ctrl}| \leq e_{max}\right] \tag{4.8}$$

Enlargement attacks have been traditionally considered feasible by the literature. However, in Chapter 3 we showed that the effect of the enlargement attack is poorly controllable. In particular, the probability of controlling an enlargement depends on how much the adversary wants to enlarge. Some objective enlargements are more probable than others to succeed. As a result, some false positions will give to the adversary more success probability than others.

We distinguish two types of adversary, depending on the choice of the false position: *predefined-objective adversary* and *free-objective adversary*. The predefined-objective adversary models an adversary which is given a predefined objective and is not free of changing it. We model this by choosing X' at random among all the points having a success probability greater than zero.

The free-objective adversary models an adversary which is not given a predefined objective and is free of choosing the most convenient one. In this case, X' is chosen to be the point with the greatest success probability. The free-objective adversary is more powerful, and she has more chances to succeed.

Note that, without other constraints, a convenient choice for the false position would be very close to the true one, or coincident with it in the extreme case. With these "easy" false positions, the adversary would have a high probability to succeed in her attack. However, such an attack would not be a true spoofing, but rather a simple degradation of the system precision. We force a *minimal spoofing distance* (d_{ms}) between the false position and the true one:

$$\|X' - X\| \geq d_{ms} \tag{4.9}$$

In other words, we assume the system to be tolerant to a precision degradation of d_{ms} meters.

4.3.1 Overshadow attack

Since the distance measurement stems exclusively from the round-trip time, the adversary's aim is to enlarge it. The only way to do that is to delay the packet ToA estimates at the verifier and/or at the prover. We suppose that the adversary mounts an *overshadow attack*, in which she repeats a legitimate packet with a certain delay and a higher power. In this way, the adversary tries to "overshadow" the legitimate communication with a delayed copy of it.

More precisely, the adversary activates on the presence of a legitimate transmission. The IEEE 802.15.4a packet format comprises two parts: a *preamble* and a *payload*. The preamble is constituted by periodic pulses with a fixed structure. It is used for the estimation of the time of arrival. The payload follows instead a pulse-position modulation, and carries the unpredictable bits. First, the adversary has to synchronize with the ongoing communication. It takes some of the initial preamble symbols to do that. Then, she starts transmitting the replayed copy (skipping those initial preamble symbols). Such a replayed signal is temporized in such a way to arrive at the receiver shifted of a certain delay with respect to the legitimate one. With this action, she delays the estimate of the packet's ToA. During the successive payload phase, the adversary replays a copy of it in the same way.

We note that the possibility to *jam* the honest signals does not really help the adversary in her objective. While the jamming operation is effective in disrupting a communication, it is not a destructive operation from the PHY-layer point of view. In

fact, jamming does not impede the start of the ToA-estimation algorithm, which is triggered by a threshold on the received energy (cfr. Chapter 3). It has only the effect to perturb the result of the estimation in a random way. We suppose that the adversary avoids jamming and follows a smarter strategy, that allows her to precisely control her attack.

An adversary enjoying single-path channels toward the victim receivers is more powerful, since she can control more precisely her attack. She can obtain this either by deploying a transmitter very close to the victim receivers, or by using a highly directive antenna toward them. We distinguish three types of adversaries, depending on their capability of establishing single-path channels with the honest nodes. The *type-I adversary* is the weaker one. She does not have any single-path channel with honest nodes. The *type-II adversary* has a single-path channel with the prover, but not with the verifiers. The *type-III adversary* is the strongest one. She has a single-path channel with the prover and one with each verifier involved in the multilateration algorithm.

4.3.2 Adversarial tactics

In the following, we will use the term "*overshadow delay*" (T_o) for the timing difference between the legitimate and the adversarial signals at the victim's receiver. Furthermore, we will use the term "*obtained delay*" (T_e) for the round-trip-time enlargement obtained by the adversary with her overshadow attack. The obtained distance enlargement grows linearly with the obtained delay:

$$e = T_e \cdot \frac{c}{2} \tag{4.10}$$

Due to the ToA-estimation algorithm, the obtained delay will be different from the overshadow delay in general. It is convenient for the adversary to introduce an overshadow delay that, with high probability, will cause an outcome delay correspondent to her objective enlargement. Even better, she can overshadow both the REQ and ACK messages, introducing two (possibly different) delays T_o^{R} and T_o^{A}. The total obtained delay will be the sum of the delay obtained on REQ and the one obtained on ACK:

$$T_e = T_e^{\mathsf{R}} + T_e^{\mathsf{A}} \tag{4.11}$$

Formally, we define an *adversarial tactic* as a couple of overshadow delays:

$$\langle T_o^{\mathsf{R}}, T_o^{\mathsf{A}} \rangle$$

Also, it could be convenient for the adversary not to attack one message. So, the following ones are valid tactics too:

$$\langle \text{no-attack}, T_o^{\mathsf{A}} \rangle$$
$$\langle T_o^{\mathsf{R}}, \text{no-attack} \rangle$$

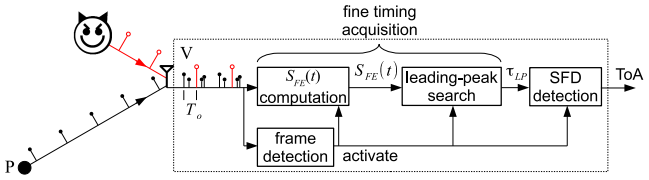


Figure 4.2. ToA estimation under attack

Given an objective enlargement, we define the *best adversarial tactic* $\langle \widehat{T}_o^{\mathsf{R}}, \widehat{T}_o^{\mathsf{A}} \rangle$ as the one which maximizes the attack control probability:

$$\langle \widehat{T}_o^{\mathsf{R}}, \widehat{T}_o^{\mathsf{A}} \rangle = \arg\max_{\langle T_o^{\mathsf{R}}, T_o^{\mathsf{A}} \rangle} (P_{ctrl}) \tag{4.12}$$

4.4 Overshadow attacks against IEEE 802.15.4a UWB

We now study the feasibility of a controlled overshadow attack against IEEE 802.15.4a UWB. The difficulty in controlling overshadow attacks is due to the way in which the legitimate and the adversarial signals get combined and processed at the receiver. This does not mean that obtaining an enlargement is impossible. Rather this means that it is obtained with a precision worse than in the honest case.

To better understand the effects of an overshadow attack, we now give some details on physical-layer procedures for threshold-based UWB ranging algorithms, which are the most widely used in UWB localization applications [29, 49, 28]. In particular, we consider the "*jump-back-search-forward*" (*JBSF*) and the "*search-back*" (*SB*) algorithms, which provide significantly different results from the security point of view. From now on, we will focus on a single overshadow against a single message. Thus, we will omit the "R" and "A" superscripts on symbols to ease the notation.

Both schemes operate in three steps (Figure 4.2). The *frame detection* step decides through energy measurements whether a packet is present or not. The *fine timing acquisition* step generates the waveform $S_{FE}(t)$ by computing a correlation with a fixed template signal. The computation of the $S_{FE}(t)$ is a classic technique that leverages on the periodicity of the modulation to improve the signal-to-noise ratio [29]. $S_{FE}(t)$ is used to perform the leading-peak search, which provides an estimate of the time of arrival of the packet. The fine timing acquisition has an ambiguity of multiples of the symbol interval, which is eliminated by the final *start-of-frame delimiter detection* step. The JBSF and SB algorithms differ only for the leading-peak search. In particular, the JBSF criterion (Figure 4.3) starts from the maximum of $S_{FE}(t)$, say τ_{HP}, jumps back by T_{JB} seconds and proceeds forward looking for the first time $S_{FE}(t)$ goes beyond a given *noise threshold* λ. The value of the noise threshold is fixed on

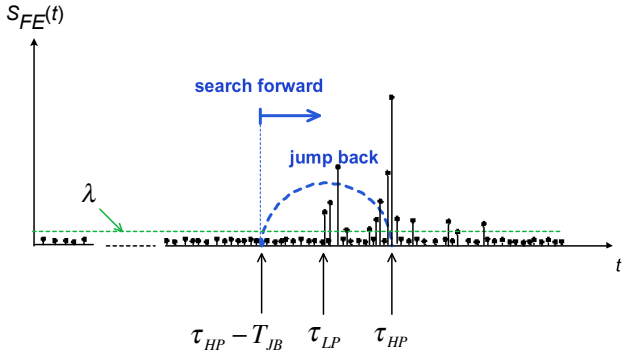


Figure 4.3. Jump-back-search-forward procedure

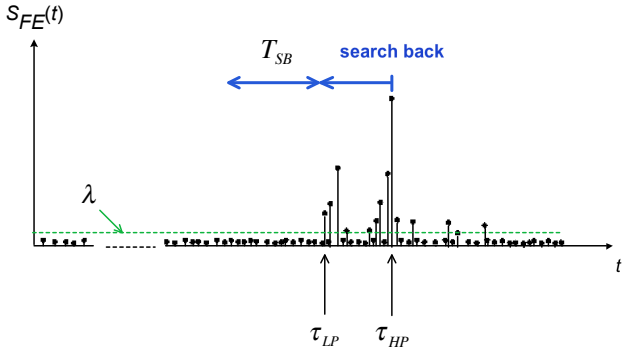


Figure 4.4. Search-back procedure

the basis of the thermal noise statistics. The distance of such a crossing time from the beginning of $S_{FE}(t)$ provides an estimate of τ_{LP}, which is the parameter we are interested in for the ranging operation. On the other hand, the SB criterion (Figure 4.4) starts from τ_{HP}, and searches backward until $S_{FE}(t)$ goes below the noise threshold and continues to be under for a time window of T_{SB} seconds. The distance of such a crossing time from the beginning of $S_{FE}(t)$ provides an estimate of τ_{LP}.

In order to determine the best adversarial tactics, we simulated the ToA-estimation algorithms described above under an overshadow attack. The honest-case error limit is supposed to be $e_{max} = 1\,\mathrm{ns} \cdot \frac{c}{2} = 15\,\mathrm{cm}$. This is because in IEEE 802.15.4a UWB the main error source is the time discretization, which has an $1\,\mathrm{ns}$-size step. We simulated a number of attacks introducing different overshadow delays over randomly generated channels, and we measured their effects[1]. Then, for each objective enlargement, we tested all the combinations of overshadow delays on the REQ and the ACK messages, selecting the one giving the highest control probability. Due to the symmetry between the REQ and the ACK transmissions, opposite tactics (e.g. $\langle 50\,\mathrm{ns}, 100\,\mathrm{ns}\rangle$ and $\langle 100\,\mathrm{ns}, 50\,\mathrm{ns}\rangle$) offer exactly the same control probability. Without

[1] The UWB channels follow the standard statistical model for a residential scenario (CM1) [70]. The signal-to-noise ratio of the honest signal is 30dB.

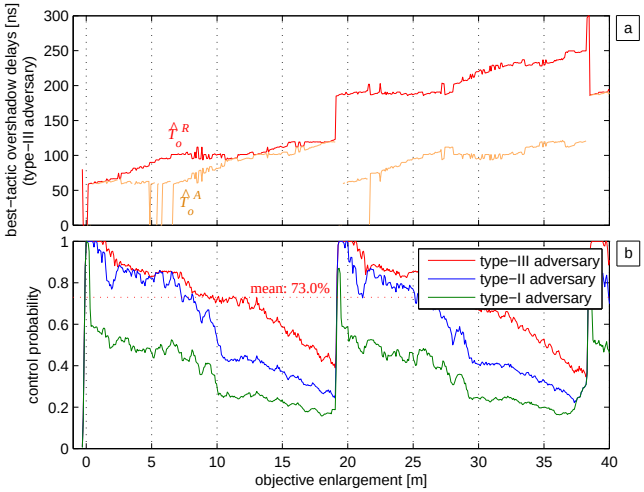


Figure 4.5. Best-tactic enlargement success probabilities against jump-back-search-forward

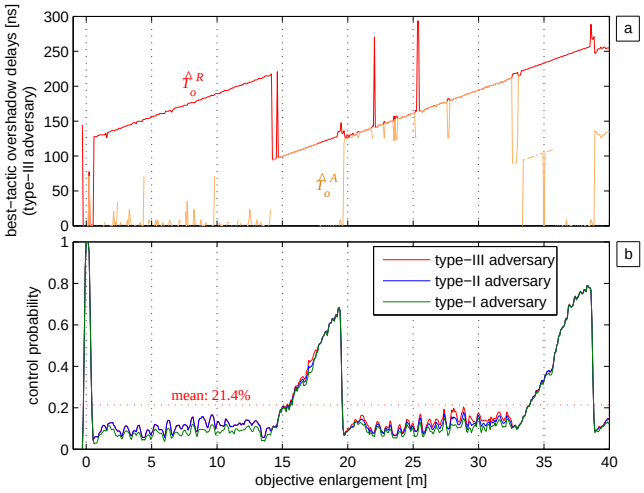


Figure 4.6. Best-tactic enlargement success probabilities against search-back

loss in generality, we thus impose:

$$\widehat{T}_o^{\mathsf{R}} \geq \widehat{T}_o^{\mathsf{A}} \tag{4.13}$$

Figures 4.5a and 4.6a show the best tactics of a type-III adversary against respectively JBSF and SB ToA-estimation algorithms. For example, in order to cause an enlargement of 10 meters against a JBSF algorithm, the best tactic is to overshadow the REQ message with a 100.4-nanosecond delay and the ACK message with a 81.7-nanosecond delay. Figures 4.5b and 4.6b show the control probabilities of such best

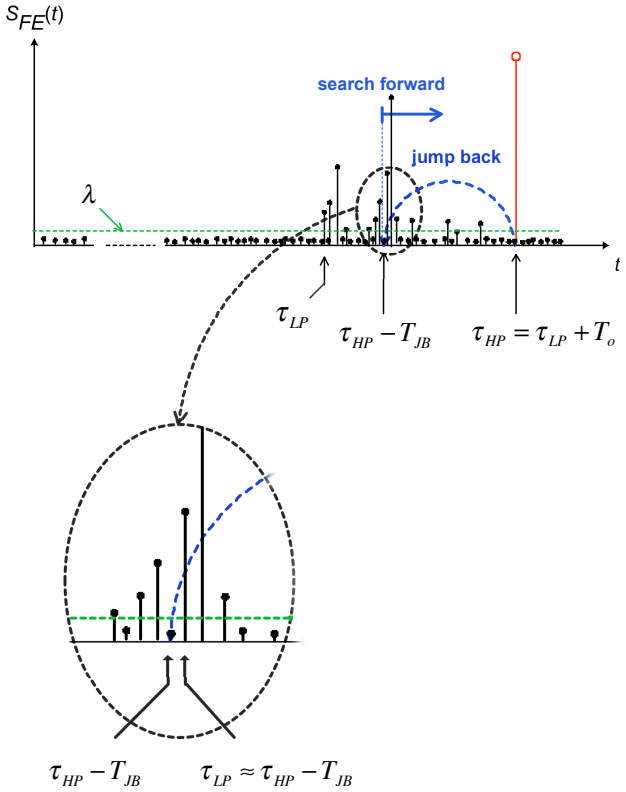


Figure 4.7. Interference attack against jump-back-search-forward procedure

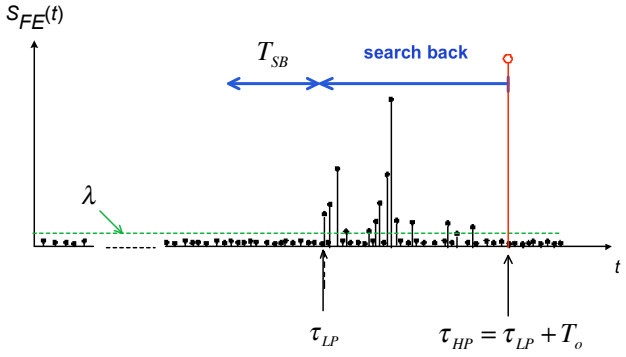


Figure 4.8. Interference attack against search-back procedure

tactics for type-I, type-II, and type-III adversaries. For example, the probability to control a 10-meter enlargement attack for a type-III adversary against a JBSF algorithm is 71.2%, whereas against a SB algorithm is 11.5%. Note that the two trends of the attack control probability are roughly periodic, following the periodicity of the preamble pulses.

As we can see from the plots, the average attack control probability of JBSF algorithm is higher than that of SB algorithm (73.0% versus 21.4% for a type-III adversary). This is due to the characteristics of the UWB propagation channel, in particular to the fact that the multipath echoes arrive at the receiver grouped into *clusters*. To explain the results of Figures 4.5b and 4.6b it is useful to analyze the effects of the overshadow attack in detail. For the sake of simplicity, we assume that the adversary is able to establish a single-path channel with the honest receiver. Anyway, the conclusions keep valid with small variations also with a multipath adversarial channel. The waveform $S_{FE}(t)$ has a component due to the legitimate signal, and a component (the strongest one) caused by the adversarial transmission. This component arrives T_o seconds after τ_{LP} (see Fig. 4.7). From Figure 4.7 it is clear that, if the delay T_o is such that the jump falls within a cluster, the noise threshold will be crossed at the very beginning of the search forward (since the echoes of a cluster are very close to each other). In this situation, the leading-peak search provides a wrong estimate of τ_{LP}, i.e. $\widehat{\tau}_{LP} \approx \tau_{HP} - T_{JB}$. Thus, the adversary obtains an enlargement delay approximately equal to $T_o - T_{JB}$ which is controllable since T_o and T_{JB} are respectively a parameter chosen by the adversary and a well-known system parameter. Obviously, the attack control probability depends on the objective enlargement. Normally the echoes are grouped at the beginning of the channel response, thus, small enlargements have greater control probability (see Fig. 4.5b).

The same does not occur with the SB algorithm, as shown in Figure 4.8. Indeed, differently from JBSF, with the SB scheme the enlargement attack is successful only if the legitimate echoes are sufficiently sparse. This condition occurs with higher probability at the end of the channel response. Accordingly, as shown in Figure 4.6b, the control probability takes significant values only for objective enlargements greater than 15 meters. Given its better properties in terms of security, we will employ exclusively the SB algorithm on the receivers of our positioning system.

4.5 EMCD-ML

We introduce EMCD-ML (*Enlargement MisControl Detection MultiLateration*), a range-based secure positioning system leveraging on the difficulty of the adversary to control the effect of the enlargement attacks. EMCD-ML is based on distance-bounding operations on the IEEE 802.15.4a UWB protocol. The basic idea is that a position measured in presence of an attacker will have a lower precision than one measured in a honest scenario. So we can detect the presence of an attacker by means of the residuals, which have large values in case of low precision.

In the honest case, a distance measured by a ranging operation will be the real distance plus a measurement error:

$$\widehat{d} = d + e \tag{4.14}$$

In the presence of an attacker, it will be the objective distance plus an attack control error:

$$\widehat{d} = d' + e_{ctrl} \tag{4.15}$$

Due to the difficulty for the adversary to control precisely the outcome of her attack, the expected value of e_{ctrl} is greater than that of e. Stemming from this, in case of attack the least-squared-error solution of Equation 4.2 will probably produce high residuals. To detect the presence of an attack, it is sufficient to discard those positions giving residuals beyond a certain *acceptance threshold* (ϵ_{max}). The number of verifiers involved in the multilateration affects the security of the system. Indeed, the more verifiers are involved, the more enlargement attacks the adversary has to control, and thus the lower probability she has to succeed. EMCD-ML establishes a *minimal number of verifiers* (N_{min}) to assure a given level of security.

Note that, although it is similar in the form, this countermeasure has an opposite objective from verifiable multilateration [100]. Also in verifiable multilateration the residuals are compared to a threshold and the measured position is discarded if they are higher. However, in verifiable multilateration the residuals are large because the adversary *cannot perform reduction attacks*. In EMCD-ML the residuals are large because the adversary *cannot control enlargement attacks*. This allows us to accept honest positions also outside the polygon formed by the verifiers, that would be rejected by verifiable multilateration. As a result, EMCD-ML can cover the same area with far less verifiers, and avoids the need of deploying additional verifiers at the borders of the deployment area.

Figure 4.9 shows a first version of the EMCD-ML algorithm. The discovery of the reachable verifiers is done with an insecure beacon-based method (Figure 4.9, Line 2). We accept positions which meet the *in-polygon check* condition of verifiable multilateration [100] (Line 9). We accept also positions measured by means of at least N_{min} verifiers and whose residuals are lower than ϵ_{max} (Line 17). We use the notation $\epsilon_{max}(N)$ because the value of the threshold depends on the number of involved verifiers.

4.5.1 Improving EMCD-ML by ranging repetition

It is possible to additionally improve the security of EMCD-ML by repeating each ranging operation k times. We will thus obtain k distance measurements for each verifier V_i: $\widehat{d}_i^{(1)}, \widehat{d}_i^{(2)}, \ldots, \widehat{d}_i^{(k)}$. The repetition of the ranging operations has two consequences in EMCD-ML. The first one is that the adversary has to carry on k overshadow attacks. Even if the adversary introduces the same delays and the channels do not change sensibly, each attack can cause quite a different enlargement. If the measured distances have an anomalous variation, the system can detect the attack.

The second consequence is that, by averaging the $\widehat{d}^{(j)}$'s, we obtain smaller errors in the honest case, and thus smaller residuals. This allows us to set a stricter

1: **procedure** EMCD-ML(N_{min})
2: prover sends a broadcast beacon
3: $\mathcal{V} \leftarrow \{\mathrm{V}_i$ that answered the beacon$\}$
4: **for all** $\mathrm{V}_i \in \mathcal{V}$ **do**
5: perform distance bounding
6: $\widehat{d}_i \leftarrow$ measured distance
7: **end for**
8: $\langle \widehat{X}, \epsilon_i \rangle \leftarrow \mathrm{multilaterate}(\mathcal{V}, \widehat{d}_1, \widehat{d}_2, \dots)$
9: **if** $\widehat{X} \in \mathrm{polygon}(\mathcal{V})$ **and** $\forall i\ \epsilon_i \leq \epsilon_{max}(|\mathcal{V}|)$ **then**
10: **return** $\widehat{X}$
11: **end if**
12: $\mathcal{SV} \leftarrow \left\{ \mathrm{V}_i \in \mathcal{V} : \widehat{d}_i \leq d_{safe} \right\}$
13: **if** $|\mathcal{SV}| < N_{min}$ **then**
14: **return** "insufficient security!"
15: **end if**
16: $\langle \widehat{X}, \epsilon_i \rangle \leftarrow \mathrm{multilaterate}(\mathcal{SV}, \widehat{d}_1, \widehat{d}_2, \dots)$
17: **if** $\exists i\ \epsilon_i > \epsilon_{max}(|\mathcal{SV}|)$ **then**
18: **return** "overshadow attack detected!"
19: **end if**
20: **return** $\widehat{X}$
21: **end procedure**

Figure 4.9. EMCD-ML algorithm (first version)

acceptance threshold on the residuals, thus achieving a better probability of attack detection. Note that the adversary does not increase her precision in the same way. In fact, the attack control error is not zero-mean, because it depends mainly on the channel response, that does not change sensibly from a repetition to another (a single IEEE 802.15.4a ranging takes about 20 milliseconds [55]).

Figure 4.10 shows the improved version of EMCD-ML algorithm. The ranging repetitions are performed on Line 5. The check on the anomalous variation of the measured distances is performed on Line 7. Now, we use the notation $\epsilon_{max}(N, k)$ because the value of the threshold depends on the number of involved verifiers and on the number of ranging repetitions. This version of EMCD-ML has a better resistance in an adversarial scenario and a better precision in a honest one, but consumes more energy since each ranging operation must be repeated k times.

4.6 Experimental evaluation

We evaluated the parameters and the properties of EMCD-ML by means of simulations. Namely, we determined the residuals' acceptance thresholds, the security level, and the number of necessary verifiers to cover an area.

1: **procedure** EMCD-ML(N_{min}, k)
2: prover sends a broadcast beacon
3: $\mathcal{V} \leftarrow \{\mathrm{V}_i$ that answered the beacon$\}$
4: **for all** $\mathrm{V}_i \in \mathcal{V}$ **do**
5: perform distance bounding $\times k$ times
6: $\widehat{d}_i^{(1)}, \ldots, \widehat{d}_i^{(k)} \leftarrow$ measured distances
7: **if** $\max_j(\widehat{d}_i^{(j)}) - \min_j(\widehat{d}_i^{(j)}) > 2e_{max}$ **then**
8: **return** "overshadow attack detected!"
9: **end if**
10: $\widehat{d}_i \leftarrow \frac{1}{k}\sum_j \widehat{d}_i^{(j)}$
11: **end for**
12: $\langle \widehat{X}, \epsilon_i \rangle \leftarrow \mathrm{multilaterate}(\mathcal{V}, \widehat{d}_1, \widehat{d}_2, \ldots)$
13: **if** $\widehat{X} \in \mathrm{polygon}(\mathcal{V})$ **and** $\forall i\ \epsilon_i \leq \epsilon_{max}(|\mathcal{V}|, k)$ **then**
14: **return** $\widehat{X}$
15: **end if**
16: $\mathcal{SV} \leftarrow \left\{\mathrm{V}_i \in \mathcal{V} : \widehat{d}_i \leq d_{safe}\right\}$
17: **if** $|\mathcal{SV}| < N_{min}$ **then**
18: **return** "insufficient security!"
19: **end if**
20: $\langle \widehat{X}, \epsilon_i \rangle \leftarrow \mathrm{multilaterate}(\mathcal{SV}, \widehat{d}_1, \widehat{d}_2, \ldots)$
21: **if** $\exists i\ \epsilon_i > \epsilon_{max}(|\mathcal{SV}|, k)$ **then**
22: **return** "overshadow attack detected!"
23: **end if**
24: **return** $\widehat{X}$
25: **end procedure**

Figure 4.10. EMCD-ML algorithm (improved version)

4.6.1 Residuals' acceptance threshold

In order to tailor the residuals' threshold ϵ_{max}, we simulated a number of multilaterations with IEEE 802.15.4a ranging. The positions of the prover and the verifiers are taken at random. We tailored the acceptance thresholds in such a way to accept 99.9% of the honest-case position measurements. Table 4.2 shows the thresholds with respect to the number of verifiers and the number of ranging repetitions[2]. We can see that by repeating the ranging operations we can lower the threshold, while maintaining the rate of accepted honest positions. As a side effect, the ranging repetition improves the positioning precision also, as shown in Figure 4.11.

4.6.2 Security level

The success probability of the attack decreases with the growing of the verifiers involved in the multilateration, and with the growing of the ranging repetitions. In order

[2] Each estimation comes from 100,000 Monte Carlo runs. $e_{max} = 15\,\mathrm{cm}$. $d_{safe} = 40\,\mathrm{m}$. 99%-confidence intervals are within $-2.8\,\mathrm{mm}$ and $+3.0\,\mathrm{mm}$ for all the thresholds.

ranging repetitions:	number of verifiers: 3	4	5	6
1	13.17cm	14.93cm	15.81cm	16.45cm
2	10.02cm	11.29cm	12.17cm	12.50cm
4	7.24cm	8.29cm	8.83cm	9.16cm
8	5.25cm	5.94cm	6.43cm	6.63cm
16	3.69cm	4.22cm	4.55cm	4.74cm
32	2.58cm	2.98cm	3.23cm	3.37cm

Table 4.2. EMCD-ML residuals' thresholds

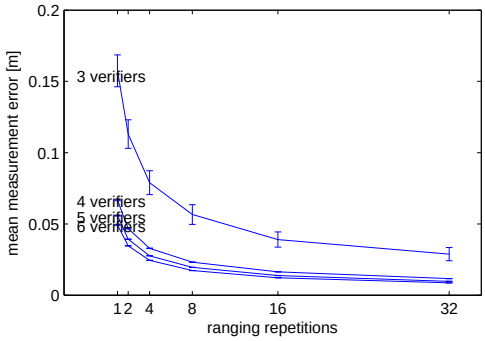


Figure 4.11. Precision of EMCD-ML

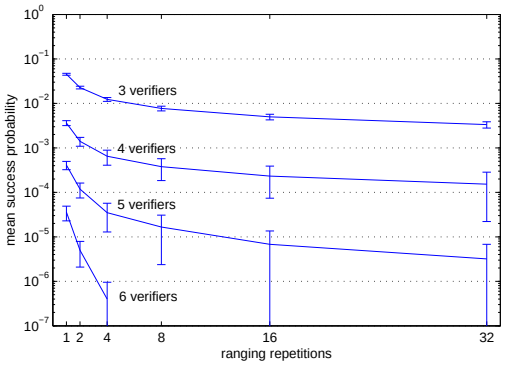


Figure 4.12. Success probability of type-III predefined-objective adversary

to quantify the attack success probability, we simulated a type-III predefined-objective adversary and a type-III free-objective adversary. The verifiers' as well as the prover's positions are taken at random. Figure 4.12 shows the success probability of a type-III predefined-objective adversary, with respect to the number of verifiers covering the prover's position and the number of ranging repetitions at each verifier[3].

[3] Each estimation comes from 5,000 Monte Carlo runs, $d_{safe} = 40\,\mathrm{m}$, $d_{ms} = 1\,\mathrm{m}$. 99%-confidence intervals are displayed.

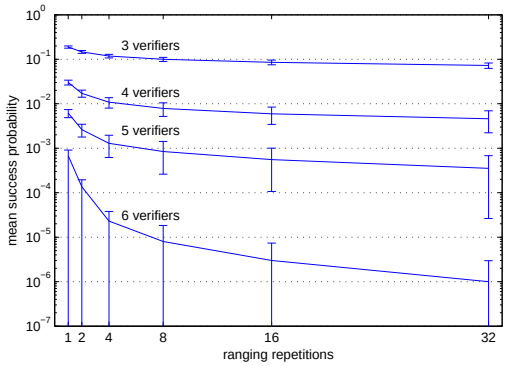


Figure 4.13. Success probability of type-III free-objective adversary

success probability:	min number of verifiers:		min ranging operations:	
$< 10^{-1}$	$N_{min} = 3$	$k = 16$	$N_{min} = 4$	$k = 1$
$< 10^{-2}$	$N_{min} = 4$	$k = 8$	$N_{min} = 5$	$k = 1$
$< 10^{-3}$	$N_{min} = 5$	$k = 16$	$N_{min} = 6$	$k = 1$
$< 10^{-4}$	$N_{min} = 6$	$k = 4$	$N_{min} = 6$	$k = 4$
$< 10^{-5}$	$N_{min} = 6$	$k = 16$	$N_{min} = 6$	$k = 16$

Table 4.3. EMCD-ML configuration parameters

Figure 4.13 shows the success probability of a type-III free-objective adversary with respect to the number of verifiers covering the prover's position, and the number of ranging repetitions at each verifier[4].

From this analysis, it is possible to tailor the parameters of EMCD-ML (N_{min} and k) in order to assure a given level of security. Table 4.3 shows possible configurations to offer a given level of security against a type-III free-objective adversary. They are chosen to minimize either the necessary verifiers (i.e. N_{min}), or the total number of ranging operations ($N_{min} \cdot k$).

4.6.3 Number of necessary verifiers

EMCD-ML permits us to cover the same area with less anchor nodes with respect to verifiable multilateration, while maintaining a high level of security. From the algorithm in Figure 4.10 we can see that, given a set of verifiers $\{\mathrm{V}_i\}$, a point X is covered by EMCD-ML iff one of the following conditions is true: (1) there exist at least three verifiers within the communication range, and the triangle formed by them contains X (enlargement presence detection); or (2) there exist at least N_{min} verifiers within d_{safe} distance (enlargement miscontrol detection). The first condition is present

[4] Each estimation comes from 1,000 Monte Carlo runs. $d_{safe} = 40\,\mathrm{m}$, $d_{ms} = 1\,\mathrm{m}$. 99%-confidence intervals are displayed.

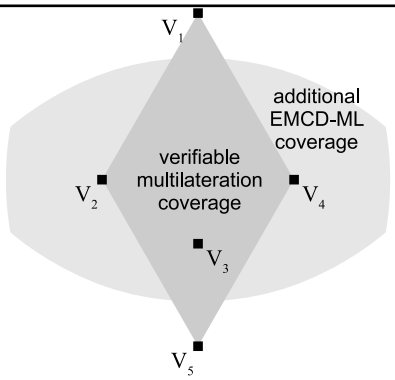


Figure 4.14. Coverage area comparison

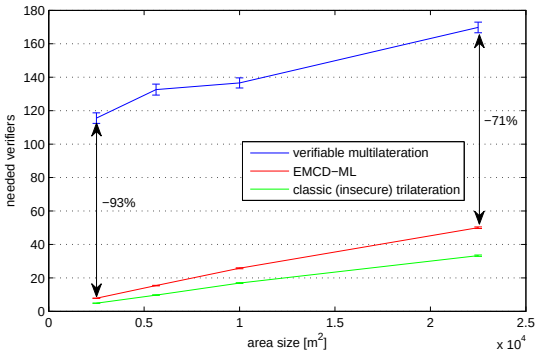


Figure 4.15. Needed verifiers for covering 90% of the area

also in verifiable multilateration [100]. The second condition gives additional coverage. Figure 4.14 gives an example of this. The additional coverage lowers the number of verifier nodes to deploy in order to cover a given area. Figure 4.15 shows the number of verifiers needed to cover 90% of a square area[5]. The verifiers are deployed at random. We supposed $N_{min} = 5$, in such a way to reach an adversarial success probability of 10^{-3}. We also supposed that d_{safe} is as long as the communication range. It can be seen that EMCD-ML greatly improves the anchor-node scalability, and it gets very near to the theoretical limit of the classic (insecure) trilateration. This big difference (-93% to -71% of needed verifiers) is also due to the fact that verifiable multilateration cannot cover outside the polygon formed by the verifiers. Thus covering the zones at the border of the deployment area is quite hard. To solve this problem, in [100] the authors use a special deployment scheme. In this scheme, the verifiers are randomly deployed in the area and also in an external band with width equal to the communication radius. The verifiers are randomly deployed in the area to

[5] Each estimation comes from 1,000 Monte Carlo runs. $d_{safe} = 40\,\mathrm{m}$. 99%-confidence intervals are displayed.

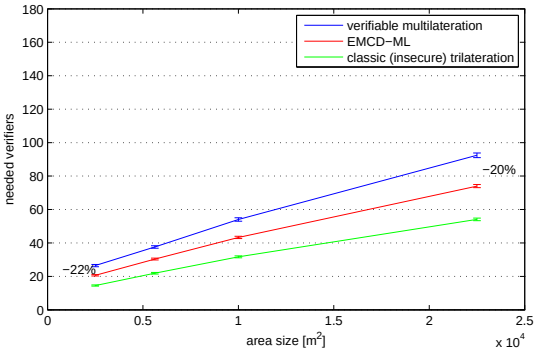


Figure 4.16. Needed verifiers for covering 90% of the area (external deployment)

cover and in the external band. Figure 4.16 shows the number of randomly deployed verifiers needed to cover 90% of a square area with this deployment scheme. It can be seen that EMCD-ML improves the anchor node scalability also with this deployment scheme which is ad-hoc for verifiable multilateration (-22% to -20% of needed verifiers).

5

Integration of privacy protection mechanisms in location-based services

Due to the proliferation of tracking technologies, like GPS, the interest in location-based services (LBSs) is growing fast. Nowadays, a plethora of technologies are capable of localizing people: palm GPS, RFID, video cameras, and so on. Recent studies [98] showed that users feel that the privacy risks of using LBSs still outweigh the benefits. Therefore, in order to be accepted by users, LBSs must be trusted on the standpoint of privacy [3, 10]. Protecting privacy in an LBS is not an easy task. Privacy policies need be flexible, integrated, customizable and context-aware [12, 23, 97, 98]. Current solutions [40, 42, 61, 66, 97] focus on *access control* approach, i.e. the system decides which information will be released and which not, basing on some authorization rules. Such rules are *context-aware*, meaning that they take into account the current date and time, the location of the user, and the situation of the user or the system itself (e.g. the presence of alarms in a particular building, etc.). Though access control is essential, the "permit-or-deny" logic behind it forces the users to choose from having the service or having the privacy. Recent years have seen the emergence of more flexible mechanisms [8, 33, 36, 48, 68]. Many applications prefer *anonymization approaches*, in which the identity of the user is detached from the information. Many others prefer *obfuscation approaches*, in which the system artificially degrades the precision of information before releasing it. In this way, users can tailor their own trade-off between privacy and quality of service.

However, this may not be enough. For many services, a single mechanism is not sufficient to meet the privacy requirements, which can be instead fulfilled only by a proper *integration* of different protection mechanisms. Think about a simple find-the-nearest-restaurant service. The user gives her position to a service provider in order to get the name of the nearest restaurant which meets some requirements. How can we protect the privacy of the user? Her identity is unnecessary, so the location information could be anonymized. Also, the service provider does not need the exact position, so an obfuscation algorithm could be applied.

The contribution of this chapter is twofold. First, we present LbSprint (Location-Based Service PRivacy INTegrator), a middleware layer for privacy protection in

location-based services. LbSprint supports multiple privacy-protection mechanisms, allows system administrators to define new ones, and allows users to setup flexible and context-aware privacy policies. LbSprint implements these features by means of the XACML language capabilities [73]. Secondly, we present optimizations that considerably improves the performance of the XACML policy evaluation. These contributions apply to LBSs as well as on privacy-protection architectures in general. This chapter has been published as a conference paper [34].

The rest of the chapter is organized as follows. Section 5.1 presents some related works, and underlines the differences from LbSprint. Section 5.2 explains the system requirements and presents a typical use case. Section 5.3 includes a detailed description of the system architecture and the privacy protection module. Section 5.4 analyzes the performance of the system and describes some optimizations to improve it.

5.1 Comparison of LbSprint with the state of the art

To the best of our knowledge, the problem of privacy mechanism integration in LBSs has not yet been addressed by research or industry.

Commercial LBSs have a poor support for privacy protection. See [98] for a complete survey. The most known examples are maybe Foursquare [42], Loopt [66] and Google's Latitude [61]. Recent facts suggest that they will move towards a better protection of users' privacy, especially after the diffusion of so-called "stalker apps" [5].

Yahoo!'s Fire Eagle [40] has been one of the first commercial LBS platforms posing particular attention to location privacy. Users are capable of specifying relatively complex privacy policies in terms of who can access their locations and when. It gives also support for obfuscation. Locations can be specified with several degrees of granularity (exact position, ZIP code, neighborhood, city, etc.). However, it does not offer the possibility to integrate different privacy protection mechanisms and to define new ones.

Locaccino [97] is a privacy-centric application for location sharing, based on the Facebook platform. Its features came from some surveys the authors did to investigate privacy preferences of people. Users can create simple policies specifying who, when and where can see their location. Locaccino is focused on access control, and gives to the users the option to permit or deny the access to their location. It poses little focus on integration between privacy mechanisms. The authors chose not to include any obfuscation mechanism, because surveys [23] showed that people do not use it to protect their privacy. However, the obfuscation method investigated in such surveys was quite an inflexible one. It gave the user the possibility to release either her exact position, or the position with city-level granularity. In LbSprint, we take into account more flexible obfuscation methods, that offer finer granularities. These methods are suitable for a plethora of location-based services, as we will show in Section 5.2.

From the research world, one of the first privacy-centric tracking systems has been LocServ [72]. In LocServ, location-based applications make their location requests and specify the privacy policies they will adopt in using such information. The users provide for location data and specify their own privacy preferences, represented by components called *validators*. Before releasing information to an application, LocServ asks the correspondent validator whether the policy of the application is compatible with the user's preferences. A validator, during its decisional process, can consult the user or other (possibly external) subordinate validators.

A related problem is the reliable communication of privacy policies [60]. Geopriv is a standard developed by IETF, aimed at the representation and transmission of location data [24]. The basic idea is that the privacy policies and the location data are encapsulated in a unique entity, the *location object*. So they are always transmitted together. The integrity of the location object is guaranteed by a digital signature. If Alice wants to share her position, she will define a privacy policy, which specifies how her location must be used and distributed. Each user able to read Alice's position is himself aware of the rules stated by Alice. Geopriv does not really forbid other users to share Alice's information. It just ensures that, if someone breaks a rule, he cannot claim he was unaware of it. LbSprint focuses on the orthogonal problem of privacy mechanism configurability and extensibility. Geopriv mechanisms are utilizable in LbSprint as well.

5.2 The case for integration of privacy protection mechanisms

The following is a use-case story which illustrates several location-based services in an example scenario: an airport. Each service needs a customized mix of privacy protection mechanisms.

> John has to take a flight together with his little son, Tim. They reach the airport and they get to the check-in area. Before leaving his luggage, John subscribes to the *track-my-luggage* service. Such a service tracks the position of John's baggage for security purposes. See [26] for an example of this. John is informed about it whenever he wants through his smartphone.
>
> Once they checked-in, John and Tim want to have lunch. John prefers vegetarian food but he does not have time for searching a vegetarian restaurant. So, he uses the *find-the-nearest-restaurant* service, which finds the nearest vegetarian restaurant. After lunch, Tim wants to visit a toy shop he saw before. John lets him go, but he wants to track him position, because he does not want him to get too far. To do this, John uses the *track-a-child* service. After a while, John realizes that the boarding time is soon. He quickly reaches Tim, because he knows his position, and they easily get to the gate.
>
> For all the story time, John and Tim had used several other location-based services. For example the *track-for-safety* service, which tracks John and

Tim's positions in order to help rescuers in case of fire, and the *find-my-friends* service, a social service that informs John about the proximity of friends.

Also, the *track-the-employees* service is active on the airport. Such a service allows an operator to track the position of the personnel for management purposes.

Each of the above services requires a different mix of mechanisms for the protection of the privacy. Track-my-luggage needs only authorization, since only John is allowed to know the location of his baggage. Permitting anyone else could be harmful for the security (e.g. luggage stealing). Find-the-nearest-restaurant does not need to know the exact position or the identity of John. Thus, John's position is obfuscated and John's identity anonymized. The service provider is allowed to receive such a position, other entities are not. Track-a-child requires authorization, as only John is permitted to access the location of his son. However, a parent wants to know its children's position very precisely. So no obfuscation method is applied. Track-for-safety requires authorization, because only rescuers can access passengers' locations, but no anonymization neither obfuscation, as they could hinder the emergency operations. However, the passengers' locations can be accessed only in a particular context, i.e. in case of a fire alarm. Find-my-friends needs to know John's identity and those of his friends, but does not need their exact positions. Thus, John can decide whether to obfuscate or not his own position. The service provider (i.e. the social network platform) is allowed to receive such a position, other entities are not. Finally, track-the-employees needs authorization, as only the personnel manager can access employees' positions and only during the working hours, and obfuscation. The current time and the presence of alarms are examples of context attributes. Table 5.1 summarizes the privacy mechanisms applied for each service. The "authorized receivers" column lists who is allowed to receive location information. The "anonymization" and "obfuscation" columns tells us whether the locations will be respectively anonymized and obfuscated.

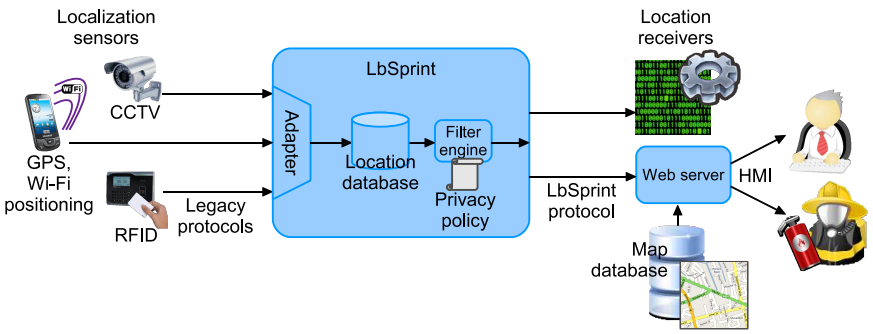


Figure 5.1. LbSprint architecture

Service type:	Authorized receivers:	Anonymized:	Obfuscated:
Track-my-luggage	the owner	–	–
Find-the-nearest-restaurant	the service provider	yes	yes
Track-a-child	the parents	–	–
Track-for-safety	the safety operators *context:* fire alarm on	–	–
Find-my-friends	the service provider	–	optional
Track-the-employees	the personnel manager *context:* working hours	–	yes

Table 5.1. Summary of the privacy policies in use-case story

It follows that an effective location-based service middleware must support different mechanisms for privacy protection as well as different ways of integrating them.

5.3 Architecture and implementation

Figure 5.2. LbSprint human-machine interface

LbSprint follows a centralized architecture, as shown in Figure 5.1. Every *location receiver* first authenticates to LbSprint, and then asks for and receives location notifications. All communications are secure in terms of confidentiality, authenticity and integrity. Different receivers have different rights for accessing location data, as summarized in Table 5.1. LbSprint supports various localization technologies: handheld GPS and Wi-Fi positioning, RFID, CCTV (Close-Circuit TeleVision) [45]. The localization sensors generate streams of raw location measurements, and send them to LbSprint through a plethora of legacy protocols. Some sensors (GPS, Wi-Fi positioning, RFID) provide for the identity of tracked users too. Some other (CCTV) does not, so that

location measurements are associated to pseudonyms (e.g. urn:lbsprint:user001). An adapter module converts the different formats of location measurements in a common format (*location data*). A *filter engine* manipulates it, basing on a *privacy policy*. Finally, the location data stream is forwarded to the receiver by means of the LbSprint protocol. LbSprint provides also for a human-machine interface (HMI), which visualizes the location data on a map. The HMI is a web application which authenticates as a normal receiver. Figure 5.2 shows a screenshot of it, taken from a practical implementation in a company.

The LbSprint protocol is built over SOAP [104]. Through the entire architecture, the location data is represented by an XML structure called `Location` (Figure 5.3). A `Location` contains the position along with possible other meta-data, such as the identity of the user, the sensor ID and type, and so on.

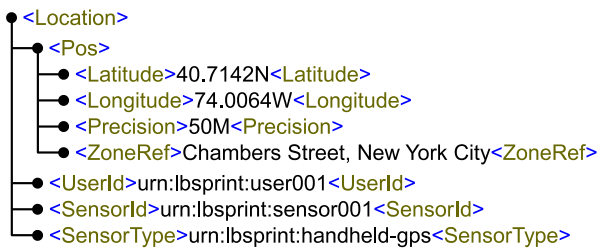


Figure 5.3. Example of `Location`

5.3.1 Filter engine

LbSprint uses the filter engine to protect the privacy of the users. The filter engine's privacy policy is configurable by the system administrator and by the users. The privacy policy takes into account many factors, including the context, and specifies different privacy mechanisms, e.g. authorization, anonymization and obfuscation.

Figure 5.4 shows how the filter engine works. Before LbSprint sends a `Location` A to a location receiver, it passes A to the *policy enforcement point* (PEP). The PEP makes a *filter strategy request* to the *policy decision point* (PDP). The PDP is a module which interprets the privacy policy and decides which manipulations must be applied on A (*filter strategy*). The PEP applies the specified manipulations and releases A' as output. A filter strategy could force a *complete filtering*, that is the location data is not disclosed at all ($A' = \mathrm{null}$). This corresponds to a denial of access. On the other hand, a *no filtering* strategy releases the location data "as-is", without manipulations ($A' = A$). Other filter strategies could apply one or more manipulation algorithms on A, in order to anonymize it, obfuscate it, etc. Subsection 5.3.3 will show how users can specify and configure such filter strategies.

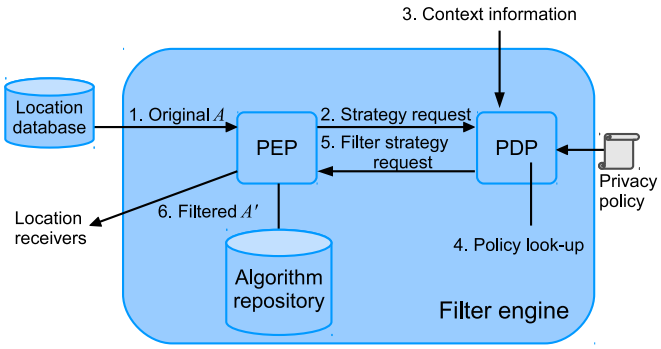


Figure 5.4. Workflow of the filter engine

5.3.2 Obfuscation methods

As a proof of concept, we provided LbSprint with two obfuscation mechanisms: *generalization* and *UniLO* (cfr. Chapter 6). They both replace an exact position with a *location area*, which contains such a position but has a larger extension. The larger is the location area, the less precise is the obfuscated location.

Generalization is based on a hierarchical description of the map (*generalization tree*), which defines zones with different coarseness. For example, a generalization tree applicable to a company could be the following: *Exact position* $\rightarrow$ *Room* $\rightarrow$ *Sector* $\rightarrow$ *Building* $\rightarrow$ *Entire system*. Generalization takes a parameter k and replaces the exact position with a zone which is k levels away from the bottom of the generalization tree. For instance, $k = 0$ corresponds to returning the exact position of the user (*Exact position*), whereas $k = 3$ corresponds to returning the building in which the user currently is (*Building*).

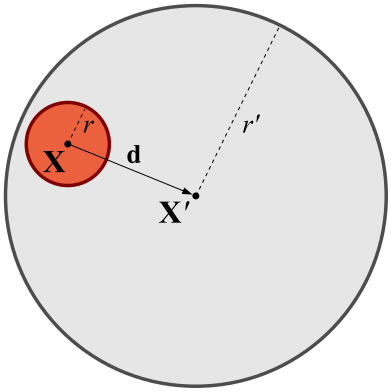


Figure 5.5. Perturbation example

On the other hand, UniLO aims at generating circular location areas. The original location data carries an indication of the *precision radius* of the sensor. For example, $r = 5\,\mathrm{m}$. UniLO takes a parameter $r' > r$, for example $r' = 25\,\mathrm{m}$, and adds noise to artificially increase the precision radius to r'. The noise is a *random vector*, which is added to the original coordinates. Figure 5.5 shows an example of UniLO application. $\mathbf{X}$ and r are respectively the original position and precision radius, whereas $\mathbf{X}'$ and $\mathbf{d}$ are the perturbed position and the random vector. The circle centered at $\mathbf{X}'$ and with radius r' is the resulting location area. If the location is asked again, and the user is still inside the same location area, the same location area is returned. This is in order to avoid that someone intersects many location areas to find the true location of the user. The random vector has a specific probability distribution, in order to generate obfuscated locations as uniform as possible from the probabilistic standpoint. See Chapter 6 for a detailed description of UniLO.

Generalization and UniLO cannot be applied together. Since no method is better than the other, the policy writer must choose the most suitable method for each case. Generalization is probably more intuitive. Nevertheless, it requires a hierarchical specification of the map, which could be missing. On the other hand, UniLO is less intuitive, but generates circular location areas. Circles have simpler shapes than rooms or buildings, and they are easier to be processed by geometry-based algorithms.

We implemented two simple obfuscation algorithms and a simple anonymization algorithm only to demonstrate the integration capabilities of the system. Many other mechanisms exist [14], aimed at anonymity [44, 48, 67] or data obfuscation [8, 27, 36], which could be integrated as well. The analysis of these techniques falls outside the scope of this chapter.

5.3.3 Privacy policies

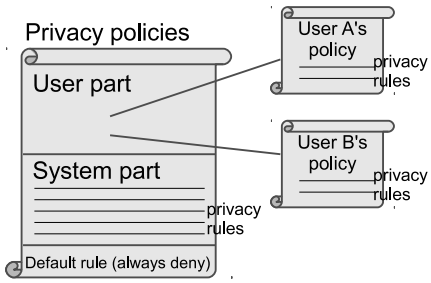


Figure 5.6. Privacy policy structure

A privacy policy comprises two parts: a *user part* and a *system part* (Figure 5.6). The user part contains policies specific to single users. These can be written by the users themselves or by the system administrator. The system part is written by the

system administrator. The PDP tries first to apply the user part, in particular the policy of the user which location data refers to. If a user did not write her own policy, or her policy does not apply to the case, the system part will be applied. If neither the system part is applicable, a complete-filtering default rule will be applied. The system administrator can install a new privacy protection mechanism. For example a new obfuscation algorithm. In such a case, she must define the URI of the algorithm (e.g. urn:lbsprint:my-obfuscation), its parameters, and finally provide a Java class implementing the new algorithm. This makes our middleware extensible and configurable.

The receiver, the location, and the context are represented as collections of *attributes*. Each attribute is a name-value pair. Each *privacy policy* is a list of *privacy rules*. A *simple* privacy rule is composed of a *target* and a *filter strategy*. The target is a boolean expression on (the attributes of) the location receiver, the location data, and the context. It decides whether the rule is applicable or not. If it is applicable, the corresponding filter strategy will be applied on location data. Otherwise, the next rule will be evaluated, and so on. To fix the ideas, in the use-case story of Section 5.2, a user could specify the following rule for the find-the-nearest-restaurant service:

> "if the location receiver is *find-the-nearest-restaurant*, then apply anonymization and UniLO obfuscation with $r' = 50\,\mathrm{m}$"

The "if" part represents the target, the "then" part represents the filter strategy.

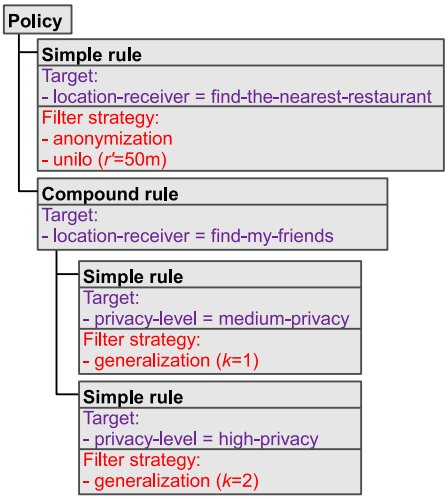


Figure 5.7. Example of policy tree

In such a way, complex policies are represented by long lists of rules. Such a "flat" schema is simple, but it is not modular. For example we may need to group all the rules referring to a particular service type, in such a way to write more compact, modular, readable policies. In addition, tree-shaped policies outperform flat ones, as shown in

Section 5.4. For these reasons, LbSprint allows the administrator to define *compound* privacy rules. Compound rules contain no filter strategy, instead they have a target and a list of *sub-rules*, each of them can be both simple or compound. The list of sub-rules cannot be empty. A compound rule is applicable only if its target evaluates to true and at least one of its sub-rules is applicable. The filter strategy of the applicable sub-rule will be the filter strategy of the entire compound rule. Compound rules allows us to define tree-shaped policies, which perform better in terms of processing time. An example of policy could be the following:

> "if the location receiver is *find-my-friends*, then: if the privacy level is medium, then apply a generalization of 1 level, otherwise if the privacy level is high, the apply a generalization of 2 levels"

Figure 5.7 shows an example of policy, with a simple rule and a compound one.

Filter strategy:	Final effect:	Obligation:	Parameter:
No filtering	Permit	-	-
Complete filtering	Deny	-	-
Anonymization	Permit	urn:lbsprint:anonymize	-
Generalization	Permit	urn:lbsprint:generalize	level (k)
UniLO	Permit	urn:lbsprint:unilo	radius (r' in meters)

Table 5.2. Filter strategies

The privacy policies are written in the XACML language [73]. XACML (eXtensible Access Control Markup Language) is an XML dialect standardized by the OASIS consortium. It expresses extremely flexible rules for access control. We used SunXACML, the off-the-shelf PDP provided by Sun Microsystems, written in the Java language [92]. The XACML language allows every authorization decision to have only two valid outcomes: "Permit" or "Deny". In order to specify more complex filter strategies, we used the XACML `<Obligation>` tag. Figure 5.8 shows the find-the-nearest-restaurant simple rule of Figure 5.7, expressed in the XACML language. The attributes of the location receiver, location data and context are referred by URIs, along with the manipulation algorithms. The `<PolicySet>` tag represents either the entire privacy policy, or a compound rule. A simple rule is represented by the `<Policy>` tag. A target is represented by the `<Target>` tag. A filter strategy is represented by the `Effect` XML attribute and the `<Obligation>` tag. An `Effect` equal to `Deny` corresponds to a complete filtering. The `<Obligation>` tags are used in XACML to dictate additional duties that the PEP must fulfill before granting the authorization. For example sending a notification email to the system administrator. We use such tags to specify (mixes of) manipulation algorithms to be applied on location data. The algorithms are applied in the same order of the corresponding `<Obligation>` tags. XACML allows us to specify the parameters of an obligation, by means of `<AttributeAssignment>` tags. We use them as the parameters of the algorithm.

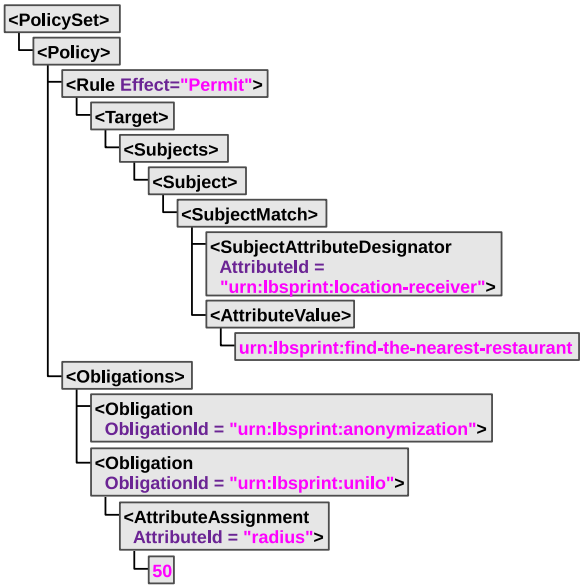


Figure 5.8. A simple rule in the XACML language

Table 5.2 tells us how basic filter strategies are expressed in terms of final effect, obligations and parameters. Mixed filter strategies, like “anonymization and generalization” are expressed with the union of the relative obligations, as shown in Figure 5.8. In such a case, the manipulations are applied in order of appearance.

Note that our integration method can be seamlessly adapted to XACML extensions focused on geographic data, like Geo-XACML [74]. For the sake of simplicity —and without loss of generality— we refer to the basic XACML standard.

5.4 Performance

The PDP is queried for each `Location` flowing through the LbSprint architecture. Therefore, it is a quite critical module for the system efficiency. Since our PDP is based on XML, which is a textual representation, some performance-related concerns could arise. We present a practical ready-to-use optimization that considerably improves the performance of the policy evaluation process.

In XACML standard, the authorization requests and the authorization decisions are represented by XML code. Such a code is usually sent to the PDP as an XML string. This is the simplest approach, since the construction of the request involves only string concatenations. The request is successively converted in a DOM (Document Object Model) representation, and then in a PDP-specific internal representation (Figure 5.9). The first conversion involves an XML parsing that is burdensome in terms of computational resources. If the PEP and the PDP communicate locally, it is better to send the filter strategy request directly in a DOM format. Such a DOM must be

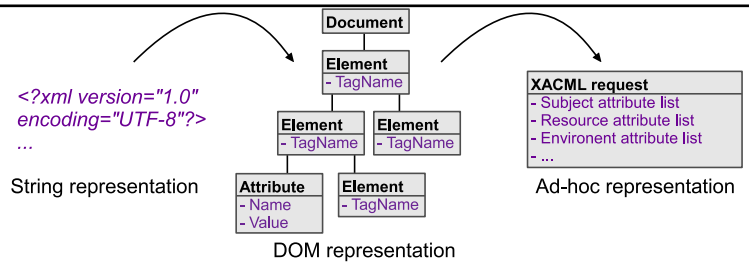


Figure 5.9. XML conversion

built node by node, by means of Java XML APIs. This requires more programming complexity than just making XML string concatenations. However, it has a good effect on performance.

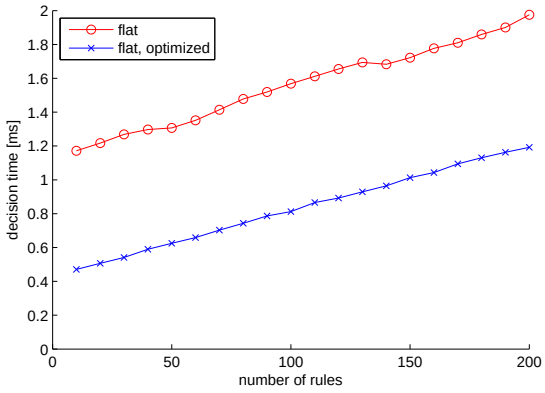


Figure 5.10. SunXACML time performance (flat policies)

We have conducted a series of tests aimed at evaluating the effect of such an optimization. Time efficiency has been evaluated on synthetic privacy policies containing up to 200 rules. Each target contained string matches and other comparisons on multiple attributes. Our experiments were carried out on a laptop PC running Windows 7 with $4\,\mathrm{Gb}$ of memory and an Intel Core i7-Q720 processor, with 4 cores at 1.6GHz and $6\,\mathrm{Mb}$ cache. Figure 5.10 shows the trend of the time performance of the evaluation of a flat policy versus the number of rules. The efficiency improvement goes from 40% (at 200 rules) to 60% (at 10 rules). In order to test the optimization effect on compound rules, we reorganized the policy in a balanced k-ary tree with $k = 10$. Non-leaf nodes represents compound rules, whereas leaves represents simple rules. All the targets evaluate a numerical attribute. The targets of the compound rules test it for being comprised inside a range. The targets of the simple rules test it for being equal to a single value. If the target of a compound rule evaluates into false, those

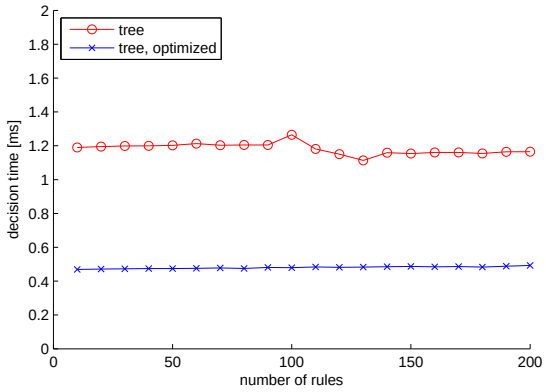


Figure 5.11. SunXACML time performance (tree policies)

of its children will not be evaluated. Thus, in a balanced tree, the number of target evaluations will grow as $\log n$, where n is the number of rules. The balanced tree is the best-case structure from the performance standpoint, whereas the flat structure is the worst case. Figure 5.11 shows the trend for such a hierarchical policy. From the experiments, we see that the evaluation performance is roughly constant with respect to the number of rules. This is because the time which the PDP takes to perform rule-independent tasks (e.g. conversion from DOM representation to ad-hoc representation) is preponderant with respect to the target evaluation time. The optimization brings a uniform efficiency improvement of about 60%.

More optimizations are possible on the PDP, based on more complex techniques like rule indexing, attribute numericalization, etc. [64, 99] Our optimization focuses on PEP-PDP communication, rather than on PDP internal working. So it can be applied in addition to such techniques, to obtain even more efficient policy decisions.

6

UniLO: a uniformity-based approach to location privacy

The retrieval of people's location raises several privacy concerns, as it is personal, often sensitive, information. The indiscriminate disclosure of such data could have highly negative effects, from undesired location-based advertising to personal safety attempts.

Samarati and Sweeney [86, 87] proposed the concept of k-*anonymity*: a system offers a k-anonymity to a user if his identity is undistinguishable from at least $k-1$ other users. k-anonymity concepts have been applied to location privacy [44, 48, 56, 93] by obfuscating the user's position in such a way to confuse it with the positions of other $k-1$ users. Location k-anonymity offers high levels of privacy, because it protects the user's identity. However, since k-anonymity does not permit the identification of the user, it is not applicable in services in which the user authenticates, e.g. payable services or location-based social networks. In addition, they require the presence of $k-1$ users in the proximity, that could be missing, and a central anonymizer, that could not be fully trusted by the users.

A different and promising approach is *data obfuscation* [8, 68]. The aim is not to reach anonymity, but rather to artificially reduce the precision of location data before disclosing it. In this way, the service can still be delivered, but an adversary cannot infer other sensitive information. We focus on obfuscation through *noise perturbation* [37, 59], an approach that recently gained the attention of industry [95]. One of the most underrated problems in the literature is how to choose a suitable noise to effectively perturb data. We found that, if noise is not chosen properly, perturbation will not resist to attacks based on statistical inference. In particular, an obfuscation operator must offer a spacial *uniformity* of probability. Such a requirement is often postulated, rather than fulfilled, by state-of-the-art perturbation methods.

We propose UniLO, a location obfuscation operator able to guarantee uniformity even in the presence of imprecise location measurements. UniLO does not require a centralized and trusted obfuscator. We deal with service differentiation by proposing and comparing three UniLO-based obfuscation algorithms offering multiple contemporaneous levels of privacy. Finally, we experimentally prove that UniLO outperforms

state-of-the-art perturbation algorithms both in terms of utility and resistance against inference attacks. All the simulations scripts of the present chapter can be downloaded from [90]. This chapter has been partially published as a conference paper [33].

The rest of the chapter is organized as follows. Section 6.1 analyzes some related works and the differences with UNILO techniques. Section 6.2 introduces some basic concepts concerning the system model and the terminology. Section 6.3 formally describes the agnostic adversary model, the concept of uniformity, and a way to quantify it. Section 6.4 presents the basic UNILO operator and show its properties in terms of uniformity. Section 6.5 presents the problem of offering multiple levels of privacy and three algorithms to adapt UNILO in this sense. Section 6.6 presents an example location-based service and evaluates UNILO algorithms in terms of utility. Section 6.7 evaluates UNILO algorithms in terms of resistance against inference attacks.

6.1 Comparison of UNILO to the state of the art

Approaches for location privacy can be roughly divided in *identity-protection* approaches and *data-protection* approaches. Identity protection avoids the re-identification of anonymous users. k*-anonymity* and *mix zones* fall in this category. Data protection avoids the disclosure of precise locations. *Obfuscation* and *private information retrieval* fall in this category.

6.1.1 Identity-protection approaches

Gruteser and Grunwald [48] first approached k*-anonymity* problem in location-based services. The proposed solution involves the subdivision of the map in quadrants with different granularities. The k-anonymity approach is broadly used in many research works [44, 47, 56, 69, 93, 102]. However, since these methods do not permit the identification of the user, they are not applicable in services in which the user authenticates himself, e.g. payable services or location-based social networks. In addition, they require the presence of $k-1$ users in the proximity, that could be missing, and a central anonymizer, that could not be fully trusted by users. Chow et al. [21] proposed a method to reach k-anonymity without a centralized anonymizer, but it requires burdensome peer-to-peer communication between mobile devices. Our approach is orthogonal to k-anonymity, since it aims at protecting the position, rather than the identity of the user.

[105] and [2] approach the problem of *trajectory* k*-anonymity*, offering methods to protect user's privacy in continuous tracking systems. Although it could be extended in that sense, the present work focuses on single-position queries, as they encompass a wide range of location-based applications.

A problem complementary to anonymity is *pseudonym unlinkability* in tracking systems, usually approached with the technique of mix zones [13, 43, 77]. Mix zones are areas of the map where users cannot be tracked and change their pseudonym.

By carefully placing and dimensioning such mix zones it is possible to thwart the adversary from linking two consecutive pseudonyms of the same user.

6.1.2 Data-protection approaches

Location obfuscation aims at reducing the precision of location data before disclosing it. The research on this topic focused mainly on what kind of service can be delivered with imprecise positions [18, 36, 68, 106]. The problem of generating such imprecise positions in a proper way is often underrated. In particular, the uniformity of the obfuscation is often postulated, rather than evaluated. As a result, the proposed solutions turn out to be poorly resistant against inference attacks.

Mascetti et al. [68] proposed a friend-proximity service based on imprecise positions. The obfuscation is achieved with a granularity-based technique, by releasing the granule containing the true position. The drawback of granularity-based obfuscation is that it produces non-circular, and often irregular, privacy areas. This is not compatible with many legacy systems, which are designed to accept circular areas.

Ardagna et al. [8] proposed a set of obfuscation operators that perturb the location: radius enlargement, radius restriction, center shift. These operators transform a measurement area into an obfuscated one. Our approach guarantees both more private and more useful obfuscated areas. More private because UNILO noise significantly increases the uniformity of the resultant privacy areas. More useful because we always guarantee that the privacy areas contain the user's position. An LBS provider can thus rely on more powerful assumptions and offer more quality of service. In addition, in [8] the resistance against attacks relies on the fact that the adversary is unaware of the privacy preference of the user. This could be an optimistic assumption, which features a form of "security by obscurity" that should be avoided [89].

Krumm [59] surveyed many different obfuscation methods and applied them to real-life GPS traces. The objective was to prevent an attacker from inferring users' home positions. Krumm tried also a perturbation-based method, which involved noise with a Gaussian-distributed magnitude. He found that this method requires a high quantity of noise ($\sigma = 5\,\mathrm{Km}$) in order to effectively prevent inference attacks. Our approach offers higher levels of uniformity, and reduces the amount of noise needed to resist to inference attacks.

Dürr et al. [37] proposed an obfuscation approach with multiple levels of privacy. They build different "shares" which are random vectors concatenated to the user's position. They store the shares in different servers to avoid a single point of trust. Each LBS provider reconstructs the position by "fusing" one or more shares from one or more servers. The privacy level is proportional to the number of shares the LBS provider is allowed to access. To build the shares, two possible algorithms are used, offering different properties. The "*a-priori* share generation algorithm", which builds the shares from the larger one to the smaller one, and the "*a-posteriori* share generation algorithm" which does vice versa. To the best of our knowledge, the "*a-*

posteriori share generation algorithm" is the approach most similar to ours. However, our obfuscation operators guarantee more resistance against inference attacks.

Inspired by differential privacy [38], Andrés et al. [6] introduced the concept of ϵ-*geo-indistinguishability*. The idea is that the user obtains more privacy in the surroundings of his true position, and less farther. To achieve this, they perturb the true position with a 2-dimensional extension of the Laplacian noise. Such a noise is highly non-uniform. As a consequence, geo-indistinguishability offers far less resistance to inference attacks compared to UNILO.

A recent example of location obfuscation is the *n-CD* approach of [63]. Authors proposed the generation of such concealed disks (CDs), whose combination gives an obfuscated user's position. The overlapping and random rotation of CDs preserves unpredictability of the resulting obfuscation area (called "anonymity zone"). The CDs are selected in such a way that their overlapping is consistent. The n-CD approach does not guarantee multiple privacy levels. Moreover, the proposed privacy metric considers only the resulting intersection area of the CDs, without analyzing the probability distribution of the target position within the anonymity zone.

Other notable obfuscation-based approaches are [18, 36, 106]. All these works postulate uniformity rather than providing for it. In contrast, our approach offers guarantees on the obfuscation uniformity, even in presence of imprecise location measurements.

Another research track [46, 52, 79] applies *private information retrieval* (PIR) techniques to protect user's location. The objective is to deliver the service without disclosing the user's location at all. These approaches offer high security, but they involve complex cryptographic operations, which scale poorly at the server side. [78] and [57] eliminates this problem by employing hardware PIR techniques. However, these solutions require trusted hardware modules, which could be unavailable on many servers.

6.2 System model

In our system, a *user* is an entity whose location is measured by a *sensor*. A *service provider* is an entity that receives the user's location in order to provide him with a *location-based service*. The user applies an *obfuscation operator* to location information prior to releasing it to the service provider. The obfuscation operator purposefully reduces the precision to guarantee a certain privacy level. Such a precision is defined by the user and reflects his requirements in terms of privacy. The more privacy the user requires, the less precision the obfuscation operator returns.

In the most general case, a *location measurement* is affected by an intrinsic error that limits its precision. Such an error depends on several factors including the localization technology, the quality of the sensor, the environment conditions. If the measurement error is small compared to the obfuscation, as it happens in professional GPS receivers, it can be approximated to zero. Otherwise, as it happens in cheap GPS receivers mounted on smartphones, or in Wi-Fi and cellular positioning,

we cannot neglect it [107]. This implies that the location cannot be expressed as a geographical point but rather as a neighborhood of the actual location. We assume that locations are always represented as *planar circular areas*, because it is a good approximation for many location techniques [8, 76, 107]. A location measurement (Fig. 6.1) can be defined as follows:

Definition 1 (Location measurement)**.** *Let* $\mathbf{X}$ *be the* actual *position of the user. A* location measurement *is a circular area* $A_0 = \langle \mathbf{X}_0, r_0 \rangle \subset \mathbb{R}^2$*, where* $\mathbf{X}_0$ *is the center and* r_0 *is the radius, such that* $\Pr[\mathbf{X} \in A_0] = 1$ (Accuracy Property)*. We further call* $\mathbf{X}_0$ *the* measured position *and* r_0 *the* precision radius*.*

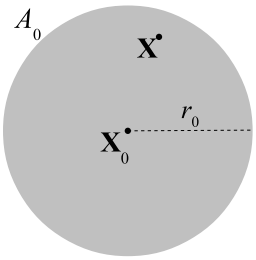


Figure 6.1. Location measurement

The Accuracy Property guarantees that the location measurement actually contains the user, or, equivalently, that the distance $\overline{\mathbf{X}\mathbf{X}_0}$ does not exceed r_0. We assume that r_0 is constant over time. This means either that the precision does not change over time, or that r_0 represents the worst-case precision.

A user specifies his privacy preference in terms of a *privacy radius* $r_1 > r_0$, meaning that he wishes to be located with a precision not better than r_1. The privacy radius is quite an easy metric to be understood by the users. This improves the overall usability of the obfuscation system. The task of an obfuscation operator is to produce a *privacy area* A_1 with radius r_1, appearing to the provider as a location measurement with a lower precision.

Definition 2 (Privacy area)**.** *Let* $\mathbf{X}$ *be the* actual *position of the user. A* privacy area *is a circular area* $A_1 = \langle \mathbf{X}_1, r_1 \rangle \subset \mathbb{R}^2$*, such that* $\Pr[\mathbf{X} \in A_1] = 1$ (Accuracy Property)*. We further call* $\mathbf{X}_1$ *the* obfuscated position *and* r_1 *the* privacy radius*.*

Definition 3 (Obfuscation operator)**.** *Let* A_0 *be a location measurement, and* $r_1 > r_0$ *a privacy radius. An obfuscation operator* $\mathcal{O}$ *transforms* A_0 *into a privacy area* A_1*:*

$$A_1 = \mathcal{O}(A_0) \tag{6.1}$$

With reference to Fig. 6.2, in order to produce a privacy area A_1, the obfuscation operator applies both an enlargement and a translation of the location measurement A_0. The enlargement aims at decreasing the precision and thus achieving the desired

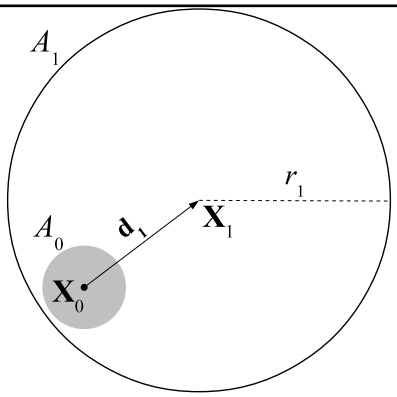


Figure 6.2. Obfuscation and shift vector

privacy level r_1. The translation is made through a randomly selected *shift vector* $\mathbf{d}_1$, i.e., $\mathbf{X}_0 + \mathbf{d}_1 = \mathbf{X}_1$. Of course, the user has to keep the shift vector secret.

The enlargement and translation operations must be such that, when composed, the resulting privacy area satisfies the Accuracy Property. The enlargement operation is straightforward, whereas the translation operation is more subtle. We state the following:

Proposition 2. *A privacy area A_1 fulfills the Accuracy Property iff:*

$$\|\mathbf{d}_1\| \leq (r_1 - r_0) \tag{6.2}$$

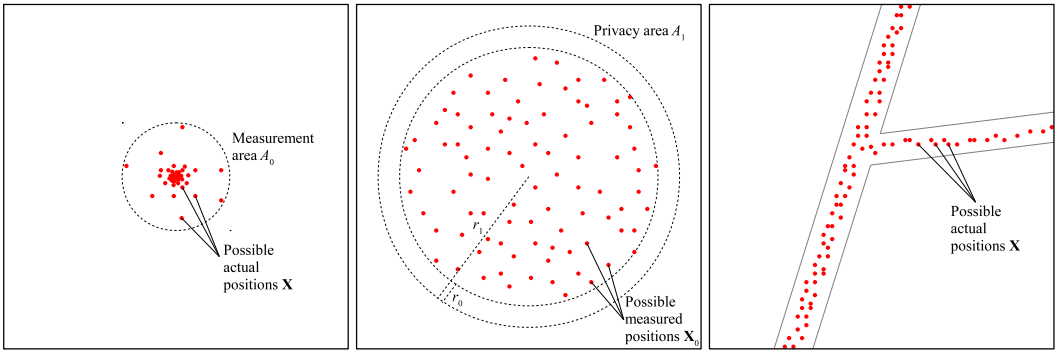
Proof. The proof stems directly from geometrical considerations (cfr. Fig. 6.2). □

6.3 Agnostic adversary and uniformity index

For the scope of the present chapter, every service provider receiving an obfuscated position is a potential adversary. We assume the adversary knows the privacy area and the precision radius. She aims at discovering the actual user's position. Since it cannot be known with infinite precision, the result of the attack will have a probabilistic nature. From now on, we will use the notation $f_{a|b}$ to refer to the conditional probability density function of the random variable a given the information b.

Three pieces of information could help the adversary: (1) the employed localization technology; (2) the employed obfuscation operator; (3) other auxiliary information. Such information pieces are modeled by three probability densities over space:

1. The density $f_{\mathbf{X}|A_0}$ (Fig. 6.3a), which describes the actual position of the user given a measurement of it. We have no control over this density but we can make hypotheses on it. For example we can suppose it is normally distributed, as it is usually done in GPS measurements [51]. Obviously, even the adversary can make the same assumptions.



(a) $f_{\mathbf{X}|A_0}$ (b) $f_{\mathbf{X}_0|A_1}$ (c) $f_{\mathbf{X}|\text{auxinfo}}$

Figure 6.3. Adversarial information

2. The density $f_{\mathbf{X}_0|A_1}$ (Fig. 6.3b), which describes the measured position of the user given an obfuscated version of it. We can control this density, and this is our main weapon against the adversary. The adversary can compute this density by analyzing the obfuscation operator. She starts from the inverse density $f_{A_1|\mathbf{X}_0}$, which describes the possible output of the obfuscation operator, and then applies Bayesian inference. In obfuscation operators based on noise addition, $f_{A_1|\mathbf{X}_0}$ depends on the distribution of $\mathbf{d}_1$, while $f_{\mathbf{X}_0|A_1}$ depends on the distribution of $-\mathbf{d}_1$.
3. The density $f_{\mathbf{X}|\text{auxinfo}}$ (Fig. 6.3c), which describes the actual position of the user given a set of auxiliary information about him. We have no control over this density and it is even hard to make hypotheses on it. The adversary could have much or little information. She could know where the user lives, works, his habits, etc. Or she could know nothing particular about the user but still suppose that he follows the overall population distribution in a given city at a given hour. On the extreme end, an adversary *who ignores any auxiliary information* cannot conclude nothing. So she has to assume that $f_{\mathbf{X}|\text{auxinfo}}\,(x, y)$ is uniform over the whole world map.

Dealing with auxiliary information is a central problem in privacy topics [87, 38]. Some works in the literature [9, 91] make assumptions on auxiliary information that is owned by the adversary, and enlarge the privacy areas to compensate it. This approach has some drawbacks. First of all, it does not respect the privacy preference that the user specified in terms of privacy radius. Second, it requires that the system knows the same auxiliary information that the adversary knows. In theory, there are no limits to the quantity of auxiliary information the adversary can have. If an adversary knew enough auxiliary information, $f_{\mathbf{X}|\text{auxinfo}}$ would collapse into a Dirac delta. In such a case, she could locate the user without needing any obfuscated position at all.

No obfuscation operator can make the adversary "*forget*" the auxiliary information. Therefore, our true aim is to give her no additional information other than the

simple one: "$\mathbf{X}$ is inside A_1." We model such a requirement with the concept of *ideal obfuscation*:

Definition 4 (Ideal obfuscation)**.**

$$f_{\mathbf{X}|\textit{auxinfo},A_1}(x,y) = f_{\mathbf{X}|\textit{auxinfo},\mathbf{X}\in A_1}(x,y) \tag{6.3}$$

An obfuscator which performs ideal obfuscation is an *ideal obfuscator*. Note that "given A_1" in the left term of Equation 6.3 differs from "given $\mathbf{X} \in A_1$" in the right term. The former means that the adversary knows the privacy area generated by the obfuscation operator. The latter means that the adversary knows that the user is inside an area A_1, not necessarily generated by an obfuscation operator. Intuitively, in order Equation 6.3 to hold, the obfuscation operator should produce a privacy area in such a way that the actual position is uniformly distributed inside it. We state the following:

Definition 5 (Uniformity Property)**.** *A privacy area A_1 fulfills the Uniformity Property iff $f_{\mathbf{X}|A_1}(x,y)$ is uniform over A_1. An obfuscator fulfills the Uniformity Property iff all the produced privacy areas fulfill the Uniformity Property.*

Theorem 1. *An obfuscator which offers Uniformity Property is ideal.*

Proof. From the definition of conditional probability, we have that:

$$f_{\mathbf{X}|\text{auxinfo},\mathbf{X}\in A_1} = \begin{cases} \frac{f_{\mathbf{X}|\text{auxinfo}}}{\iint_{A_1} f_{\mathbf{X}|\text{auxinfo}}\,\mathrm{d}x\mathrm{d}y} & \text{in } A_1 \\ 0 & \text{outside} \end{cases} \tag{6.4}$$

On the other hand, if Uniformity Property holds:

$$f_{\mathbf{X}|A_1} = \begin{cases} \frac{1}{\text{size}(A_1)} & \text{in } A_1 \\ 0 & \text{outside} \end{cases} \tag{6.5}$$

Combining (6.5) with the auxiliary information:

$$\begin{aligned} f_{\mathbf{X}|\text{auxinfo},A_1} &= \frac{f_{\mathbf{X}|\text{auxinfo}} \cdot f_{\mathbf{X}|A_1}}{\iint_{\mathbb{R}^2} f_{\mathbf{X}|\text{auxinfo}} \cdot f_{\mathbf{X}|A_1}\,\mathrm{d}x\mathrm{d}y} \\ &= \begin{cases} \frac{f_{\mathbf{X}|\text{auxinfo}}}{\iint_{A_1} f_{\mathbf{X}|\text{auxinfo}}\,\mathrm{d}x\mathrm{d}y} & \text{in } A_1 \\ 0 & \text{outside} \end{cases} \\ &= f_{\mathbf{X}|\text{auxinfo},\mathbf{X}\in A_1} \end{aligned} \tag{6.6}$$

□

Theorem 1 tells us that Uniformity Property is important regardless of the auxiliary information the adversary has, because Uniformity Property gives her no additional one.

No obfuscation system can provide uniformity against an adversary holding some auxiliary information. This is because the adversary will always have a non-uniform

a-priori probability density of the user's position. In order to study the uniformity of a generic obfuscation operator, we have to suppose an adversary *who ignores any auxiliary information*. From now here, we will call such an adversary the *agnostic adversary*. The agnostic adversary is a purely theoretic one, because real-life adversaries usually own some auxiliary information. However: (a) the agnostic adversary permits us to study the uniformity of obfuscation operators; and (b) if an obfuscation operator enjoys Uniformity Property against an agnostic adversary, it also gives no additional information to a real-life adversary, i.e. it is ideal.

6.3.1 Uniformity index

We use the agnostic adversary to measure the uniformity of a generic obfuscation method. Basing on the measurement error's density and the shift vector's density, the agnostic adversary computes the *pdf* $f_{\mathbf{X}|A_1}(x,y)$ of the user's position. After that, she defines a *confidence goal* $c \in (0,1]$ and computes the smallest area $\hat{A}^c \subseteq A_1$ which contains the user with a probability c. We call this area the *smallest c-confidence area*.

Definition 6 (Smallest c-confidence area)**.**

$$\hat{A}^c = \arg\min_{A \in \mathcal{A}^c} \{\text{size}(A)\} \tag{6.7}$$

where:

$$\mathcal{A}^c = \left\{A | A \subset \mathbb{R}^2, \Pr\left[\mathbf{X} \in A | A_1\right] = c\right\} \tag{6.8}$$

$$\Pr\left[\mathbf{X} \in A | A_1\right] = \iint_A f_{\mathbf{X}|A_1}(x,y)\,\mathrm{d}x\mathrm{d}y \tag{6.9}$$

The smallest c-confidence area is the adversary's most precise estimation of the actual position, and it will cover the zones where $f_{\mathbf{X}|A_1}(x,y)$ is more concentrated. The adversary can find it by means of a Monte Carlo approach. First, she synthesizes many "measurement-plus-obfuscation" operations, finding many couples with the form:

$$\langle \text{actual position}, \text{obfuscated position} \rangle$$

Then, she selects only those couples whose obfuscated position matches with the one she wants to deobfuscate. The distribution of the selected actual positions follows $f_{\mathbf{X}|A_1}(x,y)$. Finally, the adversary determines the smallest c-confidence area by connecting the zones having highest concentrations. The smaller $\hat{A}^c$, the more precisely the adversary locates the user. A good obfuscation operator should keep $\hat{A}^c$ as larger as possible. This is done by making the obfuscation as uniform as possible. The best case occurs when the Uniformity Property is fulfilled, and the obfuscator is ideal. This is not realizable in the general case, because we cannot force a particular *pdf* inside the privacy area if we cannot control the *pdf* inside the measurement area, which depends on the measurement error.

Another way to state the Uniformity Property is the following:

Proposition 3. *A privacy area A_1 fulfills the Uniformity Property iff:*

$$\forall A \subseteq A_1, \Pr\left[\mathbf{X} \in A | A_1\right] = \frac{\text{size}(A)}{\text{size}(A_1)} \tag{6.10}$$

That is, each region of the privacy area contains the user with a probability proportional to its size. In such a case:

$$\text{size}(\hat{A}^c) = c \cdot \text{size}(A_1) \tag{6.11}$$

Otherwise:

$$\text{size}(\hat{A}^c) < c \cdot \text{size}(A_1) \tag{6.12}$$

The uniformity can be quantified by means of Eq. 6.12, by measuring how much, for a given c, $\text{size}(\hat{A}^c)$ gets close to $c \cdot \text{size}(A_1)$. We define the following *uniformity index* by fixing $c = 90\%$:

Definition 7 (Uniformity index)**.**

$$\text{unif}\left(A_1\right) = \frac{\text{size}(\hat{A}^{90\%})}{90\% \cdot \text{size}(A_1)} \tag{6.13}$$

The uniformity index ranges from 0% (worst case), if the user's position is perfectly predictable, to 100% (best case), if the user's position is perfectly uniform. A uniformity index of 100% is necessary and sufficient for the Uniformity Property.

The uniformity index is a sort of 2-dimensional extension of the "privacy metric" used in [4] for data mining applications, except that our index is unit-less, because normalized on the privacy area size. Therefore it measures the uniformity of obfuscation, rather than the quantity of privacy. Uniformity index is proportional to the lack of precision of the attack. For example, if an obfuscation operator produces a privacy area of $400\,\text{m}^2$ with a uniformity index of 80%, an agnostic adversary cannot find his position (with 90% confidence) with more precision than $80\% \cdot 90\% \cdot 400 = 288\,\text{m}^2$.

6.4 UNILO obfuscation operator

UNILO (**Uni**form **L**ocation **O**bfuscation) adds to $\mathbf{X}_0$ a shift vector $\mathbf{d}_1 = (\mu\cos\varphi, \mu\sin\varphi)$, where μ is the magnitude and φ is the angle. μ and φ have the following probability densities (Fig. 6.4):

$$f_\Phi\left(\varphi\right) = \begin{cases} \frac{1}{2\pi} & \varphi \in [0, 2\pi) \\ 0 & \text{otherwise} \end{cases} \tag{6.14}$$

$$f_M\left(\mu\right) = \begin{cases} 2\mu/(r_1 - r_0)^2 & \mu \in [0, r_1 - r_0] \\ 0 & \text{otherwise} \end{cases} \tag{6.15}$$

These densities produce shift vectors with magnitude less than or equal to $r_1 - r_0$,

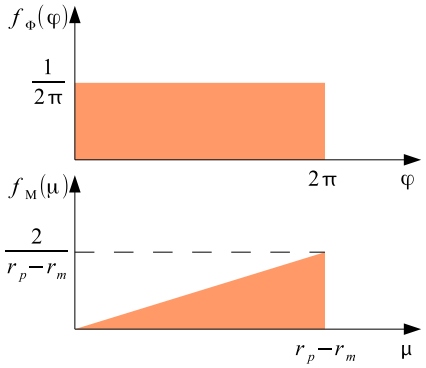


Figure 6.4. φ and μ *pdf*'s of a UNILO vector

and a perfectly uniform spacial probability density. This produces a good level of uniformity of $f_{\mathbf{X}|A_1}$. However, remind that $f_{\mathbf{X}|A_1}$ also depends on the measurement error's density, over which we have no control. So $f_{\mathbf{X}|A_1}$ will not be perfectly uniform in the general case.

UNILO fulfills the following properties:

- *Accuracy Property.* The privacy area always contains the user (Theorem 2).
- *High uniformity index.* UNILO outperforms all the other noise shapes used in the literature in terms of uniformity index, for all values of r_1/r_0.
- *Uniformity Property as* $r_0 \to 0$*.* With highly precise sensors, UNILO tends to be an ideal obfuscator. (Theorem 3).

Theorem 2. UNILO *fulfills Accuracy Property.*

Proof. By construction, $\|\mathbf{d}_1\| \leq r_1 - r_0$. Hence, from Prop. 2, Accuracy holds. □

Theorem 3. *As* $r_0 \to 0$*,* UNILO *fulfills Uniformity Property.*

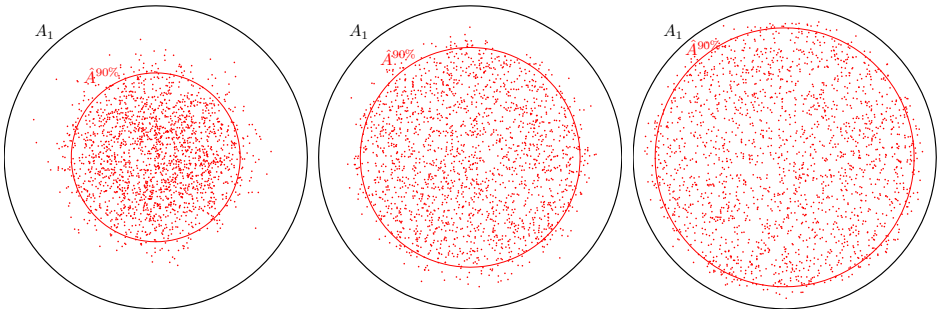
Proof. If $r_0 \to 0$, A_0 will narrow to a point, with $\mathbf{X} \equiv \mathbf{X}_0$, and the probability density of the magnitude in Eq. 6.15 will become:

$$f_M(\mu) = \begin{cases} 2\mu/r_1^2 & \mu \in [0, r_1] \\ 0 & \text{otherwise} \end{cases} \tag{6.16}$$

To show the Uniformity, we have to pass from the polar representation to the Cartesian representation. So we have to transform the densities $f_M(\mu)$, $f_\Phi(\varphi)$ to the joint density $f_{X,Y}(x, y)$. In order to perform this variable change, we equal the areas of the rectangle spaced by $\mathrm{d}x$ and $\mathrm{d}y$, and of the annulus sector spaced by $\mathrm{d}\mu$ and $\mathrm{d}\varphi$:

$$\mathrm{d}x\mathrm{d}y = \frac{(\mu + \mathrm{d}\mu)^2 - \mu^2}{2}\mathrm{d}\varphi = \mu \cdot \mathrm{d}\mu\mathrm{d}\varphi \tag{6.17}$$

Then, we equal the probabilities inside them:



(a) $r_1/r_0 = 2$ (b) $r_1/r_0 = 4$ (c) $r_1/r_0 = 10$

Figure 6.5. UNILO spatial distribution (2000 Monte Carlo runs)

$$f_{X,Y}(x,y)\,\mathrm{d}x\mathrm{d}y = f_M(\mu)\,\mathrm{d}\mu \cdot f_\Phi(\varphi)\,\mathrm{d}\varphi \tag{6.18}$$

$$= \begin{cases} 2\mu/r_1^2 \mathrm{d}\mu \cdot \frac{1}{2\pi}\mathrm{d}\varphi & \mu \le r_1 \\ 0 & \text{otherwise} \end{cases} \tag{6.19}$$

From Equations 6.17 and 6.19, we have:

$$f_{X,Y}(x,y) = \begin{cases} \frac{1}{r_1^2\pi} & \sqrt{x^2+y^2} \le r_1 \\ 0 & \text{otherwise} \end{cases} \tag{6.20}$$

which is spatially uniform in A_1. □

We will use the following notation:

$$\mathbf{d}_1 = \mathrm{UniLO}(r_1, r_0) \tag{6.21}$$

to say that $\mathbf{d}_1$ is a shift vector created by the UNILO operator with privacy radius r_1 and precision radius r_0. UNILO operator will be our basic block to build more complex obfuscators.

We evaluated the uniformity index of UNILO on simulated location measurements. The error on the location measurements was assumed to follow a Gaussian distribution, as it is usually done in GPS [51]. We truncated the distribution at $r_0 = 3\sigma$, so that no sample falls outside A_0. Such a truncated Gaussian distribution differs from the untruncated one for only 1% of samples. The tests aim at evaluating the uniformity of UNILO with respect to the ratio r_1/r_0 (*radius ratio*).

Figure 6.5 shows the statistical distribution of $\mathbf{X}$ in A_1 for different values of the radius ratio. We note that the distribution tends to be perfectly uniform as $r_1/r_0 \to \infty$. The inner areas are $\hat{A}^{90\%}$.

We compared UNILO with other state-of-the-art obfuscation noises[1]:

[1] In case of unbounded noises (e.g. Gaussian), we fulfilled Accuracy Property by truncating their magnitude at $(r_1 - r_0)$. To make meaningful comparisons, we tailored the parameters

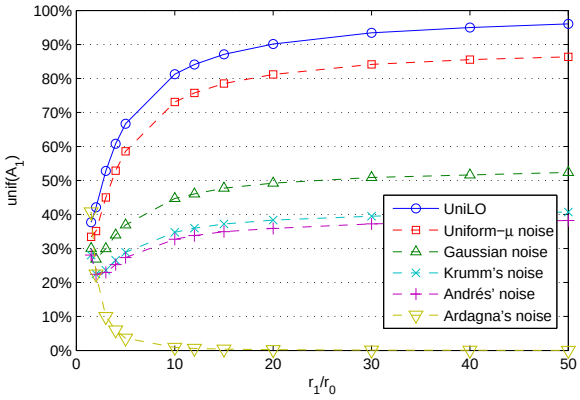


Figure 6.6. $\mathrm{unif}\,(A_1)$ wrt the radius ratio (500K Monte Carlo runs for each point)

- Gaussian noise, used for modeling 2-dimensional measurement errors.
- Krumm's noise, used by Krumm to perturb GPS data [59]. Krumm's noise has a uniformly distributed angle and a magnitude drawn from a Gaussian distribution.
- Andrés' noise, used by Andrés et al. [6]. This noise is a 2-dimensional extension of the Laplacian noise, and it is used to achieve *geo-indistinguishability*. Refer to [6] for further information.
- Dürr's noise, used by Dürr et al. in their "*a-posteriori* share generation algorithm" [37]. This is the simplest 2-dimensional noise: it has a uniformly distributed angle and a uniformly distributed magnitude. We compare UNILO with this because it is the obfuscation method most similar to ours.
- Ardagna's noise, used by Ardagna et al. in their location obfuscation operators [8]. These are a set of obfuscation operators that reduce/enlarge/shift the measurement area to produce the privacy area. The user expresses his privacy preference in terms of *final relevance*, which is assumed to be unknown by the adversary. With "Ardagna's noise" we refer here to the cumulative effect of (a) the random selection of the final relevance, (b) the random selection of the obfuscation operator, and (c) the random selection of the shift angle. These obfuscation operators do not guarantee the Accuracy Property, and the user could be outside the privacy area. Refer to [8] for further information.

Figure 6.6 shows the uniformity indexes of the noises. We can see that UNILO outperforms all the other noises for all the radius ratii. In the average case, Ardagna's noise is particularly easy to predict, because it has not been designed to thwart statistical attacks. On the other hand, it enjoys quite a high uniformity for very small privacy radii ($r_1 < 2r_0$). However, such an improved uniformity is obtained at the cost of vi-

in such a way to truncate always 1% of the samples. Namely, we tailored $\sigma = (r_1 - r_0)/3$ for Gaussian noise, $\sigma = (r_1 - r_0)/2.6$ for Krumm's noise [59], and $\epsilon = 6.5/(r_1 - r_0)$ for Andrés' noise [6].

olating the Accuracy Property, and thus possibly degrading the utility of the service. Krumm's and Gaussian noises are not so good at obfuscating. We believe this is the reason why Krumm needed a surprisingly high quantity of noise ($\sigma = 5\,\mathrm{Km}$) to effectively withstand inference attacks [59]. Andrés' noise for geo-indistinguishability is quite predictable too.

6.5 Multiple levels of privacy

A user may require different privacy radii for different services. He can require high levels of privacy for some services, for instance a friend-finder service, and small levels of privacy for others, for instance safety-related services. In general, an obfuscator must offer a user *a set* of N possible privacy radii, and must create *a set* of N random shift vectors, one for each privacy radius. The precision radius of the sensor can be considered as the minimum privacy radius. In other words, the smallest privacy area is the measurement itself.

Let $\rho = \{r_0, r_1, r_2, \cdots, r_N\}$, with $r_0 < r_1 < r_2 < \cdots < r_N$, be the *privacy radius set*, i.e. the set of the privacy radii provided by the obfuscator. Then:

- $\{\mathbf{d}_i : i = 1, 2, \cdots, N\} = \delta$ is the *shift vector set*,
- $\{\mathbf{X}_i = \mathbf{X}_0 + \mathbf{d}_i : i = 1, 2, \cdots, N\}$ is the *center set*,
- $\{A_i = \langle \mathbf{X}_i, r_i \rangle : i = 1, 2, \cdots, N\}$ is the *privacy area set*.

We will refer to r_{i-1} and r_{i+1} as, respectively, the *previous* and the *successive* privacy radii of r_i. The same convention holds for shift vectors and privacy areas. We will use the notation $\hat{A}_i^c$ to refer to the smallest c-confidence area found by an agnostic adversary able to access to the i-th privacy level.

6.5.1 On collusion attack

A subtle attack is possible when two or more service providers collude. Let us suppose that a service provider knowing A_1 colludes with a service provider knowing A_2. If the shift vectors are not chosen wisely, the adversaries can intersect A_1 and A_2 (Fig. 6.7) to find a smaller area containing the user. To avoid this possibility, an obfuscator should force each privacy area to enclose all the smaller ones. We state the following:

Definition 8 (Inclusion Property)**.** *A privacy area A_i ($i \geq 2$) fulfills the Inclusion Property iff $A_{i-1} \subset A_i$. An obfuscator fulfills the Inclusion Property iff all the produced privacy areas fulfill the Inclusion Property.*

Obviously, in case of collusion we can fulfill only the privacy preference corresponding to the smallest area known by the group of colluding adversaries.

If a privacy area must enclose the previous one, the distance between the centers must not be larger than the radii difference. Formally:

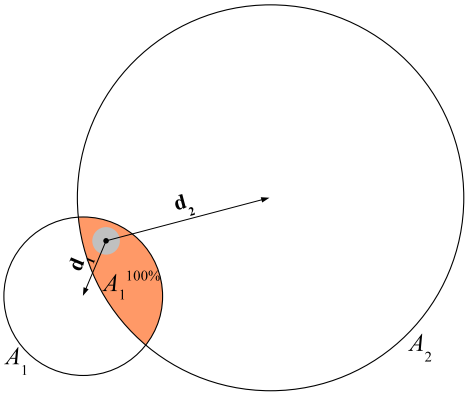


Figure 6.7. Collusion attack

Proposition 4. *A privacy area A_i ($i \geq 2$) fulfills the Inclusion Property iff:*

$$\|\mathbf{d}_i - \mathbf{d}_{i-1}\| \leq (r_i - r_{i-1}) \tag{6.22}$$

It is worth to stress that the Inclusion Property is not mandatory. In particular, it can be released if both the system prevents service providers from accessing different privacy levels, and different service providers do not collude. The Inclusion Property lowers the uniformity index of the privacy areas.

6.5.2 UNILO for multiple levels of privacy

We will now adapt the basic UNILO operator for offering a set ρ of N shift vectors. The simpler solution is to apply N times UNILO, obtaining N shift vectors independent of each other. Formally:

$$\mathbf{d}_i = \mathrm{UniLO}(r_i, r_0)\ \forall i \tag{6.23}$$

We will refer to this solution as *Independent Vectors* UNILO (IV-UNILO). Figure 6.8 shows an example with $\rho = \{r_0, r_1 = 4r_0, r_2 = 16r_0\}$.

IV-UNILO trivially fulfills the Accuracy Property for all the privacy areas. It also offers a good level of uniformity, especially for large privacy radii ($r_i \gg r_0$). Figure 6.8 shows that A_2 does not enclose A_1. Thus, IV-UNILO does not fulfill the Inclusion Property and does not defend against collusion.

6.5.3 VC-UNILO: Vector Chain UNILO

The idea of VC-UNILO is to fulfill Inclusion by assuring that the distance between $\mathbf{X}_1$ and $\mathbf{X}_2$ never goes beyond $(r_2 - r_1)$. To do this, we create $\mathbf{d}_2$ as the sum of $\mathbf{d}_1$ and an *incremental vector* $\mathbf{d}_{1,2}$, which is a random vector with maximum magnitude $(r_2 - r_1)$. The incremental vector represents in fact the distance between $\mathbf{X}_1$ and $\mathbf{X}_2$. The same procedure is repeated for $\mathbf{d}_3 \cdots \mathbf{d}_N$. In this way, we fulfill both Accuracy and Inclusion, as stated by the following two Theorems:

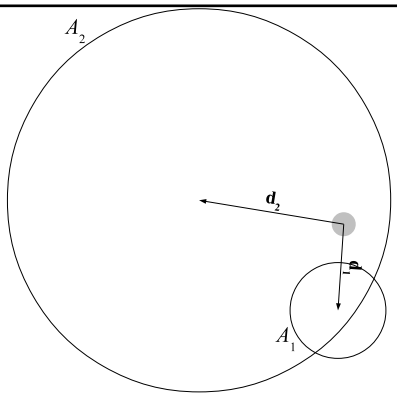


Figure 6.8. IV-UNILO example

Theorem 4. VC-UNILO *fulfills the Accuracy Property for all the privacy areas.*

Proof. We prove this by induction. From Theorem 2, A_1 fulfills Accuracy. If Accuracy holds for A_{i-1}, then $\|\mathbf{d}_{i-1}\| \leq (r_{i-1} - r_0)$. By construction $\|\mathbf{d}_{i-1,i}\| \leq (r_i - r_{i-1})$. It follows that:

$$\|\mathbf{d}_i\| \leq \|\mathbf{d}_{i-1}\| + \|\mathbf{d}_{i-1,i}\| \tag{6.24}$$
$$\leq (r_{i-1} - r_0) + (r_i - r_{i-1}) \tag{6.25}$$
$$= (r_i - r_0) \tag{6.26}$$

Hence, from Prop. 2, Accuracy Property holds for all privacy areas. □

Theorem 5. VC-UNILO *fulfills the Inclusion Property for all the privacy areas.*

Proof. We consider the generic privacy area A_i. By construction, $\|\mathbf{d}_i - \mathbf{d}_{i-1}\| = \|\mathbf{d}_{i-1,i}\| \leq (r_i - r_{i-1})$. Hence, from Prop. 4, Inclusion Property holds for all privacy areas. □

For $\mathbf{d}_{i-1,i}$ we choose vectors created by UNILO operator:

$$\mathbf{d}_{i-1,i} = \mathrm{UniLO}(r_i, r_{i-1}) \tag{6.27}$$

This is the simplest choice and still offers a good level of uniformity for A_i. To sum up, VC-UNILO algorithm creates the shift vectors with the following formula:

$$\mathbf{d}_i = \begin{cases} \mathrm{UniLO}(r_1, r_0) & i = 1 \\ \mathbf{d}_{i-1} + \mathrm{UniLO}(r_i, r_{i-1}) & i > 1 \end{cases} \tag{6.28}$$

The i-th shift vector is created by concatenating vectors, hence the name *Vector Chain*. Figure 6.9 shows an example with $\rho = \{r_0, r_1 = 4r_0, r_2 = 16r_0\}$.

VC-UNILO defends against collusion but offers a lower uniformity index than IV-UNILO. The problem is that $\|\mathbf{d}_i\|$ $(i > 1)$ has a low probability of being large. In fact,

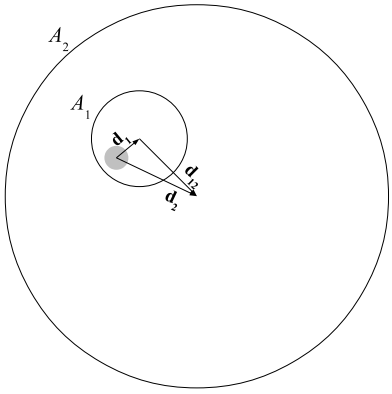


Figure 6.9. VC-UNILO example

$\mathbf{d}_i$ is the sum of two vectors ($\mathbf{d}_{i-1}$ and $\mathbf{d}_{i-1,i}$) and its magnitude gets close to the maximum (r_i) only if the vectors are aligned on the same direction and both have high magnitudes. This is a very rare event. In the majority of cases, $\mathbf{d}_i$ will have a small magnitude. So the user will be near the center with greater probability than near the borders. This limits the uniformity of the resultant privacy area A_i.

Forcing $\mathbf{d}_{i-1,i}$ to have the same direction as $\mathbf{d}_{i-1}$ is not a viable strategy, because it would make the centers $\mathbf{X}_i$, $\mathbf{X}_{i-1}$ and $\mathbf{X}_0$ aligned. Therefore, an adversary knowing A_{i-1} and A_i would automatically have a preferred direction where to find A_0. In general, $\mathbf{d}_{i-1,i}$ should be independent of the value of $\mathbf{d}_{i-1}$.

6.5.4 DVC-UNILO: Discrete Vector Chain UNILO

The idea of DVC-UNILO is to improve the uniformity index of VC-UNILO by changing the way the incremental vectors are built. We will first introduce the p-*Partitionability Property*, which is a weaker form of Uniformity, and then present DVC-UNILO, which offers such a property.

Ensuring the Uniformity Property is a hard problem, since it states that *all the possible* regions of the privacy area contain the user with a probability proportional to their size. A weaker requirement is to ensure this for at least *some* regions. We define p-Partitionability Property, which states that *at least* p regions, which partition the whole privacy area, have such a property. Formally:

Definition 9 (p-Partitionability Property)**.** *A privacy area A_i fulfills the p-Partitionability Property iff the partition of equally-spaced concentric annuli $\mathcal{P}(A_i) = \{\alpha_0, \ldots, \alpha_{p-1}\}$ (Fig. 6.10) divides A_i in such a way that:*

$$\forall j,\ \Pr\left[\mathbf{X} \in \alpha_j | A_i\right] = \frac{\text{size}(\alpha_j)}{\text{size}(A_i)} \tag{6.29}$$

DVC-UNILO fulfills the p-Partitionability of A_i by leveraging on the Accuracy of A_{i-1}. With reference to Figure 6.10, suppose that A_1 contains $\mathbf{X}$ and the magnitude

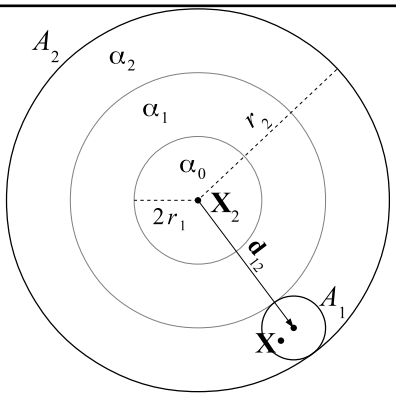


Figure 6.10. p-Partitionability regions

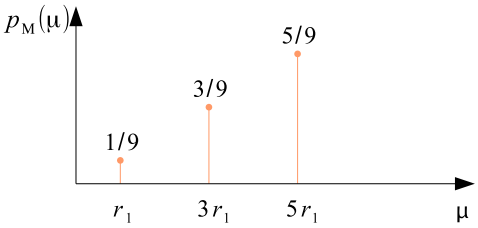


Figure 6.11. μ *pmf* of a discrete UNILO vector (example of Fig. 6.10)

of $\mathbf{d}_{1,2}$ is equal to $5r_1$. Then, we can be sure that $\mathbf{X}$ is inside the annulus α_2. If we generate such a magnitude with probability $5/9$, then $\mathbf{X}$ will be inside α_2 with the same probability, which is proportional to the size of α_2. We can repeat the same reasoning for the other annuli α_0 and α_1. The magnitude μ of the vector $\mathbf{d}_{1,2}$ becomes a discrete random variable having the *probability mass function* (*pmf*) shown in Figure 6.11. In this way, A_2 fulfills 3-Partitionability Property. Obviously this method is possible only if $2r_{i-1}$ divides exactly r_i.

By generalizing the formula we obtain the following *pmf* for the magnitude of $\mathbf{d}_{i-1,i}$:

$$p_M(\mu) = \begin{cases} (8j+4)\frac{r_{i-1}^2}{r_i^2} & \mu = (2j+1)r_{i-1} \\ 0 & \text{otherwise} \end{cases} \tag{6.30}$$

$$\text{where } \; j = 0 \ldots p-1, \quad p = \frac{r_i}{2r_{i-1}} \tag{6.31}$$

The *pmf* is depicted in Fig. 6.12. We call *discrete* UNILO *vector* a shift vector with such a magnitude and a uniform angle. We will use the following notation:

$$\mathbf{d}_{i-1,i} = \text{D-UniLO}(r_i, r_{i-1}) \tag{6.32}$$

to say that $\mathbf{d}_{i-1,i}$ is a vector created by the discrete UNILO operator with privacy radius r_i and precedent privacy radius r_{i-1}.

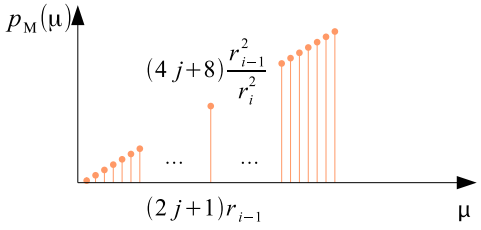


Figure 6.12. μ *pmf* of a discrete UNILO vector

If $2r_{i-1}$ does not divide r_i, DVC-UNILO will behave like VC-UNILO. The general formula for creating shift vectors is the following:

$$\mathbf{d}_i = \begin{cases} \mathrm{UniLO}(r_1, r_0) & i = 1 \\ \mathbf{d}_{i-1} + \text{D-UniLO}(r_i, r_{i-1}) & i > 1,\ r_i = 2pr_{i-1} \\ \mathbf{d}_{i-1} + \mathrm{UniLO}(r_i, r_{i-1}) & \text{otherwise} \end{cases} \tag{6.33}$$
$$\text{where } p \in \mathbb{N}$$

It is trivial to show that DVC-UNILO fulfills Accuracy and Inclusion Properties for all the privacy areas, like VC-UNILO does. In addition, we state the following:

Theorem 6. *For each privacy area A_i having $i \geq 2$ and $r_i = 2pr_{i-1}$,* DVC-UNILO *fulfills p-Partitionability Property.*

Proof. In the following, $\mu = \|\mathbf{d}_{i-1,i}\|$, and $\mu_j = (2j+1)r_{i-1}$. Let us compute the probability $\Pr[\mathbf{X} \in \alpha_j]$.

$$\Pr[\mathbf{X} \in \alpha_j] = \tag{6.34}$$
$$= \Pr[\mu = \mu_j] \cdot \Pr[\mathbf{X} \in \alpha_j | \mu = \mu_j] + \tag{6.35}$$
$$+ \quad \Pr[\mu \neq \mu_j] \cdot \Pr[\mathbf{X} \in \alpha_j | \mu \neq \mu_j] \tag{6.36}$$

If A_{i-1} enjoys Accuracy Property and $\mu = \mu_j$, then the user will surely be in annulus α_j. Thus, $\Pr[\mathbf{X} \in \alpha_j | \mu = \mu_j] = 1$. On the other hand, $\Pr[\mathbf{X} \in \alpha_j | \mu \neq \mu_j] = 0$ for the same reason. Hence:

$$\Pr[\mathbf{X} \in \alpha_j] = \Pr[\mu = \mu_j] \tag{6.37}$$
$$= (8j+4)\frac{r_{i-1}^2}{r_i^2} \tag{6.38}$$
$$= \frac{\mathrm{size}(\alpha_j)}{\mathrm{size}(A_i)} \tag{6.39}$$

□

DVC-UNILO is an improvement of VC-UNILO. It fulfills the Inclusion Property and offers a better uniformity.

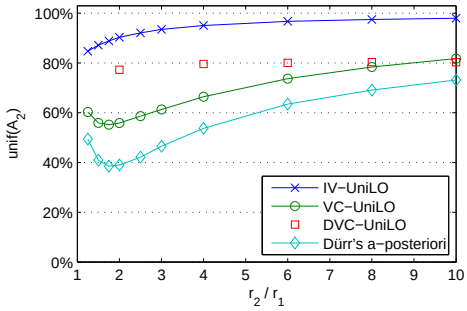


Figure 6.13. $\mathrm{unif}\,(A_2)$ with $r_1 = 10r_0$

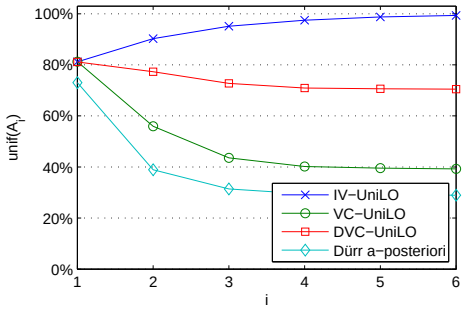


Figure 6.14. $\mathrm{unif}\,(A_i)$ with $r_1 = 10r_0$ and $r_i = 2r_{i-1}$

6.5.5 Uniformity analysis

We performed Monte Carlo simulations to compute the uniformity indexes of the privacy areas produced by IV-UNILO, VC-UNILO and DVC-UNILO under different conditions. We also compared them with "*a-posteriori* share generation algorithm" by Dürr et al. [37] which offers multiple levels of privacy with a perturbation approach. Dürr used a noise uniform in angle and uniform in magnitude to obfuscate user's positions. Actually, Dürr's algorithm dealt only with location measurements with infinite precision ($r_0 = 0$). To make meaningful comparisons, we adapted it to deal with finite-precision localization technologies. This is easily done by creating shift vectors with maximum magnitude equal to $r_1 - r_0$, as UNILO-based algorithms do. We simulated a localization technology with $r_0 = (1/10)r_1$. Tests showed that our algorithms outperform Dürr's ones in terms of uniformity.

Figure 6.13 shows the uniformity index of the second-level privacy area A_2 wrt r_2/r_1, with a precise localization technology. Note that IV-UNILO gets closer to the optimum than all the other methods. DVC-UNILO improves the performance of VC-UNILO when r_2/r_1 is not too large. Dürr's algorithm performs always worse than UNILO-based algorithms.

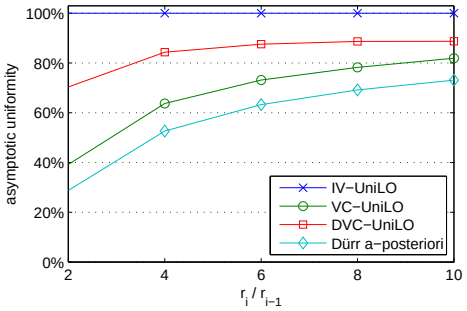


Figure 6.15. $\mathrm{unif}\,(A_\infty)$

The performance of DVC-UNILO remains high at higher levels of privacy. Figure 6.14 shows the uniformity indexes of A_1–A_6 with $r_i = 2r_{i-1}$ $(i > 1)$. The tests revealed that all the four methods approach constant values at higher privacy levels: 28.8% for Dürr's algorithm, 39.2% for VC-UNILO, 70.4% for DVC-UNILO, and 100.0% for IV-UNILO. The asymptotic value of the uniformity index $\mathrm{unif}\,(A_\infty)$ depends only on the algorithm employed and on the radius ratio r_i/r_{i-1}. Figure 6.15 shows $\mathrm{unif}\,(A_\infty)$ wrt the radius ratio. We can easily see that UNILO-based algorithms outperforms Dürr's obfuscation algorithm.

To sum up, in order to guarantee an optimal level of uniformity, the privacy radius set must be configured wisely. In particular, it is always better to set the first privacy radius far greater than the precision radius of the sensor ($r_1 \gg r_0$). In addition, if we want to defend against collusion attacks, it is better to use DVC-UNILO and set each privacy radius to be the double or quadruple of the previous one. In this way, we have both a good granularity on the privacy radii, and a good uniformity index, which tends to 70%–84% (with collusion resistance) or 100.0% (without collusion resistance) with the growing of i.

6.6 Service example

UNILO operators have the advantage to be transparent to the service provider, in the sense that a privacy area has the same properties as an ordinary measurement. A service provider designed for receiving non-obfuscated inputs can be seamlessly adapted for receiving UNILO-obfuscated inputs.

We will describe now an example social application, called "close friends", in which users share their obfuscated positions with their friends. Alice wants to find out which of her friends are in her proximity. We define "being in the proximity of Alice" as "being at a distance of 400 meters or less from Alice". The service provider gathers the obfuscated positions of Alice's friends and sends them to Alice. While Alice knows her own position, the locations of her friends are obfuscated. Suppose Bob is one of

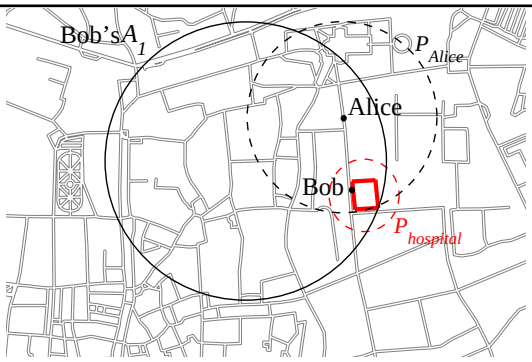


Figure 6.16. "Close friends" application

Alice's friends. Since Alice does not know his exact location, the question "is Bob in my proximity?" will necessarily have a probabilistic answer.

The problem can be modeled as depicted in Fig. 6.16. Alice builds a circle centered on its position and with 400 meters of radius (*proximity area*, P_{Alice}), and computes the intersection between that area and the privacy area of Bob (A_1). If Bob is inside this intersection, he will be in Alice's proximity. The probability that such an event happens is:

$$\Pr\left[\text{Bob} \in P_{\text{Alice}}\right] = \iint_{P_{Alice} \cap A_1} f_{X,Y}\left(x, y\right) \, \mathrm{d}x\mathrm{d}y \tag{6.40}$$

To make such a calculus, Alice should perform a statistical analysis of Bob's position and then compute numerically the integral. This operation is quite inefficient. However, if A_1 is assumed to be Uniform and Accurate, Eq. 6.40 will simplify in:

$$\Pr\left[\text{Bob} \in P_{\text{Alice}}\right] \approx \frac{\text{size}(P_{\text{Alice}} \cap A_1)}{\text{size}(A_1)} \tag{6.41}$$

Alice performs this calculus only for each friend whose $\mathbf{X}_1$ is nearer than $r_1 + 400\,\mathrm{m}$. The others have no intersection, and thus 0% probability.

6.6.1 Utility analysis

We evaluated the utility of the presented obfuscation operators in our example "close friends" application. Our utility metric is the mean *uncertainty* in the service's answer. We define the uncertainty as the absolute difference between the computed proximity probability and the true answer, i.e. 1 if the friend is close, 0 otherwise. More formally, if P_{Alice} is the proximity area of Alice and A_i is the privacy area of Bob:

$$\text{uncert}(A_i) = \left| \frac{\text{size}(P_{\text{Alice}} \cap A_i)}{\text{size}(A_i)} - \text{prox}(\text{Bob}) \right| \tag{6.42}$$

$$\text{prox}(\text{Bob}) = \begin{cases} 1 & \text{if Bob is in the proximity} \\ 0 & \text{otherwise} \end{cases} \tag{6.43}$$

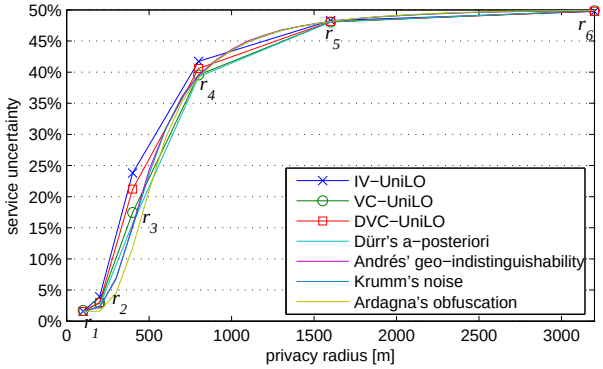


Figure 6.17. Uncertainty of "close friends" service (1000 Monte Carlo runs for each point)

Low values of uncertainty mean that the computed answers are close to the true answers. In the simulations, Bob's position is taken in Alice's proximity with 50% probability. Locations are measured with $r_0 = 10\,\mathrm{m}$, and the privacy radii follow a geometric progression $\rho = \{100\,\mathrm{m}, 200\,\mathrm{m}, 400\,\mathrm{m}, \ldots\}$. Figure 6.17 shows the mean uncertainty of Alice using the "close friends" service, versus the privacy preferences of her friends. We can see that the uncertainty depends mainly on the size of the privacy area, and only marginally on the obfuscation operator. For $i > 5$, corresponding to a privacy radius $r_5 = 1.6\,\mathrm{Km}$, the obfuscated positions lose their utility in determining the proximity. This suggests that for this kind of service, $i \in [0, 5]$ is a suitable range of privacy preferences.

Together with the utility, it is interesting to measure the error that Alice makes in considering the privacy areas as uniform when they ar not. Many privacy-aware services [8, 18] postulates the uniformity, rather than providing it. In practice, they use an approximate calculus (Equation 6.41) instead of an exact one (Equation 6.40). The impact of such an approximation can be quite high, if the obfuscation does not provide for Uniformity and Accuracy Properties. We evaluated this by measuring the mean *service error*, i.e. the mean absolute difference between the probability computed with and without the approximation. More formally:

$$\mathrm{error}(A_i) = \left| \frac{\mathrm{size}(P_{\mathsf{Alice}} \cap A_i)}{\mathrm{size}(A_i)} - \Pr\left[\mathsf{Bob} \in P_{\mathsf{Alice}}\right] \right| \tag{6.44}$$

Low values of service error mean that the computed proximity probabilities are close to the real ones. We compared UniLO algorithms with other noises, namely Dürr's "*a-posteriori* share generation algorithm" [37], Krumm's noise [59], and Ardagna's obfuscation operators [8]. Figure 6.18 shows the results of the simulations. We can see that the service's mean error depends mainly on the uniformity of the obfuscation noise. IV-UniLO and DVC-UniLO perform near to the optimum of 0% error, as they are highly uniform and they respect Accuracy Property. On the contrary, Ardagna's

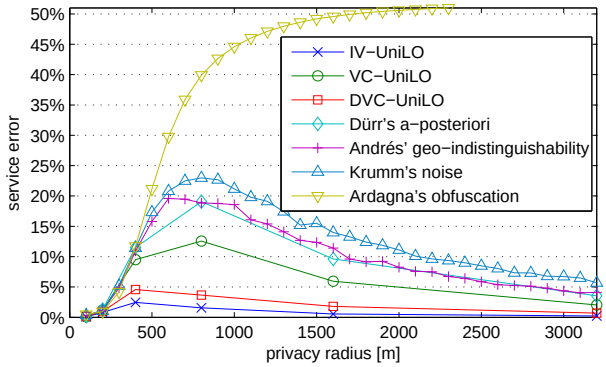


Figure 6.18. Error of "close friends" service (1000 Monte Carlo runs for each point)

obfuscation performs particularly bad, because it respects neither Uniformity nor Accuracy.

6.7 Inference adversary

In previous sections we used the concept of agnostic adversary to measure the uniformity of an obfuscation operator. Now we introduce a more realistic adversary, which owns auxiliary information (the map), and we show that a better uniformity improves the resistance against such a threat.

The *inference adversary* tries to infer sensitive information from the user's position and other auxiliary information. Let us suppose that there is a "sensitive point" on the map, in the sense that the proximity to that point can allow the adversary to infer sensitive information about the user. An example of that could be a hospital for the cancer treatment. If Bob sends his position from inside or from the close proximity of the hospital, the adversary could easily infer his health condition. Such an adversary could be the "close friends" service provider, or Alice herself. Let us suppose that Bob actually is in the hospital, and that the adversary knows his privacy area. The adversary performs a statistical analysis of Bob's position, knowing the localization technology and the obfuscation operator employed. Then she uses her auxiliary information by excluding those zones that cannot contain users (inside walls, rivers, etc.). The result of this analysis is a probability distribution over the map. Finally, the adversary computes the probability that Bob is in the hospital close proximity, say, inside a proximity area P_{hospital} of 200 meters of radius (cfr. Fig. 6.16). If such a probability is 50% or more, the adversary successfully infers the health condition of Bob.

We evaluated the success probability of the inference adversary on a real map of Pisa city center, extracted from public OpenStreetMap data [75]. Figure 6.19 shows the probability that the adversary has in guessing the health condition of Bob wrt his privacy radius. We can see that UniLO algorithms offer perfect protection even

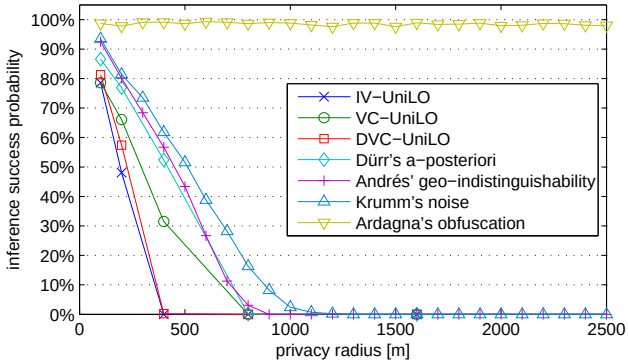


Figure 6.19. Success of inference attack (1000 Monte Carlo runs for each point)

for small privacy radii (400 meters for IV-UniLO and DVC-UniLO). Ardagna et al.'s obfuscation offers no protection against inference attacks.

7

Advanced techniques for obfuscation-based location privacy

In this chapter we present an advanced location obfuscation system capable of dealing with measurement imprecision, multiple levels of privacy, untrusted servers, and adversarial knowledge of the map. We study its resistance against deobfuscation attacks, and we improve it by means of three advanced techniques, namely *extreme vectors*, *enlarge-and-scale*, and *hybrid vectors*. Our tests show that extreme vectors can decrease the adversarial success probability by 22.50%, enlarge-and-scale technique by 31.74%, and hybrid vectors by 17.17%. The present chapter is both an integration and a follow-up of the share generation methods presented in [37], the technique to resist against map-aware adversaries presented in [91], and the obfuscation technique for position measurement imprecision and single level of privacy presented in Section 6.4. Such works have been improved in terms of resistance by means of extreme vectors, enlarge-and-scale, and hybrid vectors.

The rest of the chapter is organized as follows. Section 7.1 describes in detail our basic obfuscation system. Section 7.2 models our adversary and introduces a metric to evaluate the resistance against deobfuscation attacks. Sections 7.3, 7.4, and 7.5 describe respectively extreme vectors, enlarge-and-scale, and hybrid vectors techniques, and evaluate their relative resistance improvement on the basic obfuscation system.

7.1 Basic obfuscation system

We describe here our basic obfuscation system, which follows the approach presented in [37], integrated with [91] for adversaries holding map information, and with the obfuscation technique presented in Section 6.4 for measurement imprecision.

From now on, the notation:

$$A = \mathrm{circle}(\mathbf{X}, r)$$

will mean that A is a circle with center $\mathbf{X}$ and radius r. The *real position* of the user is a point $\mathbf{X} \in \mathbb{R}^2$. The *measured position* is a point $\mathbf{X}_m$. The *error radius* (r_m)

quantifies the precision of the positioning technology. The circle $A_m = \mathrm{circle}(\mathbf{X}_m, r_m)$ contains the real position, and it is called the *measurement area*. We suppose that the system knows the error radius. If the positioning technology does not give this information, the system can suppose an error radius, basing on the average precision of that technology.

By means of the obfuscation process, the user's position is hidden inside an *obfuscation area*, with a larger size than the measurement area. The position of a user could be contemporaneously accessed by many location-based services, and the user may require different levels of privacy for them. For example, more obfuscation could be suitable for less trusted services or for services requiring less precision.

A solution is to *independently* generate many obfuscation areas with different size, and store them on a *location server*. The location server will in turn release to the *service provider* the obfuscation area corresponding to its access rights. With n levels of privacy, we generate n obfuscation areas:

$$A_o^{(k)} = \mathrm{circle}(\mathbf{X}_o^{(k)}, r_o^{(k)}) \text{ with } 0 \leq k \leq n-1 \tag{7.1}$$

where k is the *precision index*. The higher the precision index is, the more precise the obfuscation area, and the lower the privacy level. More trusted service providers will be allowed to access to obfuscation areas with higher precision indexes. We consider the measurement area itself as the obfuscation area with precision index n:

$$A_o^{(n)} = A_m \tag{7.2}$$

So we actually have $n+1$ precision indexes, from 0 to n. The point $\mathbf{X}_o^{(k)}$ is the *k-th obfuscated position*, the radius $r_o^{(k)}$ is the *k-th obfuscation radius*, and the area $A_o^{(k)}$ is the *k-th obfuscation area*. The obfuscation radii follow a decreasing progression:

$$r_o^{(k)} = \begin{cases} r_o^{(0)}/n \cdot (n-k) & \text{if } 1 \leq k \leq n-1 \\ r_m & \text{if } k = n \end{cases} \tag{7.3}$$

This solution is simple and scalable. However, if the obfuscation areas are stored in a single location server, the user is forced to trust such a server. We solve this problem by splitting the whole set of obfuscation areas in pieces (*shares*), and then storing them in several location servers, each of which is not required to be trustworthy. The obfuscated positions are randomly generated by means of n concatenated random vectors called *obfuscation vectors* ($\mathbf{d}_o^{(k)}$). The obfuscation vectors form a chain that connects all the obfuscated positions, from the 0-th one to the measured position (Fig. 7.1). The largest obfuscation area constitutes the *master share*, and the n obfuscation vectors the *refinement shares*. The k-th obfuscation area can be reconstructed by combining the master share with the first k refinement shares. The share combination is done by reducing the obfuscation radius (following Eq. 7.3) and composing the obfuscation vectors:

$$\mathbf{X}_o^{(k)} = \mathbf{X}_o^{(0)} + \sum_{i=1}^{k} \mathbf{d}_o^{(k)} \tag{7.4}$$

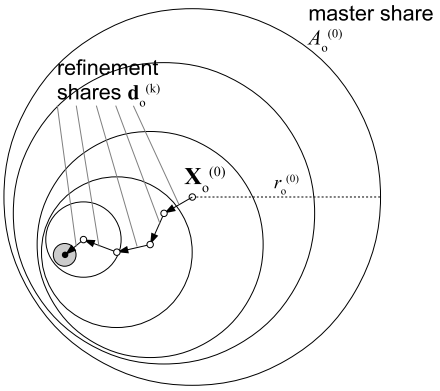


Figure 7.1. Obfuscation scheme

The user generates the master share and the refinement shares, and distributes these

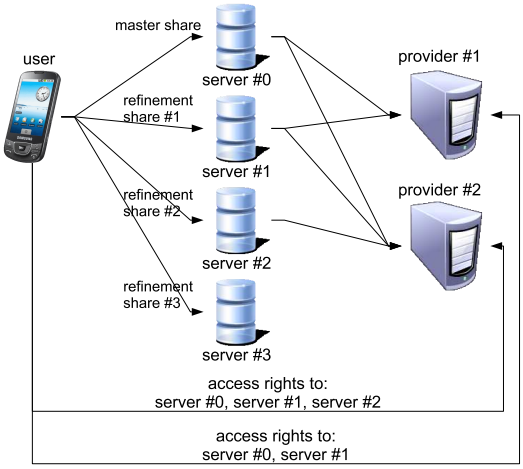


Figure 7.2. Architecture for multiple levels of privacy

$n + 1$ pieces of information to $n + 1$ location servers (Fig. 7.2). To grant a service provider the access to the k-th obfuscation area, the user simply grants him access to the first $k + 1$ location servers. The service provider retrieves the correspondent shares, and compose them to obtain back the obfuscation area.

This obfuscation system enjoys the property that neither the service providers nor the location servers have to be trusted by the user, as no one of these entities has the complete set of information. If the master share and k refinement shares get compromised, the adversary will know the position at the precision index k at most. In

other words, the user's privacy degrades gracefully with the number of compromised shares.

7.1.1 Share generation methods

In [37], we used two possible methods for the generation of the shares: *a-posteriori share generation method*, and *a-priori share generation method*. The a-posteriori one first generates the refinement shares, and then the master share. The a-priori one does vice versa.

Both methods use the concept of r*-bounded uniform vector* as a fundamental building block.

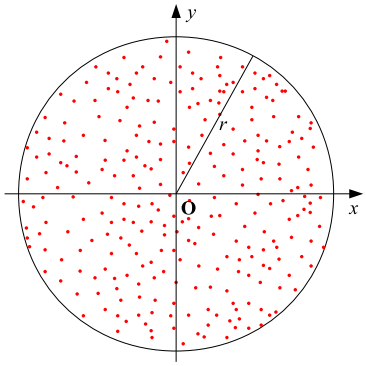


Figure 7.3. Spacial probability distribution of an r-bounded uniform vector

Definition 10. *An* r-bounded uniform vector *(Fig. 7.3) is a random vector* $\mathbf{d}$ *such that* $\|\mathbf{d}\| \leq r$*, and its spacial probability distribution is uniform inside* $\mathrm{circle}(\mathbf{O}, r)$*, where* $\mathbf{O}$ *indicates the axis origin.*

In the a-posteriori method, the obfuscation vectors are generated as n independent random vectors. Then, the 0-th obfuscated position is computed by subtracting them to the measured position. The random vectors are bounded uniform vectors, whose bounds are the following:

$$\left\|\mathbf{d}_o^{(k)}\right\| \leq \begin{cases} r_o^{(0)}/n & \text{if } 1 \leq k \leq n-1 \\ r_o^{(0)}/n - r_m & \text{if } k = n \end{cases} \tag{7.5}$$

Note that, on the contrary of the original algorithm from [37], the last obfuscation vector is bounded by $r_o^{(0)}/n - r_m$ (instead of $r_o^{(0)}/n$). Otherwise, depending on the measurement error, the real position could lie outside the obfuscation areas. Algorithm 1 shows the a-posteriori share generation method.

Algorithm 1 A-posteriori share generation method

1: **procedure** APOSTERIORI($A_m, r_o^{(0)}, n$)
2: **for** $i = 1 \to (n-1)$ **do**
3: $\mathbf{d}_o^{(i)} \leftarrow (r_o^{(0)}/n)$**-bounded uniform vector**
4: **end for**
5: $\mathbf{d}_o^{(n)} \leftarrow (r_o^{(0)}/n - r_m)$**-bounded uniform vector**
6: $\mathbf{X}_o^{(0)} \leftarrow \mathbf{X}_m - \sum_{i=1}^{n} \mathbf{d}_o^{(i)}$
7: **return** $\left\{A_o^{(0)}, \mathbf{d}_o^{(1)}, \ldots, \mathbf{d}_o^{(n)}\right\}$
8: **end procedure**

The advantage of this algorithm is that the obfuscation vectors are generated independently, so that an adversary who knows some of them takes no advantage in predicting the other ones. The main drawback is that the probability density of the user position inside the 0-th obfuscation area is very biased, as shown in [37].

In the a-priori method, we first generate the 0-th obfuscation area by means of an $(r_o^{(0)} - r_m)$-bounded uniform vector $\mathbf{d}_o^*$, called *master obfuscation vector*. Note that, on the contrary of the original algorithm from [37], the master obfuscation vector is bounded by $r_o^{(0)} - r_m$ (instead of $r_o^{(0)}$). Otherwise, depending on the measurement error, the real position could lie outside the obfuscation area. Then, we generate the refinement shares by means of a *random decomposition* of the master obfuscation vector. More formally:

Definition 11. *Given* $\mathbf{d}_o^*$ *such that* $\|\mathbf{d}_o^*\| \leq r_o^{(n)} - r_m$*, we call* random decomposition *of* $\mathbf{d}_o^*$ *a set of random sub-vectors:*

$$\left\{\mathbf{d}_o^{(1)}, \mathbf{d}_o^{(2)}, \ldots, \mathbf{d}_o^{(n)}\right\}$$

such that:

$$\sum_{i=1}^{n} \mathbf{d}_o^{(i)} = \mathbf{d}_o^* \tag{7.6}$$

and:

$$\left\|\mathbf{d}_o^{(k)}\right\| \leq \begin{cases} r_o^{(0)}/n & \text{if } 1 \leq k \leq n-1 \\ r_o^{(0)}/n - r_m & \text{if } k = n \end{cases} \tag{7.7}$$

Algorithm 2 shows an efficient way to implement the random decomposition with bounded uniform sub-vectors. It generates the first $n-1$ vectors in such a way, at each step, the remaining distance (l) is coverable by the remaining vectors (Alg. 2 Line 6). The last vector is generated as a difference, to reach exactly the master obfuscation vector (Line 7). Algorithm 3 shows the a-priori share generation method.

The main advantage of this algorithm is that the 0-th obfuscated position is generated directly, and not as a sum of previously generated random vectors. Thus, it is

Algorithm 2 Random decomposition

procedure DECOMPOSE($\mathbf{d}_o^*, r_m, r_o^{(0)}, n$)
  $\mathbf{d}_{sum} = 0$
  for $i = 1 \to (n-1)$ **do**
    $l \leftarrow (n-i)r_o^{(0)}/n - r_m$
    $\mathbf{d}_o^{(i)} \leftarrow (r_o^{(0)}/n)$**-bounded uniform vector,**
    such that: $\mathrm{dist}\left(\mathbf{d}_{sum} + \mathbf{d}_o^{(i)}, \mathbf{d}_o^*\right) \leq l$
    $\mathbf{d}_{sum} \leftarrow \mathbf{d}_{sum} + \mathbf{d}_o^{(i)}$
  end for
  $\mathbf{d}_o^{(n)} \leftarrow \mathbf{d}_o^* - \mathbf{d}_{sum}$
  return $\left\{\mathbf{d}_o^{(1)}, \ldots, \mathbf{d}_o^{(n)}\right\}$
end procedure

Algorithm 3 A-priori share generation method

procedure APRIORI($A_m, r_o^{(0)}, n, M$)
  $\mathbf{d}_o^* \leftarrow (r_o^{(0)} - r_m)$**-bounded uniform vector**
  $\mathbf{X}_o^{(0)} \leftarrow \mathbf{X}_m - \mathbf{d}_o^*$
  $\left\{\mathbf{d}_o^{(1)}, \ldots, \mathbf{d}_o^{(n)}\right\} \leftarrow \mathrm{decompose}(\mathbf{d}_o^*, r_m, r_o^{(0)}, n)$
  return $\left\{A_o^{(0)}, \mathbf{d}_o^{(1)}, \ldots, \mathbf{d}_o^{(n)}\right\}$
end procedure

possible to control it in such a way that the real user position gets uniformly distributed. The disadvantage is that the obfuscation vectors generated by decomposition are not probabilistically independent of each other. Thus, an adversary knowing one or more of them is helped in predicting the others.

In summary, the a-priori method is less resistant in case of more powerful adversaries, which already know some refinement shares. The a-posteriori method is less resistant in case of less powerful adversaries, which know zero or few refinement shares.

7.1.2 Enlarge-and-perturb method for map awareness

Until now, we took into consideration a user who moves completely free in space. In the real life, people's movements are constrained by the presence of walls, buildings, and other obstacles. An adversary who owns information about the map where the users are moving is more powerful. She can cut away from the obfuscation area the zones where the user cannot be, thus finding a *map-reduced obfuscation area*, with a smaller size. In order to guarantee a nominal level of privacy in the presence of a map-aware adversary, the obfuscation system must be map-aware too.

For our present purposes, a *map* (M) is a subset of the $\mathbb{R}^2$ space. A point is inside the map if and only if it represents a possible position for the user. For the sake of

simplicity, let us consider by now a single level of privacy ($n = 1$). We will generalize to multiple levels afterwards. Let us suppose that $r_o^{(0)}$ is the "nominal" obfuscation radius desired by the user. The obfuscation system generates an obfuscation area ($A_o^{(0)}$) whose size ($(r_o^{(0)})^2 \cdot \pi$) is the *nominal obfuscation precision*. The adversary computes a *map-reduced obfuscation area*:

$$A_M^{(0)} = A_o^{(0)} \cap M \tag{7.8}$$

which contains the real position of the user. In doing so, the adversary improves her precision with respect to the nominal obfuscation precision:

$$\text{size}\left(A_M^{(0)}\right) \leq (r_o^{(0)})^2 \cdot \pi \tag{7.9}$$

We have thus to compensate somehow this "precision gain".

In [91], we used a simple solution for this, that we call here *enlarge-and-perturb*. Enlarge-and-perturb technique first enlarges the obfuscation radius, in such a way that the size of the map-reduced obfuscation area is equal to the nominal one. Algorithm 4 shows an efficient way to do that. We employ a logarithmic search to minimize the number of area intersections (Alg. 4, Line 10). The logarithmic search stops when the map-reduced obfuscation area reaches the nominal size, with a tolerance (δ_A), that we fixed to be 1% of the nominal size:

$$\delta_A = 0.01 \cdot r_o^2 \cdot \pi \tag{7.10}$$

We indicate with $r_o'^{(0)}$ and $A_o'^{(0)}$ respectively the *enlarged obfuscation radius* and the *enlarged obfuscation area*.

Enlarging the radius is obviously not enough, as the adversary could simply narrow it again. The second phase is in fact to *perturb* the obfuscated position with an $(r_o'^{(0)} - r_o^{(0)})$-bounded uniform vector. Figure 7.4 shows an example of enlarge-and-perturb operation. $\mathbf{d}_p$ is the perturbation vector, while $\mathbf{X}'_o$ is the new obfuscated position after the perturbation. Note that, after having changed the obfuscated position, the obfuscation area has changed as well. Therefore, we have to check again if the size of the map-reduced obfuscation area has become smaller. If it has, we perform an additional enlargement and an additional perturbation, and so on. In case of $n > 1$ levels of privacy, we repeat the whole procedure for each level. Algorithm 5 shows the complete enlarge-and-perturb technique.

7.2 Adversary model

We consider an adversary who wants to *deobfuscate* a previously obfuscated position, and derive the original position of the user from it. An obfuscation function cannot be deterministically inverted, since it involves random noise. However, the adversary can perform a *statistical analysis*, to find out the spatial probability distribution of the

Algorithm 4 Radius enlargement algorithm

procedure ENLARGE($\mathbf{X}_o, r_o, M$)
  $r_{\mathrm{lo}} \leftarrow r_o$
  $r_{\mathrm{hi}} \leftarrow r_o$
  repeat// search for a higher bound to r'_o:
    $r_{\mathrm{hi}} \leftarrow \sqrt{2} \cdot r_{\mathrm{hi}}$
    $A_M \leftarrow \mathrm{circle}(\mathbf{X}_o, r_{\mathrm{hi}}) \cap M$
  until $\mathrm{size}\,(A_M) \geq r_o^2 \cdot \pi$
  loop// logarithmic search for r'_o:
    $r_{\mathrm{md}} \leftarrow \sqrt{(r_{\mathrm{lo}}^2 + r_{\mathrm{hi}}^2)/2}$
    $A_M \leftarrow \mathrm{circle}(\mathbf{X}_o, r_{\mathrm{md}}) \cap M$
    if $\left|\mathrm{size}\,(A_M) - r_o^2 \cdot \pi\right| \leq \delta_A$ **then**
      return r_{md}
    else if $\mathrm{size}\,(A_M) < r_o^2 \cdot \pi$ **then**
      $r_{\mathrm{lo}} \leftarrow r_{\mathrm{md}}$
    else
      $r_{\mathrm{hi}} \leftarrow r_{\mathrm{md}}$
    end if
  end loop
end procedure

Algorithm 5 Enlarge-and-perturb technique

procedure ENLARGE-AND-PERTURB($A_o^{(0)}, \mathbf{d}_o^{(1)}, \ldots, \mathbf{d}_o^{(n)}, M$)
  $S \leftarrow \{1, \ldots, n\}$
  for all i**:** $r_o'^{(i)} \leftarrow r_o^{(i)}$ **and** $\mathbf{X}_o'^{(i)} \leftarrow \mathbf{X}_o^{(i)}$
  repeat
    for all $i \in S$ **do** // Enlargement:
      $r_o'^{(i)} \leftarrow \mathrm{enlarge}(\mathbf{X}_o'^{(i)}, r_o'^{(i)}, M)$
    end for
    for $i = 1 \rightarrow n$ **do** // Perturbation:
      $\mathbf{d}_p \leftarrow (r_o'^{(i)} - r_o^{(i)})$**-bounded uniform vector**
      $\mathbf{d}_o'^{(i)} \leftarrow \mathbf{d}_o^{(i)} + \mathbf{d}_p$
    end for
    for all i**: compute** $\mathbf{X}_o'^{(i)}$ **and** $A_o'^{(i)}$ **from** $\mathbf{d}_o'^{(i)}$
    $S \leftarrow \left\{ i : \mathrm{size}\left(A_o'^{(i)} \cap M\right) < \pi \cdot \left(r_o^{(i)}\right)^2 \right\}$
  until $S = \emptyset$
  return $A_o'^{(0)}, \mathbf{d}_o'^{(1)}, \ldots, \mathbf{d}_o'^{(n)}$
end procedure

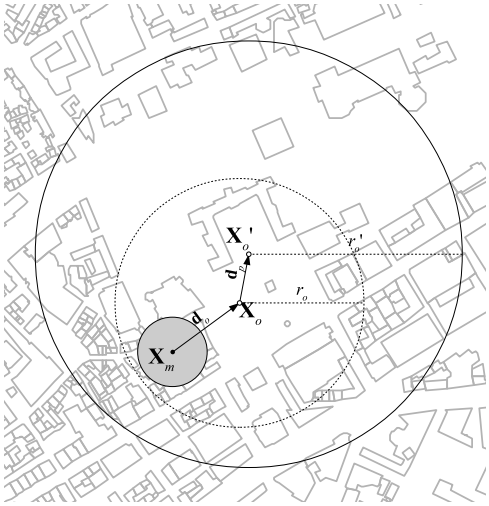


Figure 7.4. Enlarge-and-perturb technique

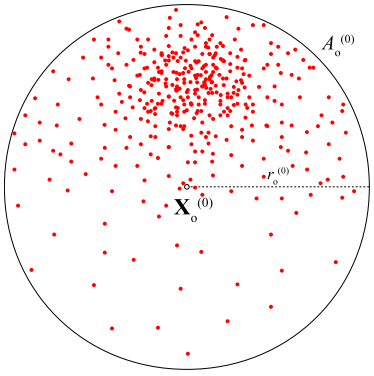


Figure 7.5. Statistical analysis example

real position inside the obfuscation area. If such a distribution is uniform, the user's position is unpredictable. Otherwise, if it presents pronounced concentrations (Fig. 7.5), the adversary can suppose the user's position is more likely to be in certain zones than in others.

In order to measure the resistance against a statistical analysis, we have to quantify the unpredictability of the user's position, i.e. the uniformity of its probability distribution. Let us suppose that the adversary identifies an area (*deobfuscation area*), which contains the user with a certain probability (*deobfuscation probability*). The best strategy for the adversary is to choose the deobfuscation area comprising the zones with the highest probability concentrations. We fix the size of the deobfuscation area to be 10% of the size of the obfuscation area. Figure 7.6 shows a single-dimension analogy. The outer and the inner segments represent respectively the obfuscation

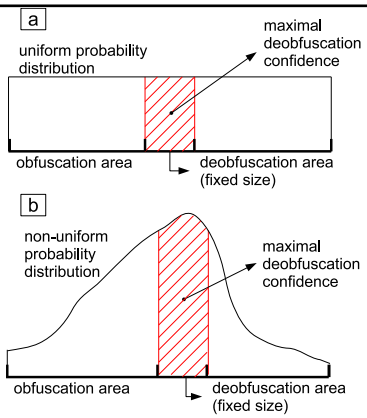


Figure 7.6. Single-dimension analogy of deobfuscation area

area and the deobfuscation area. The red area under the probability distribution represents the deobfuscation probability. The adversary is free to move the deobfuscation area in order to maximize the deobfuscation probability. Our resistance metric is the *maximal deobfuscation probability* (P_{deobf}).

Definition 12. *Given an obfuscation area $A_o^{(k)}$, the* maximal deobfuscation probability *is:*

$$P_{deobf} = \max_{A_o^{10\%}} \Pr\left[\mathbf{X} \in A_o^{10\%}\right] \tag{7.11}$$

where $A_o^{10\%}$ is a deobfuscation area such that:

$$\text{size}\left(A_o^{10\%}\right) = 10\% \cdot \text{size}\left(A_o^{(k)}\right) \tag{7.12}$$

Such a metric depends on the probability distribution of the user's position inside the obfuscation area. If it is uniform, like in Figure 7.6a, then the maximal deobfuscation probability will be minimal. I.e., it will be exactly 10%, since we fixed the deobfuscation area to be 10% of the obfuscation area. This is the best case, when the real position is completely unpredictable inside the obfuscation area. If the probability density is not uniform, like in Figure 7.6b, the adversary will have a maximal deobfuscation probability greater than 10%. The bigger the maximal deobfuscation probability is, the more predictable the user's position.

7.2.1 Malicious provider and malicious server

We model two possible kinds of adversary: the *malicious provider* and the *malicious server*. The malicious provider models a service provider which illegitimately tries to gain more precision than she is permitted to have. In this case, the adversary knows the master share and a set of $k_{adv} < n$ refinement shares she has the right to access.

The refinement shares get compromised in order, i.e. $\mathbf{d}_o^{(1)}, \ldots, \mathbf{d}_o^{(k_{adv})}$, because this is the order in which the user grants the access to them. In case of two or more colluding service providers, they will be modeled as a single adversary enjoying the widest access privilege among the colluding entities. The malicious provider first combines the shares and obtains the k_{adv}-th obfuscation area. Then she tries to deobfuscate such an area by means of statistical analysis.

On the contrary, the malicious server models a location server or a group of colluding location servers that want to use the shares they are storing for illegitimate purposes. It models also an external adversary who hacks one or more location servers and steals their shares. We consider the master share to be always compromised, otherwise no statistical attack is possible. In addition, we assume that $k_{adv} < n$ refinement shares are compromised, *with no particular order*. In this case, the adversary could miss one or more obfuscation vectors necessary to reconstruct the k_{adv}-th obfuscation area. However, we assume that she composes anyway the obfuscation vectors she has, thus obtaining an *alternative obfuscation area* A_o^*. If we call $\mathcal{D}$ the set of the compromised obfuscation vectors, the adversary computes the alternative obfuscation area in the following way:

$$A_o^* = \mathrm{circle}(\mathbf{X}_o^*, r_o^*) \tag{7.13}$$

$$\mathbf{X}_o^* = \mathbf{X}_o^{(0)} + \sum_{\mathcal{D}} \mathbf{d}_o^{(i)} \tag{7.14}$$

$$r_o^* = \begin{cases} r_o^{(0)}/n \cdot (n - k_{adv}) + r_m & \text{if } \mathbf{d}_o^{(1)} \in \mathcal{D} \\ r_o^{(0)}/n \cdot (n - k_{adv}) & \text{otherwise} \end{cases} \tag{7.15}$$

From the geometrical properties of the obfuscation vectors, the alternative obfuscation area contains the user's position. We assume the adversary performs the statistical analysis over this obfuscation area.

7.3 Extreme vectors

As we said in Section 7.1, a-posteriori and a-priori share generation methods both have some drawbacks that limit the unpredictability of the obfuscation. In particular, the drawback of a-posteriori share generation method is that the probability density of the real position inside the 0-th obfuscation area is very biased. Indeed, since we add n independent random vectors to generate it, the 0-th obfuscated position will follow the law of the Central Limit Theorem. As n grows, the real position of the user will tend to follow a Gaussian probability distribution inside the 0-th obfuscation area. Moreover, the larger n is, the more concentrated the probability at the center of the area will be. As $n \to \infty$, the real position's probability distribution tends to be a Dirac delta.

On the other hand, the drawback of a-priori share generation method is that the generated obfuscation vectors are not probabilistically independent of each other.

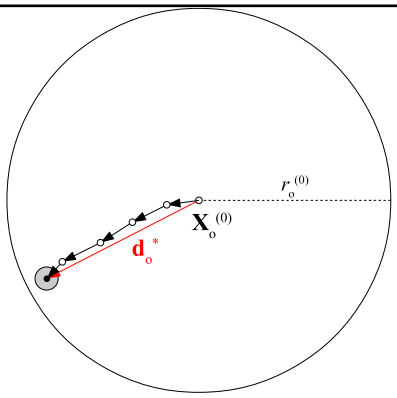


Figure 7.7. Constrained random decomposition

In fact, especially if the length of the master obfuscation vector is close to the limit $r_o^{(0)} - r_m$, the sub-vectors will be constrained to be long and to follow the same direction (Fig. 7.7). In other words, the obfuscation vectors are *correlated*. An adversary knowing one or more of them is helped in predicting the others.

We now introduce *extreme vectors*, an alternative to classic uniform vectors which significantly alleviates both these drawbacks, thus improving the uniformity of both obfuscation algorithms.

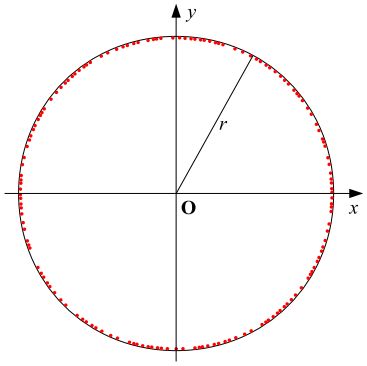


Figure 7.8. Spacial probability distribution of an r-bounded extreme vector

Definition 13. *An* r-bounded extreme vector *(Fig. 7.8) is a random vector* $\mathbf{d}$ *such that* $\|\mathbf{d}\| = r$*, and its spacial probability distribution is uniform on the border of* $\mathrm{circle}(\mathbf{O}, r)$*.*

Note that extreme vectors by themselves are more predictable than uniform ones, because they are distributed on the circumference instead of inside the whole circle. However, they enjoy two good properties:

1. *Uniform composition.* A sum of extreme vectors is less predictable than a sum of uniform vectors. This property can improve the resistance of a-posteriori share generation method.
2. *Uncorrelated decomposition.* A random decomposition in extreme sub-vectors is less correlated than a random decomposition in uniform vectors. This property can improve the resistance of a-priori share generation method.

The reason for Property 1 is that extreme vectors help spreading the probability distribution toward the borders of the obfuscation area. This avoids the concentration at the center, thus improving the uniformity. More formally, an extreme vector has a standard deviation ($\sigma = 1/\sqrt{2}$) higher than those of a uniform vector ($\sigma = 1/2$). Thus, a sum of n extreme vectors will have a higher standard deviation too, and will converge more slowly to a Dirac delta function. Table 7.1 shows the maximal deobfuscation

Table 7.1. Maximal deobfuscation probability of sums of vectors

vector kind:	$n = 1$	$n = 2$	$n = 3$	$n = 4$
uniform	10%	29.36%	42.60%	53.18%
extreme	100%	20.54%	26.78%	29.22%
vector kind:	$n = 5$	$n = 6$	$n = 7$	$n = 8$
uniform	62.12%	69.19%	75.02%	79.80%
extreme	37.49%	43.33%	48.56%	53.87%

probability of a sum of n uniform vectors and n extreme vectors[1]. It can be seen that the sum of two or more extreme vectors is more uniform.

The reason for Property 2 is that extreme vectors contain less information by their own. An adversary who knows one of the sub-vectors has no real advantage in guessing the other ones. The fact that one sub-vector is long is not as informative as in the case of uniform vectors. The output of a decomposition in extreme vectors is correlated as well, but to a lesser extent. We will call a random decomposition in extreme sub-vectors *extreme random decomposition*. Algorithm 6 shows an efficient way to implement it. Note that the last sub-vector is not extreme, since it is computed as a difference.

We use the Pearson's correlation coefficient to measure the degree of correlation. Such a coefficient is defined as:

$$\rho_{X,Y} = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y} \tag{7.16}$$

where X and Y are two random variables, $\mathrm{Cov}(X,Y)$ is their covariance, and σ_X and σ_Y are their standard deviations. The closer the coefficient is to zero, the less correlated the two variables. The following matrix contains the Pearson's correlation

[1] Each estimation stems from 100,000 vector sums.

coefficients between the sub-vectors of a random decomposition[2]:

$$\left(\rho_{\mathbf{d}_o^{(i)},\mathbf{d}_o^{(j)}}\right) = \begin{pmatrix} 1.00 & & & & \\ 0.33 & 1.00 & & - & \\ 0.33 & 0.61 & 1.00 & & \\ 0.32 & 0.60 & 0.80 & 1.00 & \\ 0.32 & 0.60 & 0.79 & 0.91 & 1.00 \end{pmatrix}$$

The (i, j)-th element contains the Pearson's correlation coefficient between the i-th and the j-th sub-vectors. It can be seen that the first sub-vector is relatively little correlated to the successive (first column). On the other hand, the last sub-vectors are very correlated to each other (second to fourth columns), because they are constrained to be long and follow the same direction. The following is the analogous matrix for an extreme random decomposition:

$$\left(\rho_{\mathbf{d}_o^{(i)},\mathbf{d}_o^{(j)}}\right) = \begin{pmatrix} 1.00 & & & & \\ 0.21 & 1.00 & & - & \\ 0.23 & 0.43 & 1.00 & & \\ 0.23 & 0.45 & 0.62 & 1.00 & \\ 0.23 & 0.44 & 0.60 & 0.71 & 1.00 \end{pmatrix}$$

It can be seen that the correlation coefficients are always lower. Thus, an extreme decomposition produces less correlated vectors.

Algorithm 6 Extreme random decomposition

procedure X-DECOMPOSE($\mathbf{d}_o^*, r_m, r_o^{(0)}, n$)
  $\mathbf{d}_{sum} = 0$
  for $i = 1 \to (n-1)$ **do**
    $l \leftarrow (n-i)r_o^{(0)}/n - r_m$
    $\mathbf{d}_o^{(i)} \leftarrow (r_o^{(0)}/n)$**-bounded extreme vector,**
    such that: $\mathrm{dist}\left(\mathbf{d}_{sum} + \mathbf{d}_o^{(i)}, \mathbf{d}_o^*\right) \leq l$
    $\mathbf{d}_{sum} \leftarrow \mathbf{d}_{sum} + \mathbf{d}_o^{(i)}$
  end for
  $\mathbf{d}_o^{(n)} \leftarrow \mathbf{d}_o^* - \mathbf{d}_{sum}$
  return $\left\{\mathbf{d}_o^{(1)}, \ldots, \mathbf{d}_o^{(n)}\right\}$
end procedure

In the following, we will show how to employ extreme vectors in a-posteriori and in a-priori share generation methods. We will refer to these unimproved methods as *vanilla* versions. The modified versions, based on extreme vectors, will be the *extreme* versions. Algorithm 7 shows the *extreme a-posteriori share generation method*. Note

[2] Each estimation stems from 100,000 decompositions, with $n = 5$, $r_m = 10\,\mathrm{m}$, and $r_o^{(0)} = 1\,\mathrm{Km}$.

that the first $n-1$ refinement shares are extreme vectors, while the last one is a uniform vector. This is in order to maintain the uniformity of the $(n-1)$-th obfuscation area. Algorithm 8 shows the *extreme a-priori share generation method*.

Algorithm 7 Extreme a-posteriori share generation method

procedure X-APOSTERIORI($A_m, r_o^{(0)}, n$)
  for $i = 1 \to n-1$ **do**
    $\mathbf{d}_o^{(i)} \leftarrow (r_o^{(0)}/n)$**-bounded extreme vector**
  end for
  $\mathbf{d}_o^{(n)} \leftarrow (r_o^{(0)}/n - r_m)$**-bounded uniform vector**
  $\mathbf{X}_o^{(0)} \leftarrow \mathbf{X}_m - \sum_{i=1}^{n} \mathbf{d}_o^{(i)}$
  return $\left\{A_o^{(0)}, \mathbf{d}_o^{(1)}, \ldots, \mathbf{d}_o^{(n)}\right\}$
end procedure

Algorithm 8 Extreme a-priori share generation method

procedure X-APRIORI($A_m, r_o^{(0)}, n$)
  $\mathbf{d}_o^{*} \leftarrow (r_o^{(0)} - r_m)$**-bounded uniform vector**
  $\mathbf{X}_o^{(0)} \leftarrow \mathbf{X}_m - \mathbf{d}_o^{*}$
  $\left\{\mathbf{d}_o^{(1)}, \ldots, \mathbf{d}_o^{(n)}\right\} \leftarrow \mathrm{X\text{-}decompose}(\mathbf{d}_o^{*}, r_m, r_o^{(0)}, n)$
  return $\left\{A_o^{(0)}, \mathbf{d}_o^{(1)}, \ldots, \mathbf{d}_o^{(n)}\right\}$
end procedure

7.3.1 Evaluation of extreme share generation methods

We evaluated the resistance of the extreme share generation methods with $n = 5$ privacy levels. Figures 7.9 and 7.10 show the maximal deobfuscation probability of extreme a-posteriori and a-priori methods compared to their vanilla counterparts, against a malicious provider[3]. The master share is considered to be always compromised. On the abscissas we have the number of compromised refinement shares, i.e. the privacy level that the malicious provider has the right to access. As expected, the extreme versions are always more resistant than the vanilla versions, independently of the number of compromised refinement shares. In the a-posteriori approach, the major improvement is in the lower privacy levels (22.50% for the 0-th privacy level). In the a-priori approach, the resistance of the 0-th privacy level is already perfect, thus cannot be furthermore improved. However, the probabilistic correlation between

[3] Each estimation stems from 100 attack simulations, with $n = 5$, $r_m = 10\,\mathrm{m}$, $r_o^{(0)} = 1\,\mathrm{Km}$. Gaussian-distributed measurement errors are assumed.

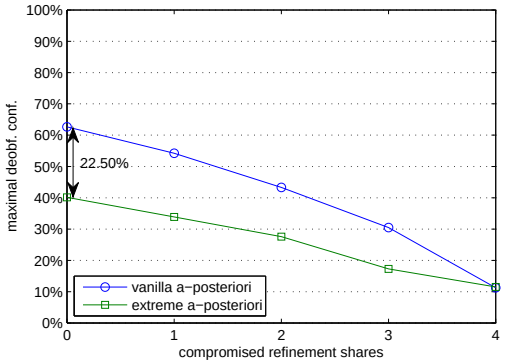


Figure 7.9. Resistance of vanilla and extreme a-posteriori share generation methods against malicious provider

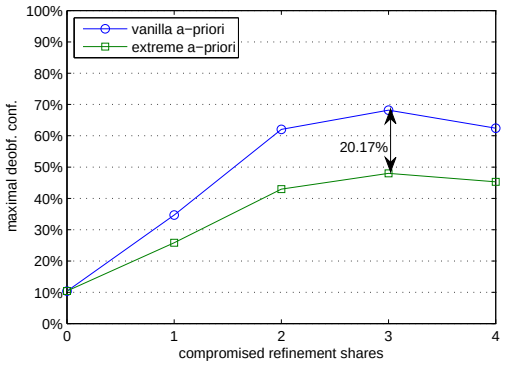


Figure 7.10. Resistance of vanilla and extreme a-priori share generation methods against malicious provider

the obfuscation vectors is significantly mitigated, and this has a positive effect on the resistance in case of higher privacy levels (20.17% for the third privacy level).

Figures 7.11 and 7.12 show the maximal deobfuscation probability of extreme a-posteriori and a-priori methods compared to their vanilla counterparts, against a malicious server[4]. The location server holding master share is considered to be always malicious. On the abscissas we have the number of malicious servers. For instance, "three servers" means that the server holding the master share and other two (randomly chosen) servers are malicious. As expected, the extreme versions are more resistant than the vanilla versions, except when there are $n - 1$ malicious servers. This is because the last non-compromised refinement share will be, with high probability, an extreme vector, which is poorly resilient by itself. However, this happens only in a very pessimistic case, because all the servers except one have to be malicious.

[4] Each estimation stems from 100 attack simulations, with $n = 5$, $r_m = 10\,\mathrm{m}$, $r_o^{(0)} = 1\,\mathrm{Km}$. Gaussian-distributed measurement errors are assumed.

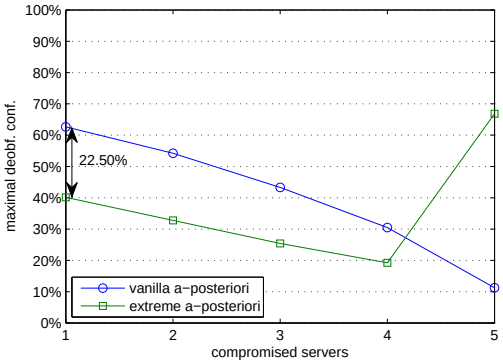


Figure 7.11. Resistance of vanilla and extreme a-posteriori share generation methods against malicious server

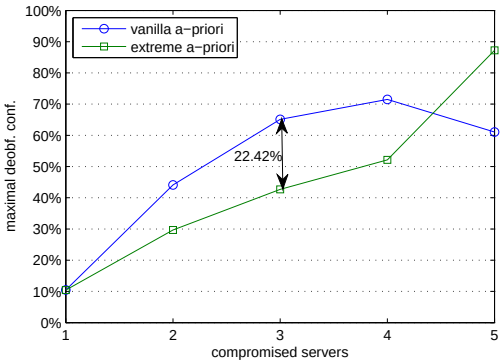


Figure 7.12. Resistance of vanilla and extreme a-priori share generation methods against malicious server

We conclude that, by using extreme vectors instead of classic uniform vectors, both share generation methods significantly improve their resistance, while maintaining their general characteristics (i.e. to be more resistant for less powerful adversaries for a-priori and vice versa for a-posteriori).

7.4 Enlarge-and-scale

We will now present *enlarge-and-scale*, a technique to significantly improve the uniformity of obfuscation algorithms in case of map-aware adversaries. Enlarge-and-scale is applicable to vanilla share generation methods, as well as to their extreme versions. Like enlarge-and-perturb (cfr. Algorithm 5), enlarge-and-scale first enlarges the obfuscation radius (cfr. Algorithm 4). Then, instead of perturbing the center, it performs a *scaling* of the obfuscation vector in accordance to the performed enlargement. For the sake of simplicity, let us consider by now a single level of privacy ($n = 1$). Fig-

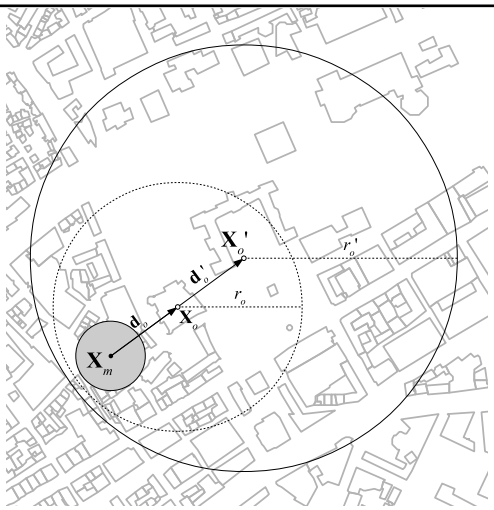


Figure 7.13. Enlarge-and-scale technique

ure 7.13 shows an example of enlarge-and-scale operation. $\mathbf{d}'_o$ is the scaled obfuscation vector, while $\mathbf{X}'_o$ is the new obfuscated position after the scaling operation. Like in enlarge-and-perturb, after having moved the obfuscation area, we have to check again if the size of the map-reduced obfuscation area has become smaller. If it has, we perform an additional enlargement and an additional scaling, and so on.

This enlarge-and-scale approach is preferable to the enlarge-and-perturb approach, because it avoids repeated sums of random vectors that reduce the unpredictability of the obfuscation. By scaling the existing obfuscation vectors, we do not change their probabilistic properties, and therefore we obtain the same uniformity of the map-free case.

The enlarge-and-scale technique is easily extensible in case of $n > 1$ privacy levels, both for a-posteriori and a-priori methods, in the following way. After the user has generated the master share and the refinement shares, he enlarges each obfuscation radius. Then, he chooses the largest relative radius enlargement, and he applies it to all the obfuscation areas. In the scaling phase, the user scales all the obfuscation vectors, according to the performed enlargement. Finally, he checks whether all the n map-reduced obfuscation areas have the nominal size or greater. If they have, the algorithm ends. Otherwise, the user makes an additional enlarge-and-scale step, and so on. Algorithm 9 shows the complete enlarge-and-scale technique. Note that the last obfuscation vector is scaled by a different ratio (Alg. 9, Line 12). This is to compensate the fact that the size of the measurement area is fixed, and cannot be enlarged coherently to the other obfuscation areas.

Algorithm 9 Enlarge-and-scale technique

procedure ENLARGE-AND-SCALE($A_o^{(0)}, \mathbf{d}_o^{(1)}, \ldots, \mathbf{d}_o^{(n)}, M$)
  $S \leftarrow \{1, \ldots, n\}$
  for all i**:** $r_o'^{(i)} \leftarrow r_o^{(i)}$ **and** $\mathbf{X}_o'^{(i)} \leftarrow \mathbf{X}_o^{(i)}$
  repeat
    for all $i \in S$ **do** // Enlargement:
      $r_o'^{(i)} \leftarrow \mathrm{enlarge}(\mathbf{X}_o'^{(i)}, r_o'^{(i)}, M)$
    end for
    $\rho_{max} = \max_i \left\{ \frac{r_o'^{(i)}}{r_o^{(i)}} \right\}$
    for $i = 1 \rightarrow n-1$ **do** // Scaling:
      $\mathbf{d}_o'^{(i)} \leftarrow \rho_{max} \cdot \mathbf{d}_o^{(i)}$
    end for
    $\mathbf{d}_o'^{(n)} \leftarrow \frac{\rho_{max} \cdot \mathbf{d}_o^{(n)} - r_m}{\mathbf{d}_o^{(n)} - r_m}$
    for all i**: compute** $\mathbf{X}_o'^{(i)}$ **and** $A_o'^{(i)}$ **from** $\mathbf{d}_o'^{(i)}$
    $S \leftarrow \left\{ i : \mathrm{size}\left(A_o'^{(i)} \cap M \right) < \pi \cdot \left(r_o^{(i)} \right)^2 \right\}$
  until $S = \emptyset$
  return $A_o'^{(0)}, \mathbf{d}_o'^{(1)}, \ldots, \mathbf{d}_o'^{(n)}$
end procedure

7.4.1 Evaluation of enlarge-and-scale technique

We evaluated the resistance of enlarge-and-scale technique against map-aware adversaries, and we compared it to the performance of enlarge-and-perturb. We tested both algorithms on synthetic Manhattan-like maps, with square-shaped buildings, roads' width equal to $10\,\mathrm{m}$, and a varying distance between parallel roads. The obfuscation areas have been created by means of extreme a-priori share generation method. However, the enlarge-and-scale technique is independent from the share generation method, and the same results apply to the other presented methods as well. First, we want to show that enlarge-and-scale does not produce larger obfuscation areas than enlarge-and-perturb, and thus it does not decrease the quality of service. Figure 7.14 shows the average relative enlargement of the 0-th obfuscation area (with $n = 5$ levels of privacy) versus the ratio of free space of the map[5]. Obviously, the less the free space is, the more the obfuscation areas have to be enlarged, both in enlarge-and-scale and in enlarge-and-perturb methods. However, we can see that both methods enlarge the obfuscation areas quite equally on every map. Thus, enlarge-and-scale does not degrade the quality of service with respect to enlarge-and-perturb.

[5] Each estimation stems from 500 obfuscation simulations, with $n = 5$, $r_m = 10\,\mathrm{m}$, $r_o^{(0)} = 1\,\mathrm{Km}$. Gaussian-distributed measurement errors are assumed.

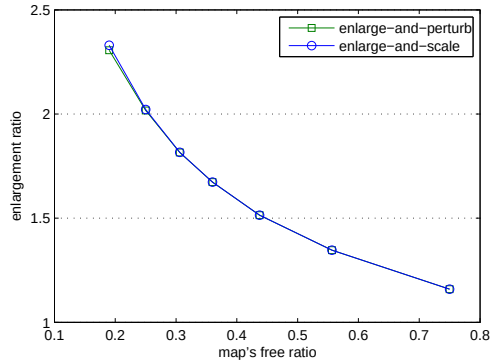


Figure 7.14. Enlargement ratio against walkable space ratio

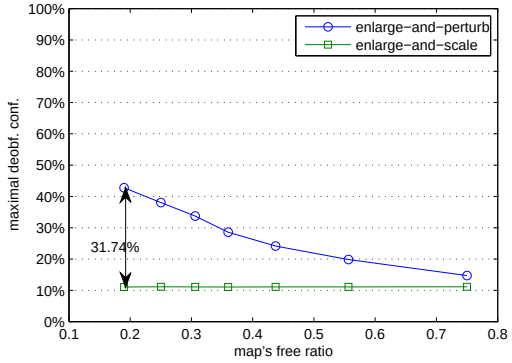


Figure 7.15. Maximal deobfuscation probability in case of map-aware adversary

Regarding the deobfuscation resistance, Figure 7.15 shows the maximal deobfuscation probability of a malicious provider knowing only the master share[6]. On the abscissa we have the *map's free ratio*, i.e. the percentage of walkable space in the map. As expected, the enlarge-and-scale technique is more resistant than the enlarge-and-perturb one. In particular, it has the same uniformity expected from a-priori share generation in free space. This is because the scaling operation does not change the probabilistic properties of the obfuscation vectors. It can be seen that the resistance gain is particularly high for maps with a low walkable ratio (31.74% for walkable ratio equal to 0.19).

To sum up, enlarge-and-scale technique is able to significantly improve the resistance with respect to enlarge-and-perturb, because it avoids repeated sums of random vectors. This resistance gain does not cause a degradation on the quality of the service.

[6] Each estimation stems from 10,000 attack simulations, with $n = 5$, $r_m = 10\,\mathrm{m}$, $r_o^{(0)} = 1\,\mathrm{Km}$. Gaussian-distributed measurement errors are assumed.

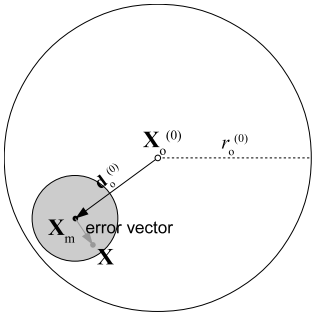


Figure 7.16. Case of non-negligible measurement error

7.5 Hybrid vectors

Every position measurement carries with itself an error, due to technological imprecision. Different technologies have different precisions [65]. If the average position error is very small, for example below 1–5 meters, as it happens in differential GPS or in UWB positioning, we can approximate it to zero. Otherwise, if the measurement noise is comparable to the obfuscation one, as it happens in smartphone's cheap GPS receivers or in cellular positioning, we cannot neglect it. In this case, the obfuscation system has to take into account the measurement imprecision in order to obfuscate in a proper way.

For the sake of simplicity, let us consider by now a single level of privacy ($n = 1$). We call *error vector* the vector $\mathbf{d}_m = \mathbf{X}_m - \mathbf{X}$ (Fig. 7.16), i.e. the vector between the measured and the real user's position. The error vector is a random vector over which the obfuscation system has no control. We assume that the error radius is tailored to be always longer than or equal to the error vector:

$$\|\mathbf{d}_m\| \leq r_m \tag{7.17}$$

In this way, the real position always lies inside the measurement area. If a technology exhibits a theoretically non-bounded error (e.g. a Gaussian one), the obfuscation system can approximate it by truncation at $r_m = 3\sigma$. In this way, only a negligible amount of measurement samples will fall outside the measurement area.

We can think the error vector as an additional "obfuscation vector". The system has no control over this vector, but the adversary has to deobfuscate it anyway if she wants to find the real position, which represents the true personal piece of information. As a result of the presence of an error vector, even if the obfuscation vector is uniform, the distribution of the real position will not be uniform (cfr. Chapter 6). Since the obfuscation and the error vectors constitute a sum of random vectors (cfr. Fig. 7.16), the extreme vectors turn out to be useful to improve the overall resistance. However, using a simple extreme vector is not convenient this time, as it produces an area with zero probability distribution at the center (red area in Fig. 7.17a). On the other hand, adding a classic uniform vector fills the hole at the center but, as shown

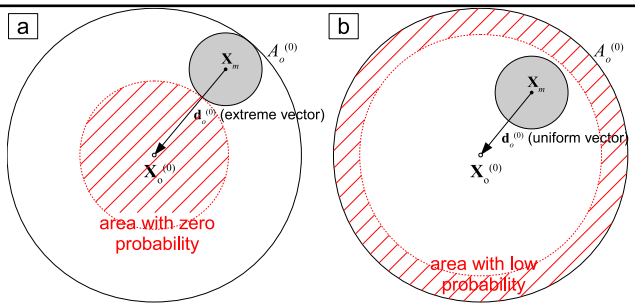


Figure 7.17. Using extreme vectors or uniform vectors for obfuscating imprecise position measurements

in Section 6.4, it produces a lack of probability distribution close to the borders of the obfuscation area (red area in Fig. 7.17b).

Our idea is to use a mix between a uniform vector and an extreme one, that we call *hybrid vector*. An hybrid vector depends on a real parameter $\alpha \in [0, 1]$, that we call *extremeness*.

Definition 14. *An* r-bounded hybrid vector *with extremeness* $\alpha \in [0, 1]$*, is an* r*-bounded uniform vector with probability* α*, or an* r*-bounded extreme vector with probability* $1 - \alpha$.

Note that with an extremeness equal to 0 we obtain a uniform vector, and with an extremeness equal to 1 we obtain an extreme vector. The *optimal extremeness* (α_{opt}) is the one which maximizes the uniformity of the probability distribution, and it is somewhere in the range $[0, 1]$. It depends on the probability distribution of the error vector, and on the *radius ratio* (ρ), defined as:

$$\rho = r_o^{(0)} / r_m \tag{7.18}$$

A radius ratio close to one indicates that the measurement imprecision is of the same magnitude order of the obfuscation noise. On the contrary, a radius ratio tending to infinite indicates that the imprecision is negligible compared to the obfuscation noise.

Computing the optimal extremeness is burdensome for a resource-constrained device, since it requires to simulate a deobfuscation attack for each value of extremeness, and then choosing the one giving the best uniformity. We propose a *heuristic extremeness* (α_{heur}) that approximates the optimal one, given the radius ratio:

$$\alpha_{\text{heur}} = \begin{cases} k_1(2\rho - k_1)/\rho^2 & \rho \in (1, \rho_1] \\ k_2(2\rho - k_2)/\rho^2 & \rho \in (\rho_1, \infty) \end{cases} \tag{7.19}$$

k_1, k_2, and ρ_1 are parameters of the heuristic. Such a heuristic function is based on geometrical considerations (Fig. 7.18). We divide the obfuscation area in two concentric regions: an external one (A), and an internal one (B). The external region has a

width proportional to the error radius, with a proportionality constant k. Then, we make the (approximating) assumption that the real position will be in the external region if and only if the obfuscation vector is extreme (Fig. 7.18, upper vector), and in the internal region if and only if it is uniform (Fig. 7.18, lower vector). This implies that the external region contains the real position with a probability equal to the extremeness of the obfuscation vector:

$$\Pr[\mathbf{X} \in A] = \alpha \tag{7.20}$$

$$\Pr[\mathbf{X} \in B] = 1 - \alpha \tag{7.21}$$

Finally, we fix the extremeness in such a way that both regions contain a probability proportional to their size. By imposing this, we improve the overall uniformity by balancing the probability between the external and the internal regions. We used two values for k (k_1 or k_2) depending on the range in which the radius ratio lies: $(1, \rho_1]$ or (ρ_1, ∞). We noticed that this makes the heuristic closer to the optimal. As the ra-

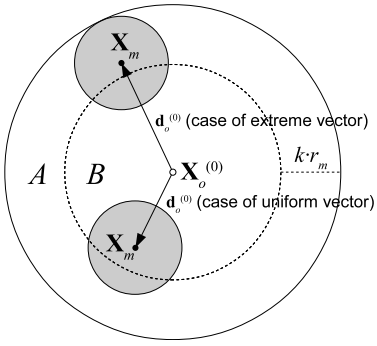


Figure 7.18. Rationale behind the heuristic

dius ratio grows, the heuristic extremeness converges to zero. This is an expected behaviour because, as the measurement imprecision becomes negligible, a uniform obfuscation vector is more suitable. We consider two kinds of measurement error: (a) the *Gaussian error*, typical of GPS; and (b) the *uniform error*, typical of cellular/Wi-Fi positioning. We computed the best parameters of the heuristic for both error models, i.e. the parameters which maximize the uniformity of the probability distribution, averaged on the radius ratio. Table 7.2 shows the best parameters computed for the Gaussian and for the uniform error models. These values minimize the maximal de-obfuscation probability, averaged on the radius ratio.

In order to employ hybrid vectors with $n > 1$ levels of privacy, it is sufficient to use them whenever an obfuscation vector is directly added to the error vector. In case of a-posteriori share generation method, the first obfuscation vector will be generated as hybrid. In case of a-priori share generation method, the master obfuscation vector will be generated as hybrid.

Table 7.2. Best parameters for the heuristic

Error shape:	ρ_1	k_1	k_2
Gaussian	2.4	1.22	0.35
Uniform	3.9	1.89	0.38

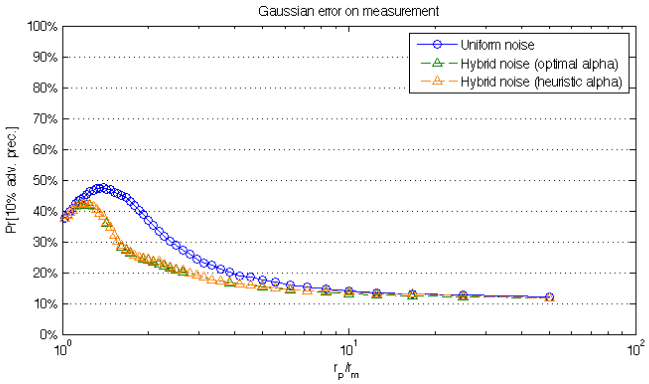


Figure 7.19. Resistance of uniform and hybrid obfuscation with Gaussian error model

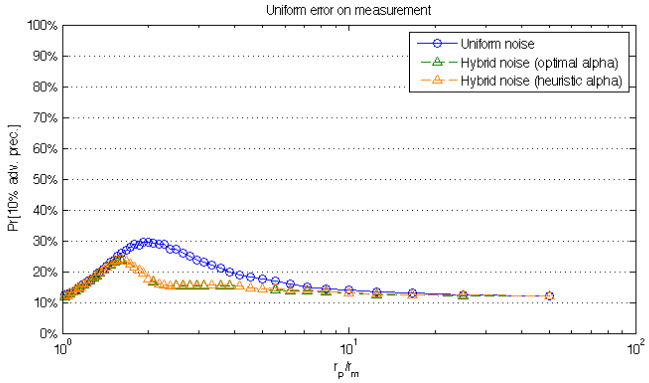


Figure 7.20. Resistance of uniform and hybrid obfuscation with uniform error model

7.5.1 Evaluation of hybrid vectors

Figures 7.19 and 7.20 show the maximal deobfuscation probability under respectively Gaussian and uniform error models, obfuscated by uniform vectors, by optimal hybrid vectors, and by heuristic hybrid vectors. It can be noted that hybrid vectors always overwhelm uniform ones in terms of obfuscation uniformity. They can reduce the maximal deobfuscation probability of 17.17% under Gaussian error model, and of 13.56% under uniform error model. Also, the performance of the heuristic closely follows the optimum. In the worst case, our heuristic increases the maximal deobfuscation probability of only 1.34% under Gaussian error model, and of 0.66% under uniform error model.

To sum up, hybrid vectors are capable of significantly improving the resistance against statistical analysis in case of non-negligible measurement error. The heuristic we presented permits to compute the value of the extremeness in an efficient way, without losing resistance with respect to the optimum.

Conclusions

In this Ph.D. dissertation we approached some security and privacy problems on position data. Namely, we focused on the topics of secure positioning and location privacy.

Secure positioning aims at measuring the position of a device in presence of an adversary trying to deceive the measurement process. We focused on range-based secure positioning techniques, which are based on distance-bounding protocols. Distance bounding allows us to determine a secure upper bound to the distance between two devices. We first approached the sub-problem of *enlargement attacks*. Enlargement attacks aim at deceiving the distance bounding into measuring a distance larger than the real one. They can follow a jam-and-replay strategy or an overshadow strategy. We proposed *SecDEv*, a distance bounding protocol able to withstand jam-and-replay strategies. Then, we studied the feasibility of overshadow strategies against distance bounding implemented on IEEE 802.15.4a UWB. Basing on the results of this analysis, we proposed *EMCD-ML*, a method for secure positioning based on the impossibility of the adversary to control the effect of an overshadow attack. EMCD-ML sensibly reduces the necessary number of anchor nodes with respect to state-of-the-art methods.

Location privacy aims at avoiding the disclosure of (precise) position data in location-based services (*LBSs*). Privacy-preserving mechanisms can be various and orthogonal to each other. We proposed *LbSprint*, a software architecture to integrate different privacy-preserving mechanisms by means of the standard language XACML. Then, we developed *UniLO*, a mathematical operator for location privacy. UniLO reduces the precision of a position before its disclosure, in such a way that an adversary cannot reconstruct original data. We also extended it to provide for multiple contemporaneous levels of privacy. We showed that UniLO surpasses state-of-the-art obfuscation methods in terms of resistance against statistical attacks, while still permitting the delivery of the service. Finally, we developed some advanced techniques that further improve the resistance of UniLO in case of untrusted location servers, map-aware adversaries and imprecise position measurements.

References

1. A. Abu-Mahfouz and Gerhard P. Hancke. Distance bounding: A practical security solution for real-time location systems. *Industrial Informatics, IEEE Transactions on*, 9(1):16–27, Feb 2013.
2. Osman Abul, Francesco Bonchi, and Mirco Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 376–385. Ieee, 2008.
3. Linda Ackerman, James Kempf, and Toshio Miki. Wireless location privacy: A report on law and policy in the United States, the European Union, and Japan. *DoCoMo USA Labs Technical Report DCL-TR*, 1:2003, 2003.
4. Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. *ACM Sigmod Record*, 29(2):439–450, 2000.
5. Foursquare alters API to eliminate apps like Girls Around Me. http://aboutfoursquare.com/foursquare-api-change-girls-around-me/.
6. Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 901–914. ACM, 2013.
7. Claudio A Ardagna, Marco Cremonini, Ernesto Damiani, Sabrina De Capitani di Vimercati, and Pierangela Samarati. Supporting location-based conditions in access control policies. In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 212–222. ACM, 2006.
8. Claudio Agostino Ardagna, Marco Cremonini, Sabrina De Capitani di Vimercati, and Pierangela Samarati. An obfuscation-based approach for protecting location privacy. *Dependable and Secure Computing, IEEE Transactions on*, 8(1):13–27, 2011.
9. Claudio Agostino Ardagna, Marco Cremonini, and Gabriele Gianini. Landscape-aware location-privacy protection in location-based services. *Journal of Systems Architecture*, 55(4):243–254, 2009.
10. Louise Barkhuus, Barry Brown, Marek Bell, Scott Sherwood, Malcolm Hall, and Matthew Chalmers. From awareness to repartee: sharing location within social groups. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 497–506. ACM, 2008.
11. Louise Barkhuus and Anind K Dey. Location-based services for mobile telephony: a study of users' privacy concerns. In *INTERACT*, volume 3, pages 702–712. Citeseer, 2003.
12. Michael Benisch, Patrick Gage Kelley, Norman Sadeh, Tuomas Sandholm, Janice Tsai, Lorrie Faith Cranor, and Paul Hankes Drielsma. The impact of expressiveness on the effectiveness of privacy mechanisms for location-sharing. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, page 22. ACM, 2009.

13. Alastair R Beresford and Frank Stajano. Location privacy in pervasive computing. *Pervasive Computing, IEEE*, 2(1):46–55, 2003.
14. Claudio Bettini, Sergio Mascetti, X Sean Wang, Dario Freni, and Sushil Jajodia. Anonymity and historical-anonymity in location-based services. In *Privacy in Location-Based Applications*, pages 1–30. Springer, 2009.
15. Stefan Brands and David Chaum. Distance-bounding protocols. In *Advances in Cryptology–EUROCRYPT'93*, pages 344–359. Springer, 1994.
16. Laurent Bussard and Walid Bagga. Distance-bounding proof of knowledge to avoid real-time attacks. In *Security and Privacy in the Age of Ubiquitous Computing*, pages 223–238. Springer, 2005.
17. Nishanth Chandran, Vipul Goyal, Ryan Moriarty, and Rafail Ostrovsky. Position based cryptography. In *Advances in Cryptology-CRYPTO 2009*, pages 391–407. Springer, 2009.
18. Reynold Cheng, Yu Zhang, Elisa Bertino, and Sunil Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *Privacy Enhancing Technologies*, pages 393–412. Springer, 2006.
19. Jerry T Chiang, Jason J Haas, Jihyuk Choi, and Yih-Chun Hu. Secure location verification using simultaneous multilateration. *Wireless Communications, IEEE Transactions on*, 11(2):584–591, 2012.
20. Jerry T Chiang, Jason J Haas, and Yih-Chun Hu. Secure and precise location verification using distance bounding and simultaneous multilateration. In *Proceedings of the second ACM conference on Wireless network security*, pages 181–192. ACM, 2009.
21. Chi-Yin Chow, Mohamed F Mokbel, and Xuan Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, pages 171–178. ACM, 2006.
22. Jolyon Clulow, Gerhard P Hancke, Markus G Kuhn, and Tyler Moore. So near and yet so far: Distance-bounding attacks in wireless networks. In *Security and Privacy in Ad-Hoc and Sensor Networks*, pages 83–97. Springer, 2006.
23. Sunny Consolvo, Ian E Smith, Tara Matthews, Anthony LaMarca, Jason Tabert, and Pauline Powledge. Location disclosure to social relations: why, when, & what people want to share. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 81–90. ACM, 2005.
24. Alissa Cooper and John Morris. An Architecture for Location and Location Privacy in Internet Applications. RFC 6280 (Informational), July 2011.
25. Cas Cremers, Kasper Bonne Rasmussen, Benedikt Schmidt, and Srdjan Capkun. Distance hijacking attacks on distance bounding protocols. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 113–127. IEEE, 2012.
26. Elaine A Crider and Valerie D Francies. Method and apparatus for electronically tracking luggage, May 19 2009. US Patent 7,535,358.
27. Maria Luisa Damiani, Elisa Bertino, and Claudio Silvestri. Protecting location privacy against spatial inferences: the PROBE approach. In *Proceedings of the 2nd SIGSPATIAL ACM GIS 2009 International Workshop on Security and Privacy in GIS and LBS*, pages 32–41. ACM, 2009.
28. A.A. D'Amico, U. Mengali, and L. Taponecco. Energy-based TOA estimation. *Wireless Communications, IEEE Transactions on*, 7(3):838–847, March 2008.
29. Antonio A D'Amico, Umberto Mengali, and Lorenzo Taponecco. Toa estimation with the IEEE 802.15.4a standard. *Wireless Communications, IEEE Transactions on*, 9(7):2238–2247, 2010.
30. Yvo Desmedt. Major security problems with the 'unforgeable' (Feige)-Fiat-Shamir proofs of identity and how to overcome them. In *SecuriCom*, volume 88, pages 15–17, 1988.
31. Gianluca Dini, Francesco Giurlanda, and Lucia Pallotttino. Neighbourhood monitoring for decentralised coordination in multi-agent systems: a case-study. In *Computers and Communications (ISCC), 2011 IEEE Symposium on*, pages 681–683. IEEE, 2011.

32. Gianluca Dini, Francesco Giurlanda, and Pericle Perazzo. SecDEv: Secure distance evaluation in wireless networks. In *ICNS 2013, The Ninth International Conference on Networking and Services*, pages 207–212, 2013.
33. Gianluca Dini and Pericle Perazzo. Uniform obfuscation for location privacy. In *Data and Applications Security and Privacy XXVI*, pages 90–105. Springer Berlin Heidelberg, 2012.
34. Gianluca Dini and Pericle Perazzo. Integration of privacy protection mechanisms in location-based services. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pages 717–724. IEEE, 2013.
35. Thomas D'Roza and George Bilchev. An overview of location-based services. *BT Technology Journal*, 21(1):20–27, 2003.
36. Matt Duckham and Lars Kulik. A formal model of obfuscation and negotiation for location privacy. In *Pervasive computing*, pages 152–170. Springer, 2005.
37. Frank Dürr, Pavel Skvortsov, and Kurt Rothermel. Position sharing for location privacy in non-trusted systems. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, pages 189–196. IEEE, 2011.
38. Cynthia Dwork. Differential privacy. In *Automata, languages and programming*, pages 1–12. Springer, 2006.
39. Fredrik Espinoza, Per Persson, Anna Sandin, Hanna Nyström, Elenor Cacciatore, and Markus Bylund. GeoNotes: Social and navigational aspects of location-based information systems. In *Ubicomp 2001: Ubiquitous Computing*, pages 2–17. Springer, 2001.
40. FireEagle. http://fireeagle.yahoo.net.
41. Manuel Flury, Marcin Poturalski, Panos Papadimitratos, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. Effectiveness of distance-decreasing attacks against impulse radio ranging. In *Proceedings of the third ACM conference on Wireless network security*, pages 117–128. ACM, 2010.
42. Foursquare. http://foursquare.com.
43. Julien Freudiger, Reza Shokri, and Jean-Pierre Hubaux. On the optimal placement of mix zones. In *Privacy enhancing technologies*, pages 216–234. Springer, 2009.
44. Bugra Gedik and Ling Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *Mobile Computing, IEEE Transactions on*, 7(1):1–18, 2008.
45. Sinan Gezici. A survey on wireless position estimation. *Wireless Personal Communications*, 44(3):263–282, 2008.
46. Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan. Private queries in location based services: anonymizers are not necessary. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 121–132. ACM, 2008.
47. Gabriel Ghinita, Keliang Zhao, Dimitris Papadias, and Panos Kalnis. A reciprocal framework for spatial k-anonymity. *Information Systems*, 35(3):299–314, 2010.
48. Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42. ACM, 2003.
49. Ismail Guvenc, Zafer Sahinoglu, Andreas F Molisch, and Philip Orlik. Non-coherent TOA estimation in IR-UWB systems with different signal waveforms. In *Proc. IEEE Int. Workshop on Ultrawideband Networks (UWBNETS), Boston, MA*, pages 245–251, 2005.
50. Gerhard P Hancke and Markus G Kuhn. An RFID distance bounding protocol. In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pages 67–73. IEEE, 2005.
51. Bernhard Hofmann-Wellenhof, Herbert Lichtenegger, and James Collins. *Global Positioning System: Theory and Practice*. Springer, 2001.
52. Haibo Hu, Jianliang Xu, Qian Chen, and Ziwei Yang. Authenticating location-based services without compromising location privacy. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 301–312. ACM, 2012.

53. Wen Hu, Hailun Tan, Peter Corke, Wen Chan Shih, and Sanjay Jha. Toward trusted wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 7(1):5, 2010.
54. Yih-Chun Hu, Adrian Perrig, and David B Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1976–1986. IEEE, 2003.
55. IEEE Computer Society. IEEE Std 802.15.4a-2007 (Amendment 1: Add Alternate PHYs), 2007.
56. Panos Kalnis, Gabriel Ghinita, Kyriakos Mouratidis, and Dimitris Papadias. Preventing location-based identity inference in anonymous spatial queries. *Knowledge and Data Engineering, IEEE Transactions on*, 19(12):1719–1733, 2007.
57. Ali Khoshgozaran, Cyrus Shahabi, and Houtan Shirani-Mehr. Location privacy: going beyond k-anonymity, cloaking and anonymizers. *Knowledge and Information Systems*, 26(3):435–465, 2011.
58. Jiejun Kong, Zhengrong Ji, Weichao Wang, Mario Gerla, Rajive Bagrodia, and Bharat Bhargava. Low-cost attacks against packet delivery, localization and time synchronization services in under-water sensor networks. In *Proceedings of the 4th ACM workshop on Wireless security*, pages 87–96. ACM, 2005.
59. John Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, 2009.
60. Marc Langheinrich. A privacy awareness system for ubiquitous computing environments. In *UbiComp 2002: Ubiquitous Computing*, pages 237–245. Springer, 2002.
61. Latitude. http://www.google.com/latitude.
62. Loukas Lazos and Radha Poovendran. SeRLoc: Robust localization for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 1(1):73–100, 2005.
63. Ming Li, Sergio Salinas, Arun Thapa, and Pan Li. n-CD: A geometric approach to preserving location privacy in location-based services. In *INFOCOM, 2013 Proceedings IEEE*, pages 3012–3020. IEEE, 2013.
64. Alex X Liu, Fei Chen, JeeHyun Hwang, and Tao Xie. Xengine: a fast and scalable XACML policy evaluation engine. In *ACM SIGMETRICS Performance Evaluation Review*, volume 36, pages 265–276. ACM, 2008.
65. Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1067–1080, 2007.
66. Loopt. http://www.loopt.com.
67. Luciana Marconi, Roberto Di Pietro, Bruno Crispo, and Mauro Conti. Time warp: how time affects privacy in LBSs. In *Information and Communications Security*, pages 325–339. Springer, 2010.
68. Sergio Mascetti, Claudio Bettini, Dario Freni, Xiaoyang Sean Wang, and Sushil Jajodia. Privacy-aware proximity based services. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pages 31–40. IEEE, 2009.
69. Mohamed F Mokbel, Chi-Yin Chow, and Walid G Aref. The new Casper: query processing for location services without compromising privacy. In *Proceedings of the 32nd international conference on Very large data bases*, pages 763–774. VLDB Endowment, 2006.
70. Andreas F Molisch, Dajana Cassioli, Chia-Chin Chong, Shahriar Emami, Andrew Fort, Balakrishnan Kannan, Johan Karedal, Juergen Kunisch, Hans Gregory Schantz, Kazimierz Siwiak, et al. A comprehensive standardized model for ultrawideband propagation channels. *Antennas and Propagation, IEEE Transactions on*, 54(11):3151–3166, 2006.
71. PY Montgomery, Todd E Humphreys, and Brent M Ledvina. A multi-antenna defense: Receiver-autonomous GPS spoofing detection. *Inside GNSS*, 4(2):40–46, 2009.

72. Ginger Myles, Adrian Friday, and Nigel Davies. Preserving privacy in environments with location-based applications. *Pervasive Computing, IEEE*, 2(1):56–64, 2003.
73. OASIS. OASIS eXtensible Access Control Markup Language (XACML) (http://www.oasis-open.org/committees/xacml/)), 2007.
74. Open Geospatial Consortium. Geospatial eXtensible Access Control Markup Language (GeoXACML) (www.opengeospatial.org/standards/geoxacml), 2008.
75. OpenStreetMap. www.openstreetmap.org.
76. Amitangshu Pal. Localization algorithms in wireless sensor networks: Current approaches and future challenges. *Network Protocols & Algorithms*, 2(1), 2010.
77. Balaji Palanisamy and Ling Liu. MobiMix: Protecting location privacy with mix-zones over road networks. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, pages 494–505. IEEE, 2011.
78. Stavros Papadopoulos, Spiridon Bakiras, and Dimitris Papadias. Nearest neighbor search with strong location privacy. *Proceedings of the VLDB Endowment*, 3(1-2):619–629, 2010.
79. Russell Paulet, Md Golam Koasar, Xun Yi, and Elisa Bertino. Privacy-preserving and content-protecting location based queries. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 44–53. IEEE, 2012.
80. Marcin Poturalski, Manuel Flury, Panos Papadimitratos, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. The cicada attack: degradation and denial of service in IR ranging. In *Ultra-Wideband (ICUWB), 2010 IEEE International Conference on*, volume 2, pages 1–4. IEEE, 2010.
81. Marcin Poturalski, Manuel Flury, Panos Papadimitratos, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. Distance bounding with IEEE 802.15.4a: Attacks and countermeasures. *Wireless Communications, IEEE Transactions on*, 10(4):1334–1344, 2011.
82. Marcin Poturalski, Manuel Flury, Panos Papadimitratos, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. On secure and precise IR-UWB ranging. *Wireless Communications, IEEE Transactions on*, 11(3):1087–1099, 2012.
83. Kasper Bonne Rasmussen and Srdjan Čapkun. Location privacy of distance bounding protocols. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 149–160. ACM, 2008.
84. Christof Rohrig, Daniel Heß, Christopher Kirsch, and F Kunemund. Localization of an omnidirectional transport robot using IEEE 802.15.4a ranging and laser range finder. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3798–3803. IEEE, 2010.
85. Zafer Sahinoglu and Sinan Gezici. Ranging in the IEEE 802.15.4a standard. In *Wireless and Microwave Technology Conference, 2006. WAMICON'06. IEEE Annual*, pages 1–5. IEEE, 2006.
86. Pierangela Samarati. Protecting respondents identities in microdata release. *Knowledge and Data Engineering, IEEE Transactions on*, 13(6):1010–1027, 2001.
87. Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, Technical report, SRI International, 1998.
88. Naveen Sastry, Umesh Shankar, and David Wagner. Secure verification of location claims. In *Proceedings of the 2nd ACM workshop on Wireless security*, pages 1–10. ACM, 2003.
89. Bruce Schneier. Secrecy, security, and obscurity (www.schneier.com/crypto-gram-0205.html), May 2002.
90. Matlab simulation scripts for UniLO. www.iet.unipi.it/g.dini/download/code/unilo-simulations.zip.
91. Pavel Skvortsov, Frank Dürr, and Kurt Rothermel. Map-aware position sharing for location privacy in non-trusted systems. In *Pervasive Computing*, pages 388–405. Springer, 2012.
92. Sun. Sun XACML implementation (http://sunxacml.sourceforge.net/).

93. Kar Way Tan, Yimin Lin, and Kyriakos Mouratidis. Spatial cloaking revisited: Distinguishing information leakage from anonymity. In *Advances in Spatial and Temporal Databases*, pages 117–134. Springer, 2009.
94. Lorenzo Taponecco, Pericle Perazzo, A D'Amico, and Gianluca Dini. On the feasibility of overshadow enlargement attack on IEEE 802.15.4a distance bounding. *IEEE Communications Letters (to appear)*, 2013.
95. Don Tennant. GPS dating app yields position-shifting technology for social media. http://www.itbusinessedge.com/blogs/from-under-the-rug/gps-dating-app-yields-position-shifting-technology-for-social-media.html, jan 2014.
96. Nils Ole Tippenhauer and Srdjan Čapkun. ID-based secure distance bounding and localization. In *Computer Security–ESORICS 2009*, pages 621–636. Springer, 2009.
97. Eran Toch, Justin Cranshaw, Paul Hankes-Drielsma, Jay Springfield, Patrick Gage Kelley, Lorrie Cranor, Jason Hong, and Norman Sadeh. Locaccino: a privacy-centric location sharing application. In *Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing-Adjunct*, pages 381–382. ACM, 2010.
98. Janice Y Tsai, Patrick Gage Kelley, Lorrie Faith Cranor, and Norman Sadeh. Location-sharing technologies: Privacy risks and controls. In *In Research Conference on Communication, Information and Internet Policy (TPRC)*. Citeseer, 2009.
99. Fatih Turkmen and Bruno Crispo. Performance evaluation of XACML PDP implementations. In *Proceedings of the 2008 ACM workshop on Secure web services*, pages 37–44. ACM, 2008.
100. Srdjan Čapkun and Jean-Pierre Hubaux. Secure positioning in wireless networks. *Selected Areas in Communications, IEEE Journal on*, 24(2):221–232, 2006.
101. Adnan Vora and Mikhail Nesterenko. Secure location verification using radio broadcast. *Dependable and Secure Computing, IEEE Transactions on*, 3(4):377–385, 2006.
102. Ting Wang and Ling Liu. Privacy-aware mobile services over road networks. *Proceedings of the VLDB Endowment*, 2(1):1042–1053, 2009.
103. Kyle Wesson, Mark Rothlisberger, and Todd Humphreys. Practical cryptographic civil GPS signal authentication. *Navigation*, 59(3):177–193, 2012.
104. World Wide Web Consortium. Simple object access protocol 1.1 (SOAP1.1) specification (http://www.w3.org/tr/2000/note-soap-20000508/), May 2000.
105. Toby Xu and Ying Cai. Exploring historical location data for anonymity preservation in location-based services. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008.
106. Man Lung Yiu, Christian S Jensen, Xuegang Huang, and Hua Lu. SpaceTwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 366–375. IEEE, 2008.
107. Paul A Zandbergen. Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning. *Transactions in GIS*, 13(s1):5–25, 2009.
108. Yanchao Zhang, Wei Liu, Yuguang Fang, and Dapeng Wu. Secure localization and authentication in ultra-wideband sensor networks. *Selected Areas in Communications, IEEE Journal on*, 24(4):829–835, 2006.

Acknowledgment

I would like to thank:

My advisor, Prof. Gianluca Dini
For having directed my creativity in concrete projects.
For having analyzed, discussed, and sometimes attacked my ideas to improve them.

My wife, Beatrice Sanna
Thank you Beatrice, for your deep love.

My family
Especially my mother Paola Calvetti, my father Paolo Perazzo, and my grandparents Noemi Bigazzi, Enzo Calvetti, Franca Bonfanti, Piero Perazzo. They gave me a lot.

Davide Cavaciocchi, for his precious tutorship on music, books, films, and philosophy. And for the Beatles.

All my friends of the "Signa's philosophical meetings:"
Luigi Arceri, Davide Cavaciocchi, Giulia Cefis, Marco Damasceni, Gianpiero Fossi, Catello Marciano, Clarissa Panicagli, Gabriele Scalini, Federico Stagi.

My colleagues:
Stefano Abbate, Hakjeon Bang, Alessio Bechini, Simone Brienza, Stefano Campanelli, Luca Cassano, Daniel Cesarini, Mario Cimino, Roberta Daidone, Antonio Alberto D'Amico, Eleonora D'Andrea, Nicoletta De Francesco, Domenico De Guglielmo, Davide Di Baccio, Adriano Faggiani, Adriano Fagiolini, Ilaria Giannetti, Francesco Giurlanda, Francesco Filia, Alessandro Improta, Koteswararao Kondepu, Mariantonietta Noemi La Polla, Enrico Lauria, Giuseppe Lettieri, Leonardo Loparco, Alessandro Lori, Valerio Luconi, Francesco Magno, Simone Martini, Giovanni Nardini, Chiara Orsini, Alessandro Pischedda, Andrea Saracino, Giovanni Stea, Giacomo Tanganelli,

Lorenzo Taponecco, Marco Tiloca, Gigliola Vaglini, Carlo Vallati, Antonio Virdis.

Roberto "Freak" Antoni (1954–2014), his poetry made me smile and think.

All the great and underrated Italian rock'n'roll artists, among which:
Nada Malanima, Skiantos, Verdena.

The following music artists and bands, that colored these three years:
Arcade Fire, Arctic Monkeys, Atoms For Peace, Beatles, Black Sabbath, Bob Dylan, Caparezza, Captain Beefheart, CCCP, Chuck Berry, Clash, Cure, David Bowie, Dead Can Dance, Deerhunter, Deep Purple, Dirty Three, Doors, Eels, Elio e le storie tese, Elvis Presley, Ennio Morricone, Fabrizio De André, Flaming Lips, Fleet Foxes, Frank Zappa, Have a Nice Life, Holy Model Rounders, Iggy Pop and the Stooges, Interpol, Iron Maiden, Janis Joplin, Jefferson Airplane, Jimi Hendrix, John Lennon, Joy Division, Killing Joke, King Crimson, Led Zeppelin, Lou Reed, Low, Massive Attack, Megadeth, Metallica, My Bloody Valentine, Nada Malanima, Neil Young, Neutral Milk Hotel, Nick Cave and the Bad Seeds, Nine Inch Nails, Nirvana, Pantera, Patti Smith, Paolo Conte, Paul McCartney, Pere Ubu, Pink Floyd, Pixies, Portishead, Prefab Sprout, Primal Scream, Primus, Radiohead, Rainbow, Ramones, Red House Painters, Red Temple Spirits, Rino Gaetano, Rolling Stones, Sex Pistols, Sigur Rós, Siouxsie and the Banshees, Sixto Rodriguez, Skiantos, Slayer, Slint, Smashing Pumpkins, Stato Sociale, Stranglers, Talking Heads, Talk Talk, Tim Buckley, Tool, Tripwires, Velvet Underground, Venom, Verdena, Who.