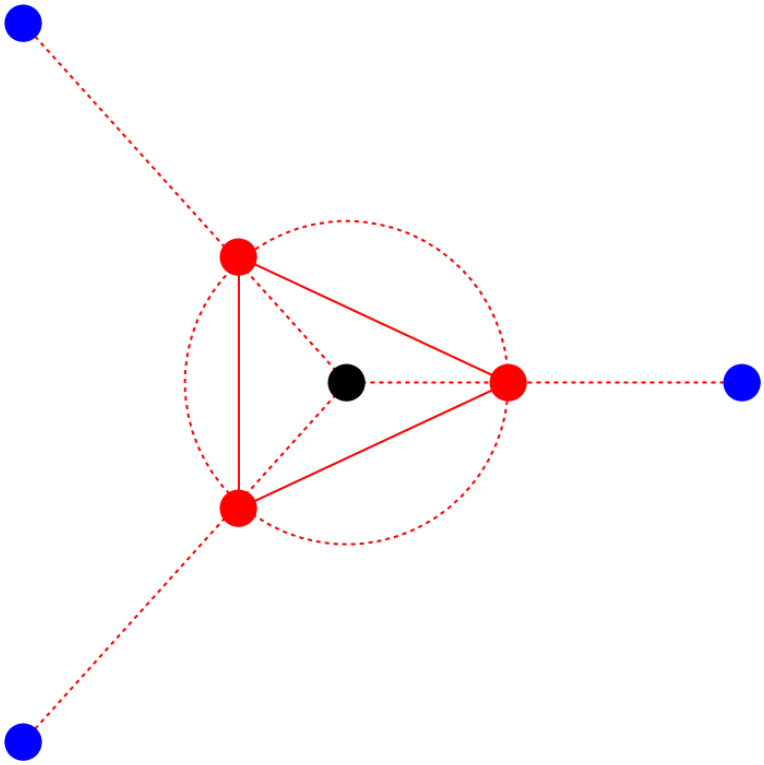
Position-Based Cryptography

Joost Helbing, Roelof Kuipers
10244956, 10220321



Tweedejaarsproject Wiskunde
Supervisor: Christian Schaffner.

July 16, 2014

Abstract

Position-verification is the process of verifying the presence of a device at a certain position. This verification is secure when no adversaries can falsely claim to be present there. In this article several models for position-verification are reviewed. First, the Vanilla model is reviewed. This model is the most basic model for position-verification, for it does not impose limitations on the abilities or knowledge of the adversaries. We prove that no secure position-verification protocol can exist in the Vanilla model. Next, the Hidden-Base Model, in which the location of one or more of the verifiers is unknown, and the Moving-Base Model, in which a verifier is not stationary but moves uniformly at random, are reviewed. We prove that in both models, no secure position-verification protocol can exist, unless assumptions are made that would make these models inapplicable.

Contents

Chapter 1

Introduction

You would never tell anybody you don't know information about your bank accounts. The Dutch government has propagated this through campaigns like *Nepmail, daar trapt u niet in* or *Hang op, klik weg, bel uw bank!* and nowadays it is part of our common sense. Nonetheless, when you walk into a bank and discuss information about your bank account with the clerk, this is exactly what you are doing. This person is probably a stranger to you. Then why do you trust this person with your valuable and confidential information? Clearly, you trust the clerk with information about your bank accounts, simply because he or she is at that precise position: behind the counter of your bank. Apparently the position of someone we are communicating with can be a deciding factor in whether or not we should trust this person.

This special form of trust has got a digital equivalent. Cryptography in which the position of a device is (implemented as a part of) the key, is called *Position-Based Cryptography* (PBC). The research field of PBC is about the question if and how the position of a device can be used to construct secure cryptosystems. If this is possible, it can be used in the secure exchange of certain data.

An obvious application would be communication between a Ministry of Defense and a military base near hostile territory. If the information sent from the Ministry of Defense could only be decrypted on a specific area well inside the military base, this would reduce the chances of the enemy being able to successfully acquire this delicate information.

This gives rise to a question: what if some other device could make it look like it was at the position required for it to be trusted with some information. It could then be granted access to information, whilst not being at the required position, thus whilst not being trustworthy. In this article an answer is given to the question: *Is it possible to securely determine the position of a device?*

We answer this question by analysing and comparing the articles "Position Based Cryptography" by Chandran et al. (2009) and "Secure Location Verification with Hidden and Mobile Base Stations" by Capkun et al. (2008). By comparing the different arguments about secure position-verification suggested in these articles we give a clear and consistent review of the possibilities and challenges for Position-Based Cryptography. We also provide alternative calculations and proofs and clearify the argumentation of these articles.

In Chapter 2, the standard model for PBC is defined on which proofs about the security of PBC are given. In the following chapter this model, the Vanilla model, is elaborated on, as well as the effect of changing certain assumptions of the model on the security of PBC. In the end the assumptions necessary for PBC to be secure are evaluated in the light of applications.

Chapter 2

The Vanilla Model

Just like vanilla is the standard flavour for icecream, the Vanilla model is the standard model for Position-Based Cryptography. The Vanilla model is a model with three types of parties: the *verifiers* try to identify a *prover* at position P, and the *adversaries* try to falsely make it look as if someone is located at P (Chandran et al., 2009). The model is explained in the Section 2.1. In Section 2.2, a proof is given for the claim that there can not be a secure position-verification protocol in the Vanilla Model.

The Vanilla Model is a model in which the positions of all verifiers are fixed and known to all players. Therefore impossibility of secure position-verification in the Vanilla Model does not mean no applicable secure position-verification protocol could ever exist. With other assumptions could come other results to the (im)possibility claim. In Chapter 3 some models with different assumptions are discussed.

2.1 The Model

In our review, 3-dimensional space is considered. Every player knows the security parameter κ as well as the positions of the individual verifiers and the position under investigation, P. This P must lie in the convex hull enclosed by the verifiers (Chandran et al., 2009). All players can send either *broadcast messages* or *directional messages*. Broadcast messages are messages that, when sent from a position Q, move away from Q in concentric hyperspheres. Directional messages are messages that, when sent form a position Q, travel in a region of concentric hyperspheres centred at Q. In addition to these two types of messages, both the verifiers and the adversaries have the ability to communicate internally over a covert channel. When the verifiers communicate over their covert channel, neither adversaries nor prover can obtain any information about their message. The same goes for communication over the covert channel by the adversaries. In this article we assume all messages travel at the speed of light. As a consequence, we can say that distance and time are the same. Any protocol for position-verification by the verifiers is assumed known to all players.

Secure Position-Verification

Let $\mathcal{V} = \{V_1, \cdots, V_n\}$ be the set of verifiers and let $\mathcal{A} = \{A_1, \cdots A_k\}$ be the set of adversaries. A position-verification protocol is a protocol in which the set of verifiers interact with a prover at position P' and who jointly return "Accept" when $P' = P$. The interaction consists of messages M_i

sent by verifiers $V_i \in \mathcal{V}$, on which the prover performs some computation in order to get a response message it sends back to the verifiers. Recall that any protocol executed by the verifiers is known to all players.

A position-verification protocol is said to be *secure* when the verifiers return "Accept" after having interacted with the set of adversaries, non of which actually being present at P, with probability ε. Here ε is negligible in security parameter κ.

The situation in which both adversaries, none of them at P, and a prover at P are present, is not considered in this article. This is because when a prover is at P, any adversaries also present would simply do nothing, for the verifiers would already return "Accept" after having interacted with the prover.

Assumptions
In the Vanilla Model, all players can read and perform computations on received messaged instantaneously. This is another way of saying that the adversaries always have devices with equal or even superior computational power in comparison with the prover.

2.2 Impossibility of Secure Position-Verification in the Vanilla Model

In this section the impossibility of secure position-verification in the Vanilla Model is proven. We show that the adversaries $\mathcal{A}$ together can simulate the secure positioning protocol of the verifiers $\mathcal{V}$ in such a way that the verifiers $\mathcal{V}$ cannot distinguish between executing the protocol with prover P or with the adversaries $\mathcal{A}$.

Theorem 1. *There does not exist a protocol to achieve secure position-verification in the Vanilla model.*

Chandran et al. (2009) have given a proof where adversaries $\mathcal{A}$ together can simulate the position-verification protocol carried out by the verifiers $\mathcal{V}$ in such a way, that the $\mathcal{V}$ cannot see a difference in an execution of their protocol with prover at position P or with the adversaries $\mathcal{A}$. In this way the adversaries $\mathcal{A}$ can always pretend that someone is present at position P, even if no one is. The verifiers $\mathcal{V}$ then are not able to securely check if the prover is present at position P.[1]

Proof. The main idea behind this proof is that the distance between each adversary A_i and $A_{i'}$ is less or equal to 2α, with α the distance from any adversary A_i to prover P. In other words: $dist(i,i') \leq 2\alpha$, where *dist*(i,i') is the distance between adversary A_i and $A_{i'}$ (see Table 2.1 and Figure 2.1). Note that this is the Triangle Inequality.
We now give the strategy for each adversary A_i, $i = 1,2,\ldots,m$, with which they can pretend that the prover is present at position P.

Strategy. Before the execution of the protocol by the verifiers $\mathcal{V}$, each A_i needs to do the following: Every adversary A_i positions himself on the straight line between verifier V_i and prover P on a distance α from P (if two verifiers V_j and $V_{j'}$ are laying on the same straight line then we

[1] For the sake of simplicity, we will call 'the prover present at position P' simply 'prover P' from now on.

Parameter	
M_i	message sent by V_i
t_i	distance between V_i and A_i
α	distance between A_i and P
T_i	$t_i + \alpha$, distance between V_i and P
$dist(i,i')$	distance between A_i and $A_{i'}$, ≥ 0
$delay(i,i')$	$2\alpha - dist(i,i')$

Table 2.1: Parameters for the non-existence proof of secure position-verification in the Vanilla model

only need one adversary A_j). All the adversaries $\mathcal{A}$ are now positioned on a circle with radius α and centre P. So they are all equally far away from P.

Next, each adversary A_i listens to a message M_i sent by V_i, $i = 1,2,\ldots,m$ to P. When he has received it, he holds it for a time of $delay(i,i') = 2\alpha - dist(i,i')$.[2] Hereinafter he sends message M_i through to every other adversary $A_{i'}$, located within a distance of $dist(i,i')$ from A_i, over the covert channel between them. It then reaches $A_{i'}$ $delay(i,i') + dist(i,i') = 2\alpha$ time after the message M_i arrives at A_i (see Figure 2.1).

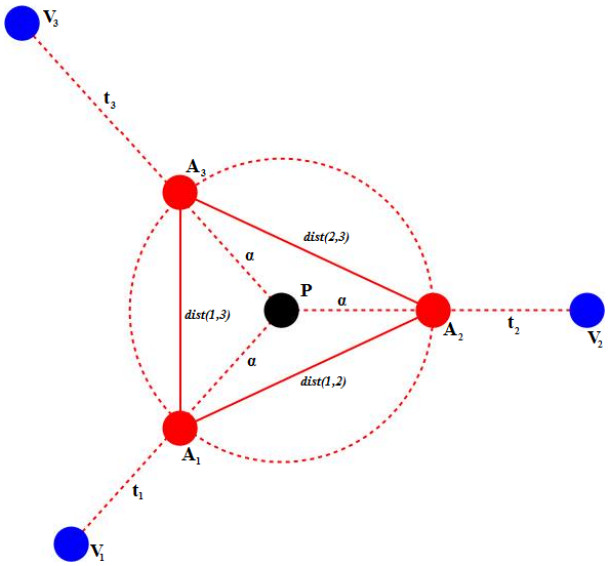


Figure 2.1: Vanilla model for Theorem 1 with three verifiers and three adversaries.

Result. We now show that if the adversaries $\mathcal{A}$ carry out the strategy described above, they can send the same message to each adversary V_i as P does and that this message arrives at the same time as when it is sent by P. In this way the adversaries $\mathcal{A}$ can pretend that the prover is present at position P even when he is not. So the verifiers $\mathcal{V}$ never see the difference between executing the protocol with prover P or with the adversaries $\mathcal{A}$.

If the verifiers $\mathcal{V}$ execute the position-verification protocol, then each verifier V_i sends a message M_i, $i = 1,2,\ldots,m$, to P. Next, P reads the messages and does the prescribed operation (since the required operation is known from the publicly accessible position-verification protocol, both P and

[2] Recall, since messages are sent with the speed of light, time and distance are the same, see Section 2.1

the adversaries $\mathcal{A}$ know what this operation is) to get a single new message M that he then sends to every verifier to show that he is present at position P. [3]

First, we look at the time it takes before the message M has reached all the verifiers $\mathcal{V}$ after each verifier V_i has sent his message M_i. Let t_i be the distance from verifier V_i to adversary A_i, and T_i the distance from V_i to prover P. Since T_i is the distance from verifier V_i to prover P, $T_i = t_i + \alpha$ (see Figure 2.1). When a verifier V_i sends his message M_i at time t, it reaches adversary A_i at $t + t_i$. After that, it arrives at P at time $t + t_i + \alpha$. So the message M_i sent by V_i arrives at prover P at time $t + t_i + \alpha = t + T_i$. If P has received the messages of all the verifiers and has processed them to obtain M, he sends it to every verifier $V_{i'}$, $i' = 1, 2, \ldots, m$. It arrives at each $V_{i'}$ at time $T_{i'} = \alpha + t_{i'}$ time after P has sent it. So message M arrives at each $V_{i'}$ at time $t + t_i + t_{i'} + 2\alpha = t + T_i + T_{i'}$, see Figure 2.1.

Now we show that the message M arrives at all the verifiers $\mathcal{V}$ at the same time when the adversaries $\mathcal{A}$ apply their strategy and intercept the messages M_i sent by each verifier V_i. As shown above the message M_i arrives at time $t + t_i$ at each adversary A_i. Adversary A_i applies its strategy and holds the message with a time duration of $delay(i, i') = 2\alpha - dist(i, i')$. It thereinafter sends it to all the other adversaries. Since each adversary $A_{i'}$ is positioned $dist(i, i')$ away from A_i, the message M_i arrives at every $A_{i'}$ at $t + t_i + delay(i, i') + dist(i, i') = t + t_i + 2\alpha - dist(i, i') + dist(i, i') = t + t_i + 2\alpha$, see Figure 2.1. Every $A_{i'}$ then receives all the messages M_i from every A_i $i = 1, 2, \ldots, m$, processes it and obtains message M.[3] Next, each $A_{i'}$ sends M to verifier $V_{i'}$ where it arrives at $t_{i'}$ time after it has reached $A_{i'}$. So if verifier V_i sends his message M_i, the message M reaches each $V_{i'}$ via $A_{i'}$ at time $t + t_i + 2\alpha + t_{i'} = t + T_i + T_{i'}$, see Figure 2.1. This is the exact same time as when the verifiers $\mathcal{V}$ carry out the protocol with P.

Conclusion. If verifiers $\mathcal{V}$ execute a position-verification protocol and the adversaries $\mathcal{A}$ apply the strategy described in this proof for their attack, the verifiers $\mathcal{V}$ will not see the difference between executing the protocol with the prover present at position P or the adversaries pretending the prover to be present at P. Thus, verifiers $\mathcal{V}$ can never securely determine the position of a prover P in the Vanilla model. □

[3] The computers of the adversaries $\mathcal{A}$ have the same computational power as the one prover P has. Therefore any computation takes the same amount of time for both adversaries and prover. So for simplicity we say that the time it takes to process the message is zero.

Chapter 3

Alternative Models

With the impossibility of secure position-verification in the Vanilla Model proven, the question arises what assumptions can be made so that position-verification could ever be secure. In this chapter, some modifications of the Vanilla Model are considered and evaluated for security.

The first section is about the Hidden-Base Model. In this model, the position of one or more of the verifiers is unknown to all players that are not verifiers. Therefore the adversaries cannot position themselves on the lines from the verifiers to position P. Clearly the impossibility proof of Section 2.2 can not be applied to this model. The last section of this chapter is about the Moving-Base Model. In this model, one of the verifiers is not fixed in d-dimensional space, but moves around in it. The (im)possibility of secure position-verification in these models is proven below.

3.1 Hidden-Base Model

In this section, secure position-verification in the Hidden-Base Model is investigated. First, a secure position-verification protocol is constructed that depends on the location of one of the verifiers being unknown to the adversaries. Next, we determine whether or not the adversaries could determine the location of a hidden verifier. Note that in Section 2.2 we have shown that when the adversaries know the location of all verifiers, they can position themselves in such a way that they can fool any position-verification protocol.

3.1.1 Secure Position-Verification in the Hidden-Base Model

In 2008, Capkun et al. proposed a position-verification protocol relying on a hidden verifier. The model they use is very similar to the Vanilla model from Chapter 2.

Model

The set of verifiers $\mathcal{V}$ is divided in a set of verifiers $\mathcal{V}_k$ whose positions are known to the non-verifier players and a set of verifiers $\mathcal{V}_u$ whose positions are unknown. The verifiers in $\mathcal{V}_u$ will be called "hidden" from this moment on. Normal verifiers act like the verifiers in the Vanilla Model. Hidden verifiers can listen to ongoing communication and communicate with the other verifiers over the covert channel only. They do not send broadcast or directional messages, so their position cannot be detected in that way.

Protocol

In the protocol of Capkun et al., a player A sends its position p_A to the verifiers in two messages simultaneously: a message m_r transported by radio waves and a message m_u transported by ultrasound. An honest player would do this via broadcast signals. Because radio waves and ultrasound messages travel at different speeds, the messages reach the verifiers at different times. Suppose that m_r and m_u reach a certain verifier V at times t_r and t_u respectively. This verifier now computes distance $d_A^m = (t_r - t_u) \cdot (v_r - v_u$, with v_r and v_s being the speed of light and the speed of (ultra)sound respectively. Verifier V checks if d_A^m is close enough to the distance $d_A^c = d(p_A, p_V)$. Here a combined localisation and ranging error Δ is taken into account. Thus when $|d_A^c - d_A^m| \leq \Delta$, V accepts the position. Else, V rejects it.

Security Proof

Suppose adversaries try to pretend some device is present at position P, when in fact there is none. An adversary A at location p_A can send directional messages m_r and m_u to all normal verifiers at times such that for all of those verifiers, $|d_P^c - d_A^m| \leq \Delta$. However, the adversary cannot chose the right times to send the messages to a hidden verifier $HV \in \mathcal{V}_u$, because he does not know the position of HV and therefore he does not know the distance to HV. The security proof of the protocol suggested by Capkun et al. (2008) is based on the probability of a hidden verifiers HV being at a distance to an adversary A for which coincidentally, $|d_P^c - d_A^m| \leq \Delta$.

We calculate the chance of success for adversary A: $P(|d_P^c - d_A^m| \leq \Delta, p_A \neq p_P)$. Back in Chapter 2 we assumed the adversaries could freely chose their position in the convex hull spanned by the verifiers. Note that when an adversary positions itself within the convex hull enclosed by the verifiers it knows the position of, it will also be positioned within the convex hull enclosed by all verifiers. Since the adversaries can chose their position freely, suppose they are positioned in such a way that their chance of successfully fooling the verifiers is optimal.
We now compute the maximum chance of success for the adversaries. For this calculation, let A be the adversary attempting to fool the verifiers. Suppose there is a hidden verifier $HV \in \mathcal{V}_u$ and define $d_A = d(A, HV)$. Now we see that

$$\begin{aligned} P(d - \Delta \leq d_A \leq d + \Delta) &= \frac{\mathrm{V}(B((0,0), d+\Delta)) - \mathrm{V}(B((0,0), d-\Delta))}{\mathrm{V}(B((0,0), R)} \\ &= \frac{\frac{4}{3}\pi(d+\Delta)^3 - \frac{4}{3}\pi(d-\Delta)^3}{\frac{4}{3}\pi R^3} \\ &= \frac{6d^2\Delta + 2\Delta^3}{R^3}. \end{aligned}$$

Notice that when the adversary is not positioned near the centre of the convex hull, part of the disc of a certain distance from the adversary lies outside of the convex hull. Since no verifier could be positioned there, the chance of the hidden verifier being located inside the disc would be smaller than $\frac{6d^2\Delta + 2\Delta^3}{R^3}$. Therefore, in order to compute the maximum chance of success for the adversary in the above computation, we assumed that the adversary is positioned near the centre of the convex hull.

We see that the chance of success decreases when the localisation and ranging error Δ decreases or when the radius of the convex hull increases. Therefore the verifiers can arrange their infrastructure in such a way that the chance of success for the adversaries becomes arbitrarily small. Hence, this position-verification protocol is secure.

3.1.2 Locating a Hidden Verifier

Now that we know that secure position-verification is possible when the location of one of the verifiers is unknown to the adversaries, the question arises if there is a way for the adversaries to effectively and efficiently determine the position of such a hidden verifier. In this section a protocol for determining the position of the hidden verifier is constructed. For this method to work, the following assumptions have to be made (Chandran et al., 2009):

- The adversaries receive feedback on whether or not their position was accepted or rejected by the verifiers.
- The method can be used $O(\log(\frac{1}{\delta}))$ times, with δ representing the precision of locating the hidden verifier by the adversaries.

The model on which the position-verification protocol is based is a slightly altered version of the Vanilla Model (Chandran et al., 2009). This model is explained in the following paragraph and afterwards the position-verification protocol is given.

Protocol for Determining the Position of a Hidden Verifier

In the position-verification protocol suggested by Capkun et al. (2008) it was assumed that the adversaries cannot guess the position of a hidden verifier with significant probability. Based only on the assumptions made previously in this section, it can be shown that the adversaries can in fact determine the location of a hidden verifier up to within a radius of δ. This is called "determining the position with δ-precision". First the procedure for locating a single hidden verifier is shown and next the procedure for finding multiple.

Assume there are three adversaries, l_1,l_2 and l_3, that we call *locators* and that do not lie on the same straight line. Because all of them are adversaries, the locators can communicate using the adversaries covert channel. The locators are positioned within the space enclosed by the verifiers. Therefore, when they send their claimed position to the verifiers and the verifiers check this position, the verifier will respond with "Accept" or "Reject". Using this response, the locators can determine the position of the hidden verifier with δ-precision.

The locators run a binary search. This binary search developed by (Chandran et al., 2009) is described in Protocol 1.

Protocol 1: Locate Single Hidden Verifier

1. Locator l_i runs the following procedure:
   (a) Let the sector in which the hidden verifier may exist be π^i. Divide this sector in two sectors π_1^i and π_2^i of equal area.
   (b) The adversaries broadcast the verification message to π_1^i, but not to π_2^i.
   (c) If the response from the verifiers is "Accept", set $\pi^i = \pi_1^i$. Else, set $\pi^i = \pi_2^i$. If π^i is 'narrow enough' to resemble a straight line, go to step 2, else go to step (a).
2. Now, knowing that the hidden verifier lies on narrow sectors π^i and π^j for $1 \leq i, j \leq 3$, the locators can compute the position of he hidden verifier with δ-precision.

Note that this protocol will not work when locators l_i and l_j and the hidden verifier lie on the same straight line, as π^i and π^j will be the same narrow sector. Hence, the assumption of three locators that do not lie on the same straight line is necessary.

The area of the sector that the hidden verifier must lie inside is decreased by half each iteration. Therefore the required number of iterations in order to achieve δ-precision is $O(\log(\frac{1}{\delta}))$. Because the smaller the required precision, the higher the number of iterations required, we see that the required number of iterations is $O(\log(\frac{1}{\delta}))$. Hence the second assumption of this Section had to be made.

In Protocol 1, the computation of the position of the hidden verifier with δ-precision can start when the sector π^i is 'narrow' enough. To clarify this, let l_1 and l_2 be two locators that return different sectors from Protocol 1 and let x be the distance between l_1 and l_2. Now let γ be the angle of the two sectors they have found using Protocol 1. Since both sectors π^1 and π^2 were found using Protocol 1, the hidden verifier must lie in the intersection of the two sectors. This intersection is enclosed by the points where the borders of sectors π^1 and π^2 intersect. Let these points be P_1, P_2, Q_1 and Q_2 as also depicted in Figure 3.1.
The locators have determined that the hidden verifier must lie in the quadrilateral $Q_1P_1Q_2P_2$. Let $y = d(Q_1, Q_2)$ and $z = d(P_1, P_2)$ be the diagonals of $Q_1P_1Q_2P_2$. To determine the position of the hidden verifier with δ-precision, we must have $\max\{y, z\} < \delta$. Note that y and z have a linear, thus polynomial, relationship with δ.

In order to calculate the lengths of y and z, first define α to be the angle between the lines $\overrightarrow{l_1P_2}$ and $\overrightarrow{l_1l_2}$ and β to be the angle between the lines $\overrightarrow{l_2P_1}$ and $\overrightarrow{l_2P_1}$. For simplicity, with A, B being points in d-dimensional space, write $\overline{AB}$ for $d(A, B)$. The calculation requires use of the Sine and Cosine Rules.

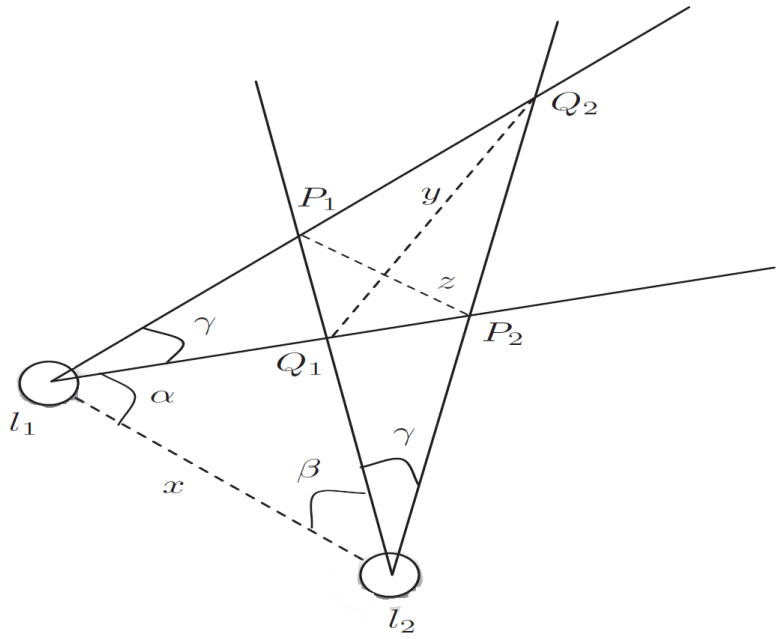


Figure 3.1: Intersection of π^i and π^j (reproduced from Chandran et al. (2009))

Theorem 2 (Sine Rule)**.** *With angles and lengths as in Figure 3.2, for any triangle we have*

$$\frac{a}{\sin\alpha} = \frac{b}{\sin\beta} = \frac{c}{\sin\gamma}.$$

Theorem 3 (Cosine Rule)**.** *With angles and lengths as in Figure 3.2, for any triangle we have*

$$\begin{aligned} a^2 &= b^2 + c^2 - 2bc\cos\alpha, \\ b^2 &= a^2 + c^2 - 2ac\cos\beta, \\ c^2 &= a^2 + b^2 - 2ab\cos\gamma. \end{aligned}$$

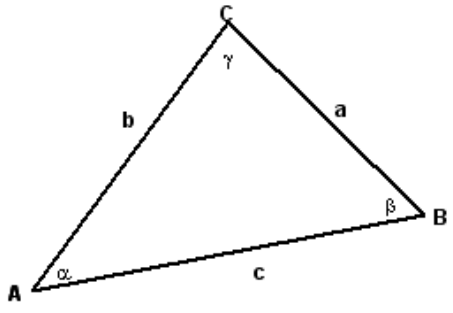


Figure 3.2: Triangle with marked angles and edges

Using the Cosine Rule we find the following values, corresponding to the values in Figure 3.1.

$$\begin{aligned} y^2 &= \overline{Q_2 l_2}^2 + \overline{Q_1 l_2}^2 - 2 \cdot \overline{Q_2 l_2} \cdot \overline{Q_1 l_2} \cdot \cos\gamma, \\ z^2 &= \overline{P_2 l_2}^2 + \overline{P_1 l_2}^2 - 2 \cdot \overline{P_2 l_2} \cdot \overline{P_1 l_2} \cdot \cos\gamma. \end{aligned}$$

We find the values of the lengths in these equations using the Sine Rule.

$$\begin{aligned}
\overline{Q_2 l_2} &= \frac{x \cdot \sin(\alpha+\gamma)}{\sin(\pi-(\alpha+\beta+2\gamma))} = \frac{x \cdot \sin(\alpha+\gamma)}{\sin(\alpha+\beta+2\gamma)}, \\
\overline{Q_1 l_2} &= \frac{x \cdot \sin(\alpha)}{\sin(\pi-(\alpha+\beta))} = \frac{x \cdot \sin(\alpha)}{\sin(\alpha+\beta)}, \\
\overline{P_2 l_2} &= \frac{x \cdot \sin(\alpha)}{\sin(\pi-(\alpha+\beta+\gamma))} = \frac{x \cdot \sin(\alpha)}{\sin(\alpha+\beta+\gamma)}, \\
\overline{P_1 l_2} &= \frac{x \cdot \sin(\alpha+\gamma)}{\sin(\pi-(\alpha+\beta+\gamma))} = \frac{x \cdot \sin(\alpha+\gamma)}{\sin(\alpha+\beta+\gamma)}.
\end{aligned}$$

Substituting these expressions into y^2 and z^2 yields

$$\begin{aligned}
y^2 &= \left(\frac{x \cdot \sin(\alpha+\gamma)}{\sin(\alpha+\beta+2\gamma)}\right)^2 + \left(\frac{x \cdot \sin(\alpha)}{\sin(\alpha+\beta)}\right)^2 - 2\left(\frac{x \cdot \sin(\alpha+\gamma)}{\sin(\alpha+\beta+2\gamma)}\right)\left(\frac{x \cdot \sin(\alpha)}{\sin(\alpha+\beta)}\right)\cos\gamma, \\
z^2 &= \left(\frac{x \cdot \sin(\alpha)}{\sin(\alpha+\beta+\gamma)}\right)^2 + \left(\frac{x \cdot \sin(\alpha+\gamma)}{\sin(\alpha+\beta+\gamma)}\right)^2 - 2\left(\frac{x \cdot \sin(\alpha)}{\sin(\alpha+\beta+\gamma)}\right)\left(\frac{x \cdot \sin(\alpha+\gamma)}{\sin(\alpha+\beta+\gamma)}\right)\cos\gamma.
\end{aligned}$$

Each step in Protocol 1, the areas of sectors π^i and π^j are divided into halves. Hence, with each iteration, the angle γ is divided by two. We seek to find a relationship between angle γ and the lengths y and z in order to find a relationship between angle γ and precision requirement δ. Expanding the sine and cosine terms to first order Taylor only series yields

$$\begin{aligned}
y^2 &= \frac{\alpha^2 x^2}{(\alpha\beta)^2} + \frac{x^2(\alpha+\gamma)^2}{(\alpha+\beta+2\gamma)^2} - \frac{2\alpha x^2(\alpha+\gamma)}{(\alpha+\beta)(\alpha+\beta+2\gamma)} &&= \frac{\gamma^2 x^2(\beta-\alpha)^2}{(\alpha+\beta)^2(\alpha+\beta+2\gamma)^2}, \\
z^2 &= \frac{\alpha^2 x^2}{(\alpha+\beta+\gamma)^2} + \frac{x^2(\alpha+\gamma)^2}{(\alpha+\beta+\gamma)^2} - \frac{2\alpha x^2(\alpha+\gamma)}{(\alpha+\beta+\gamma)^2} &&= \frac{\gamma^2 x^2}{(\alpha+\beta+\gamma)^2}.
\end{aligned}$$

Clearly, both y and z have a polynomial relationship with γ when γ approaches zero. Note that we are only interested in increasingly small angles γ, so we assume the relationship y and z have with γ is indeed polynomial.

Consider precision requirement δ. Because the relationship of y and z with δ is polynomial and the relationship of y and z with γ too, the relationship between γ and δ is polynomial. For any precision requirement δ, a certain number of iterations of Protocol 1 is required. The smaller δ, the more iterations. Hence, the required amount of iterations depends on $\frac{1}{\delta}$. Since with any iteration the angle γ is divided by half, only $O(\log(\frac{1}{\delta}))$ iterations are necessary. This is still polynomial. This assures us the suggested protocol for locating hidden verifier is applicable for efficient adaptations in position-verification programs.

In Section 4.1 the applicability of secure position-verification in the Hidden-Base Model is evaluated.

3.2 Mobile-Base Model

Another alteration of the Vanilla model which could make secure positioning possible is the Mobile-Base Model. The Mobile-Base Model is a position-verification model based on moving verifiers, called Mobile-Base Stations (MBS). Apart from that the verifiers are moving, for which they will need a slightly different position-verification protocol, the MBS model works the same as the Vanilla model. The MBS check if someone is present at position P and adversaries want to 'spoof' (make it seem that someone is present at) this position. First we construct a mobile position-verification protocol. After that we check whether or not adversaries can still spoof position P, even though the verifiers are moving.

3.2.1 Position-Verification Protocol for the Mobile-Base Model

For the altered Vanilla model with MBS, Capkun et al. give a position-verification protocol for a single MBS. When there are multiple MBS they all execute this protocol. The protocol is based on the time difference of arrival for a by the MBS broadcasted message. Before we look at this, we need to make a couple of assumptions:

- There exists a circle around the position P with radius R within which an MBS does not enter when it executes the position-verification protocol. We can make this assumption, because if there would not be such a circle an MBS could verify if someone is at position P just by going to position P and check if there is someone there.
- The MBS move uniformly at random over the entire disk of radius R', the area where position-verification takes place.
- MBS have an error Δ that they can tolerate in the delay of receiving a signal response from a node. Without this error the MBS could almost securely do their position-verification protocol (except for when all the adversaries lie on the straight line between the MBS and P) and position-verification by, for example, satellites is not always completely precise.

Mobile Position-Verification Protocol

Because we now deal with Mobile Base Stations, also the prover P does not know where to send his received message back to. So we have a slightly different protocol than the regular position-verification protocol in the Vanilla model.

Both the MBS and the prover P broadcast their messages. The MBS wants to know the position of P by checking how long it takes to get his message back after he has broadcasted it. This method is called the Time Difference of Arrival method. Since the MBS itself is moving, the position he measures is also influenced by the distance he travels. So he wants to know the position of P regardless of the time (and thus the distance) that he needs to move to an other position. To accomplish this, an MBS sends his message with extra information T_P about how long the prover P must delay his message before sending it back. If the MBS sends his message at t_1 and arrives at an other place at t_2 this delay will be $T_P = t_2 - t_1$, the time it took fot the MBS to go from the first place to the second. As one can see in Figure 3.3, if a MBS broadcasts a message at t_1, then it reaches P at $t_1 + t_{p_1}$. Prover P then holds it for $T_P = t_2 - t_1$ and after that broadcasts it at time $t_1 + t_{p_1} + t_2 - t_1 = t_{p_1} + t_2$. The message reaches the MBS again at time $t_2 + t_{p_1} + t_{p_2}$. So at the

same time as MBS only with the extra time it took the message to go from the MBS to P and back to the MBS at the next point. In this way the MBS can compute P's position by the time difference of arrival independently of his own movement.

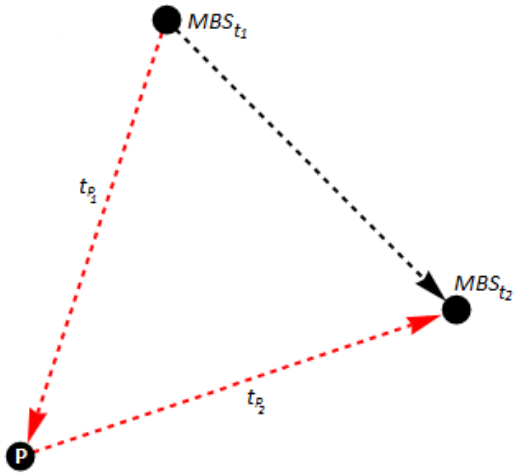


Figure 3.3: Image for the mobile position-verification protocol

3.2.2 Attack on the Position-Verification Protocol for the Mobile-Base Model

In their article, Capkun et al. (2008) have shown that MBS can securely carry out a mobile position-verification protocol in order to verify position P. They assume the base stations to move randomly, and hence the adversaries cannot pre-determine the position of the base stations. But in a response on this article Chandran et al. (2009) have shown that his assumption is not valid and that adversaries can even make moving verifiers believe that there is someone on position P. And because of this, MBS cannot securely determine if someone is at position P. We show how this attack on the mobile position protocol works and in Chapter 4 discuss to what extend this attack is feasible.

Attack on the Protocol in Mobile Base Stations Model

The area within which the position-verification takes place is a disk with radius R' and centre P, the place that the adversaries want to spoof (see Figure 3.3). We only look at the disk with radius R' since Capkun et al. (2008) have shown that it is sufficient to show it for a disk with a random radius (which is in our case R'). Let Δ be the error in time (and thus distance) that the MBS tolerate when they receive a response signal from P.

Now the strategy for k adversaries $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$ is to first place themselves on the boundary of the disk with radius R and centre P inside the disk with radius R'.[1] These adversaries are thus positioned on a distance R from P.

When the adversaries receive a signal from an MBS they then will delay it with a time duration of $2R$ before broadcasting it back with an angle α in sector L_i, A_i, R_i with $R_i = L_{i+1}$ and radius

[1] We have assumed that R is small enough such that there is a high probability that the MBS will not enter the disk.

$L_iA_i = a$ (see Figure 3.4).

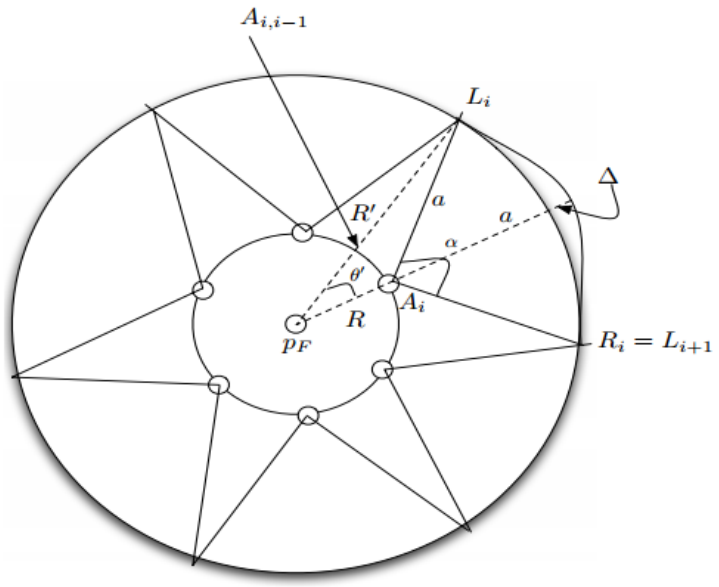


Figure 3.4: Image from the article of Chandran et al. (2009) which shows the situation where adversaries attack the position-verification protocol.

If each adversary A_i sends back the message with radius α "small enough", then any MBS in sector L_i, A_i, R_i will accept position P. This, because the smaller α gets, the more an adversary approaches the straight line between the MBS in sector L_i, A_i, R_i and P. Thus the better an adversary can spoof position P.[2] This angle α must be so small that the response message from an adversary reaches the MBS within the error time Δ. We later discuss how small α exactly must be to accomplish this (this depends on the angle θ' and the number of adversaries).

An MBS in area $A_iL_iA_{i-1}$ does not receive a signal from any adversary. So if the adversaries want to succeed in spoofing position P the MBS must not be in that area. So for one MBS the probability λ that adversaries can pretend that there is someone at position P is the ratio of the area L_i, A_i, R_i within which the adversaries are able to spoof the location to the total area L_i, A_{i-1}, A_i, R_i for $i = 1, 2, \ldots, k$ (this ratio is the same for every i). Thus for multiple MBS the probability that the adversaries can spoof position P is $\lambda^{|MBS|}$.

Now assume an MBS is in sector L_i, A_i, R_i (and thus not in area $A_iL_iA_{i-1}$). Since the MBS is moving and hidden, the adversary A_i will not know his exact place. So he (except for when he accidentally is) won't be positioned on the straight line between the MBS and P.[2] If the MBS broadcasts his signal and A_i receives it, delays it with $2R$ time and broadcasts it back in an angle of α, it will reach the MBS later than the message send back by P, since it is not on the straight line between the MBS and P. But now, by assumption, there is an error Δ that the MBS tolerates when receiving a signal response from a node. So if the message send back by the adversary A_i

[2] We have seen in chapter 2 that if each adversary A_i lies on the straight line between the verifier V_i and P, the verifies $\mathcal{V}$ cannot successfully carry out a secure position-verification protocol.

arrives at the MBS within the error bound Δ, the adversary can succesfully "spoof" position P.

Number of Adversaries Needed to Spoof Position P

The more adversaries we have positioned on the disk with radius R from P, the higher the chance of one being very close to the straight line between the MBS and P. And thus the bigger the chance their response message reaches the MBS within the error bound Δ. So we want to determine the number of adversaries needed such that the response signal of the adversaries will always arrive at the *MBS* on time, i.e. wont exceed the error bound Δ. First we look at the angle θ', which is the angle between $\overline{PA_i}$ and $\overline{PL_i}$ (see Figure 3.4). As one can see, this angle depends on the number of adversaries k. The response signal does not exceed the error bound if this angle θ' will be such that $a+R-R' = \mu \leq \Delta$. In other words: if the adversaries want to spoof location P successfully, then the surplus μ of the distance $R = d(A_i, P)$ plus the radius a with which an adversary broadcasts his signal minus the radius R of the disk must be less then the error bound Δ. You can see in Figure 3.5, Figure 3.6 and Figure 3.7 that this surplus μ depends on θ' and thus on the number of adversaries.

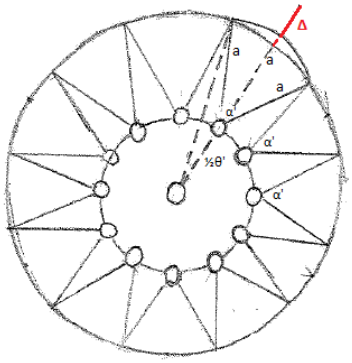


Figure 3.5: Twelve adversaries, $^1/_2\theta'$

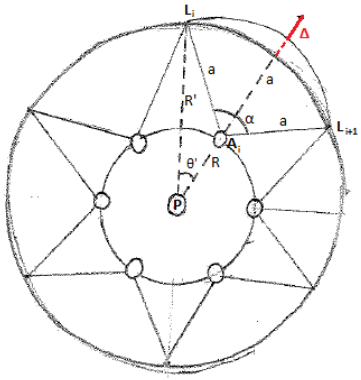


Figure 3.6: Six adversaries, θ'

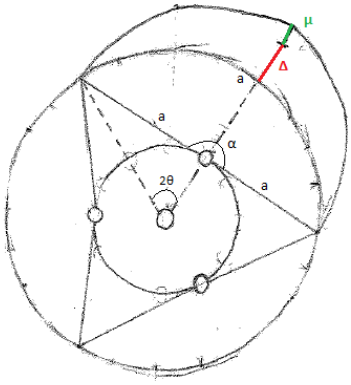


Figure 3.7: Three adversaries, $2\theta'$

Note that fixing θ' gives us a value for a which gives us the amount with which we will or will not exceed the error bound Δ. We want to have an upper bound θ for θ' such that any smaller θ' than θ will not make the adversaries' response signal exceed the error bound Δ. Once we have determined this upper bound θ we can compute the number of adversaries needed to have θ' be smaller or equal to this upper bound.

Using the Cosine Rule we find the following values, corresponding to the values in Figure 3.4.

$$a^2 = R^2 + R'^2 - 2 \cdot R \cdot R' \cdot \cos\theta.$$

And thus,

$$\begin{aligned}\cos\theta &= \frac{-a^2+R^2+R'^2}{2\cdot R\cdot R'}\\ &= \frac{a^2-R^2-R'^2}{-2\cdot R\cdot R'}\end{aligned}$$

Since for upper bound θ the following must hold $a+R-R' \leq \Delta$ and thus $a \leq -R+R'+\Delta$ to not exceed the error bound, we can implement this in our equation. Giving

$$\cos\theta \leq \frac{(\Delta+R'-R)^2-R^2-R'^2}{-2\cdot R\cdot R'}.$$

So

$$\theta \leq \cos^{-1}\left(\frac{(\Delta+R'-R)^2-R^2-R'^2}{-2\cdot R\cdot R'}\right).$$

Giving this upper bound one can see in Figure 3.4 that the number of adversaries must be always more or equal to $k = 2\pi/2\theta = \pi/\theta$.

Probability of Spoofing Position P

Now we know how many adversaries we need to let the MBS believe that someone is positioned at position P, we also want to know how big the chance λ is that we achieve this. This is the chance that for each adversary A_i the MBS is in sector $L_iA_iR_i$, the "good area", and not in the "bad area" $A_{i-1}L_iA_i$ in where adversary A_i cannot spoof position P (see Figure 3.4 and the detailed version in Figure 3.8). This chance is determined by dividing the size r_{good} of the good area by the size r_{total} of the total area $\lambda = \frac{r_{\text{good}}}{r_{\text{total}}}$, where the size of the total area is the size of the good area plus the size r_{bad} of the bad area.

The area A of a circle sector with radius R and angle θ is $A = \frac{r^2\theta}{2}$. As one can see in Figure 3.4 and Figure 3.8 the total area r_{total} for a single adversary A_i is the surface of the area $A_{i-1}L_iA_i$ plus area $L_iA_iR_i$. This is the same as the sector L_iPR_i with radius R' minus the same sector L_iPR_i with radius R. So the size of the total area is

$$\begin{aligned}r_{\text{total}} &= \frac{R'^2\cdot 2\theta'}{2} - \frac{R^2\cdot 2\theta'}{2}\\ &= \frac{(R'^2-R^2)2\theta'}{2}\\ &= (R'^2-R^2)\theta'.\end{aligned}$$

Now instead of looking at the good area we can also look at the "bad area" $A_{i-1}L_iA_i$ that the MBS may not enter if the adversaries successfully want to spoof position P, since $\lambda = \frac{r_{\text{good}}}{r_{\text{total}}} = 1 - \frac{r_{\text{bad}}}{r_{\text{total}}}$. We want to know the size r_{bad} of the bad area $A_{i-1}L_iA_i$. Herefore we look at the surface of triangle $\triangle A_{i-1}L_iA_i$ with as *height* the distance from $A_{i,i-1}$ to L_i a and as a *base* the straight line $\overline{A_{i-1}A_i}$ between A_{i-1} and A_i (see Figure 3.8).

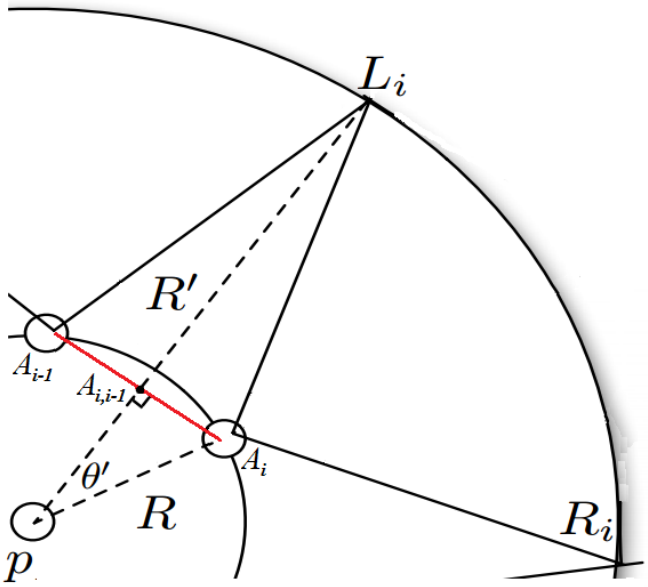


Figure 3.8: Detailed version of Figure 3.4 with good area $L_iA_iR_i$ and bad area $A_{i-1}L_iA_i$.

The *height* $h_{\triangle A_{i-1}L_iA_i}$ of triangle $\triangle A_{i-1}L_iA_i$ is $R'-R$ plus the height of the circular segment, which is the part of triangle $\triangle A_{i-1}L_iA_i$ inside the disk with radius R and centre P. The height h_A of a circular segment A with radius R and angle θ is $h_A = R(1-\cos(\theta/2))$. So the height $h_{\text{cseg}(\triangle A_{i-1}L_iA_i)}$ of the circular segment with radius R and angle $2\theta'$ inside triangle $\triangle A_{i-1}L_iA_i$ is $h_{\text{cseg}(\triangle A_{i-1}L_iA_i)} = R(1-\cos\frac{2\theta'}{2}) = R(1-\cos\theta')$. Now, the height $h_{\triangle A_{i-1}L_iA_i}$ of triangle $\triangle A_{i-1}L_iA_i$ is

$$\begin{aligned} h_{\triangle A_{i-1}L_iA_i} &= R' - R + h_{\text{cseg}(\triangle A_{i-1}L_iA_i)} \\ &= R' - R + R(1-\cos\theta') \\ &= R' + R\cos\theta'. \end{aligned}$$

The length of *base* $\overline{A_{i-1}A_i}$ is twice the opposite site of θ' in the triangle $\triangle A_{i,i-1}PA_i$, where $A_{i,i-1}$ is the position in the middle of $\overline{A_{i-1}A_i}$ (see Figure 3.8). So

$$\overline{A_{i-1}A_i} = 2\cdot R\cdot\sin\theta'.$$

Since the size of a triangle is $\frac{\text{base}\cdot\text{height}}{2}$, the size $r_{\triangle A_{i-1}L_iA_i}$ of the triangle $\triangle A_{i-1}L_iA_i$ is:

$$\begin{aligned} r_{\triangle A_{i-1}L_iA_i} &= \frac{2\cdot R\cdot\sin\theta'\cdot(R'-R\cdot\cos\theta')}{2} \\ &= R\cdot\sin\theta'\cdot(R'-R\cdot\cos\theta'). \end{aligned}$$

To determine the size r_{bad} of the bad area $A_{i-1}L_iA_i$ we need to subtract the size of triangle $\triangle A_{i-1}L_iA_i$ inside the disk with radius R from the size of triangle $\triangle A_{i-1}L_iA_i$ (see Figure 3.8). This is the circular segment with radius R and angle $2\theta'$. The size r_{cseg} of a circular segment is the

area of the circle sector minus the area of the triangular portion, so

$$r_{\text{cseg}} = \frac{R^2}{2}(2\theta' - \sin(2\theta')).$$

So the size r_{bad} of the bad area $A_{i-1}L_iA_i$ is

$$\begin{aligned} r_{\text{bad}} &= r_{\triangle A_{i-1}L_iA_i} - r_{\text{cseg}} \\ &= R \cdot \sin\theta' \cdot (R' - R \cdot \cos\theta') - \frac{R^2}{2}(2\theta' - \sin(2\theta'), \end{aligned}$$

and the chance of spoofing position P successfully is

$$\begin{aligned} \lambda &= \frac{r_{\text{good}}}{r_{\text{total}}} \\ &= 1 - \frac{r_{\text{bad}}}{r_{\text{total}}} \\ &= 1 - \frac{R \cdot \sin\theta' \cdot (R' - R \cdot \cos\theta') - \frac{R^2}{2}(2\theta' - \sin(2\theta')}{(R'^2 - R^2) \cdot \theta'} \\ &= -\frac{R \cdot R' \cdot \sin\theta' - \theta' \cdot R'^2}{(R'^2 - R^2) \cdot \theta'}. \end{aligned}$$

If we have k MBS then the chance of spoofing P successfully is the chance that all the adversaries stay in the good area. So this chance is simply λ^k.

3.3 Other Alternative Models

We have looked at two obvious alterations of the basic insecure Vanilla model; the Hidden-Base Model and the Mobile-Base Model. It has been show that secure positioning is not possible with these alternative models, but there could be other alterations of the Vanilla Model that might make secure positioning possible. Although we do not review other models in this article, there are some that might be interesting to research. Three of these other alterations on the Vanilla model are:

- **The Bounded-Retrieval Model**. In this model, the amount of information adversaries can receive and store is limited. The existence of this alteration is plausible since in some situations verifiers can sent a big amount of information at a very high speed, by which adversaries can only receive a constant fraction of it. This, for example, happens when verifiers have several sources where they broadcast there information and when they also broadcast it at different frequencies. Chandran et al. (2009) have shown that there exist a protocol in this model that is secure against any attack by adversaries. However, previous papers have said the same about the Hidden-Base Model and Mobile-Base Model, but we have seen that this is not the case. If someone could come up with an attack build on the same assumptions as the position-verification protocols of the verifiers, then the Bounded-Retrieval model is unsafe as well.

- **Quantum-information**. Since in the attacks on the Vanilla Model, the Hidden-Base Model and the Moving-Base Model adversaries intercept the information sent by the verifiers, hold it and then forward it to the other adversaries an obvious alteration could be the use of

quantum information. Because the *no-cloning theorem* does not allow for making perfect copies of quantum information (Wootters and Zurek, 1982), it is not possible to copy a quantum message in order to hold a copy for as long as required to spoof the verifiers. It has however already been shown by Buhrman et al. (2014) that even with quantum-information secure positioning is not possible.

- **The Noisy-Channel Model**. Another alteration is to make use of so called "noisy channels". In this model, due to imperfections in broadcasting or reception hardware, a random alteration of the sent information is detected by the players. This random alteration is called *noise*. Noise depends on the hardware used by either the broadcasting or reception hardware and on the distance travelled by the message. As a result, the noise is independent for each receiver. Dziembowski and Zdanowicz (2014) prove that secure position-verification is possible when noisy channels are taken into account.

Chapter 4

Model Evaluation

In this chapter the security of position-verification in the different models discussed in previous chapters is evaluated with an emphasis on applicability. Following Chapter 3, first the Hidden-Base Model (HMB) en then the Mobile-Base Model (MBM) are reviewed.

4.1 Security in the Hidden-Base Model

We have seen that secure position-verification is possible when the location of one or more of the verifiers is unknown to the adversaries. Recall the chance of success for the adversaries is $P(d-\Delta \leq d_A \leq d+\Delta) = \frac{6d^2\Delta+2\Delta^3}{R^3}$. Note that none of the variables depend on the adversary. Therefore, future improved adversary computers will not be able to achieve higher chances of successfully fooling the verifiers. Moreover the localisation and ranging error Δ positively relates to $P(d-\Delta \leq d_A \leq d+\Delta)$. Therefore, future improved localisation and ranging hardware that allow for smaller Δ will decrease the adversaries chance of success. This error Δ is discussed more in Section 4.3.

The security proof of Capkun et al. relies on the hidden verifier remaining hidden. Hence, the protocol constructed by Chandran et al. for locating a hidden verifier poses a fatal problem. At most $O(\log(\frac{1}{\delta}))$ attempts are needed to locate any hidden verifier (Chandran et al., 2009). When all verifiers are located, the adversaries can successfully fool the verifiers in the way described in the impossibility proof of Section 2.2.

A possible way for the verifiers to prevent the successful localisation of some or all hidden verifiers is to allow only a predetermined number of attempts to connect to the verification infrastructure by any device. This predetermined number of attempts has to be significantly smaller than $O(\log(\frac{1}{\delta}))$. Such a limitation on the infrastructure would impose problems on the applicability of this protocol:

1. When individual devices are limited in their number of attempts to connect to the verification infrastructure, multiple adversaries can work together to locate a hidden verifier if their added number of allowed attempts reaches $O(\log(\frac{1}{\delta}))$.

2. When the limitation of attempts is enforced on more devices combined, any honest device is retained from connecting to the verification infrastructure if adversaries, intentionally or not, attempt to connect more often than allowed. Though this does not render the position-verification protocol insecure, it does make it impractical for applications.

4.2 Security in the Mobile-Base Model

In Section 3.2 we have seen another way of trying to determine if someone is at a given position P by using moving verifiers, called Mobile-Base Stations (MBS). We also have seen that adversaries can still carry out an attack to let these MBS believe that someone is at position P even when he is not. In this section we are going to look at the applicability of this attack. Is it realistic to say that adversaries can easily spoof position P when they apply their attack as described in Section 3.2? There are two obstacles which could prevent adversaries from successfully spoofing position P:

- The number of adversaries needed could be too high for practical implementations. Since the response signal from the adversaries has to arrive at the MBS within the error bound Δ, the number of adversaries needed to achieve successful spoofing when the error bound Δ is small could be very high.
- The adversaries could have a low chance of success because the area in where the adversaries cannot spoof position P is too large compared to the area in where they can spoof position P.

Assume that the disk where the MBS are has a radius $R' = 500m$, the disk that we assume the *MBS* will not enter has a radius of $R = 50m$ and the MBS have an error bound of $\Delta = 2.5m$.
Let us assume that the MBS is in the area in where adversaries can spoof position P. We need to have enough adversaries such that the delay of the signal broadcast back by the adversaries arrives at the MBS within the error bound Δ. As shown in Section 3.2, we need to have $k = \frac{\pi}{\theta}$ adversaries to achieve this, where θ, the upper-bound of the angle θ', is

$$\theta \leq \cos^{-1}\left(\frac{(\Delta + R' - R)^2 - R^2 - R'^2}{-2 \cdot R \cdot R'}\right).$$

If $R' = 500m$, $R' = 50m$ and $\Delta = 2.5m$ we need to have $k = 11$ adversaries to stay within the error bound $\Delta = 2.5m$, since

$$\begin{aligned}\theta &\leq \cos^{-1}\left(\frac{(2.5 + 500 - 50)^2 - 50^2 - 500^2}{-2 \cdot 50 \cdot 500}\right) \\ &= \cos^{-1}\left(\frac{-47743.75}{-50000}\right) \\ &\approx \frac{\pi}{10.418}.\end{aligned}$$

So if we take upper bound for θ' to be $\theta \approx \frac{\pi}{10.418}$, then number of adversaries is $k = \pi/\theta = \pi/\frac{\pi}{10.418} \approx 10.4$. So we need 11 adversaries.
Now assume we have 11 adversaries, i.e. that the response signal from the adversaries always arrives at the MBS within the error bound $\Delta = 2.5$, but we do not know if the MBS are in the area in where the adversaries can pretend that the prover is at position P (the "good area"). Recall that the chance that an MBS is in the good area, and thus that the adversaries can successfully spoof position P, is

$$\lambda = -\frac{R \cdot R' \cdot \sin\theta' - \theta' \cdot R'^2}{(R'^2 - R^2) \cdot \theta'}.$$

And if we have k MBS than the chance of success is λ^k. In this example where we have $R' = 500m$, $R' = 50m$ and $\Delta = 2.5m$ (and 11 adversaries) this chance is for a single MBS

$$\begin{aligned}\lambda &= -\frac{500 \cdot 50 \cdot \sin(\frac{\pi}{10.418}) - \frac{\pi}{10.418} \cdot 500^2}{(500^2 - 50^2) \cdot \frac{\pi}{10.418}} \\ &= -\frac{7425.199 - 74251.993}{74635.529} \\ &\approx 0.91.\end{aligned}$$

So the chance of success against a single MBS is approximately $\lambda \approx 0.91$. Even with 20 MBS in the disk with radius $R' = 500m$, the adversaries still have a 0.12 chance of spoofing position P. If we want this chance to be below .01 we need 44 MBS. If we have $R' = 2000m$, $R = 50m$ and $\Delta = 2.5m$ we still need 11 adversaries to stay within the error bound and our chance of success with a single base-station is approximately .98.

Chandran et al. (2009) have shown how the number of adversaries needed to successfully spoof P depends on the sizes of R', R and Δ. Since only the ratios of these parameters to oneanother matter and not the actual values, Chandran et al. have plotted two graphs with different values of these ratios (See Figure 4.1 and Figure 4.2).

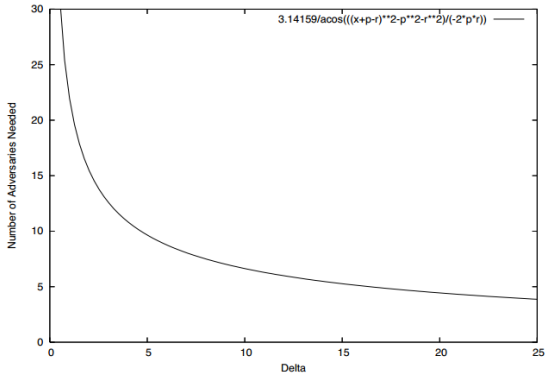


Figure 4.1: Graph by Chandran et al. (2009) where the number of adversaries is plotted as a function of Δ with $R = .5R'$ and $\Delta = .01R'$.

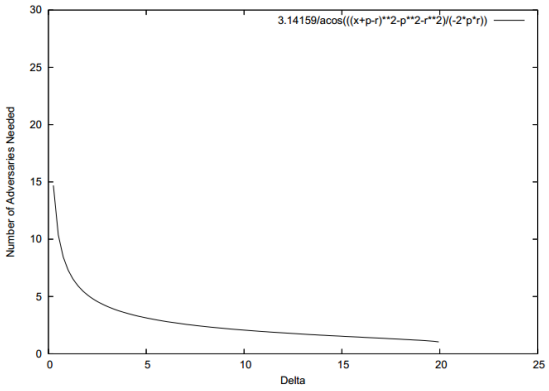


Figure 4.2: Graph by Chandran et al. (2009) where the number of adversaries is plotted as a function of Δ with $R = .1R'$ and $\Delta = .01R'$.

We see in both graphs that we do not need a lot of adversaries to stay within the error bound Δ. So the number of adversaries needed is not too high for practical implementations. Only when the error bound Δ is very small and the radius of R' compared to R' very big, then the number of adversaries needed to stay within the error bound Δ could become very big. So if in the future improved hardware could make positioning possible with a very small Δ, then secure positioning might be possible.

Chandran et al. (2009) have also looked at the probability of successfully spoofing P in the "worst case scenario". Where they defined the worst case to be a very small error bound $\Delta = .00001$ for

a fixed R/R' ratio, because then a large number of adversaries is needed to stay within the error bound Δ. But one can see in Figure 4.3 that even with this restrictions the worst case probability of success will still be very high (always bigger than 0.5).

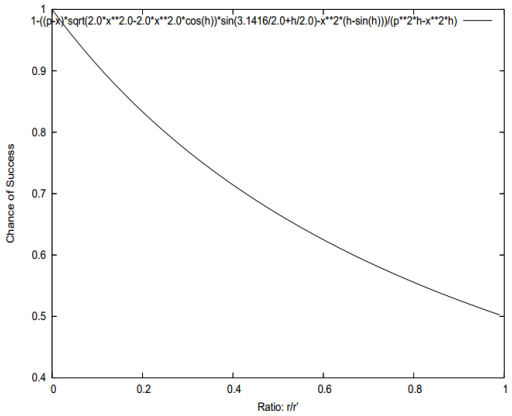


Figure 4.3: Graph by Chandran et al. (2009) where the chance of success is given as a function of the ratio R/R' with a very small error bound $\Delta = .00001$.

So the number of adversaries needed to stay within the error bound Δ in the attack for the Mobile-Base Station Model is very small and even the worst case probability of success is always very high. Thus it is realistic that the attack described in Section 3.2 is easy to carry out and has a high probability of success.

4.3 Influence of Localisation and Ranging Error Δ

As we have seen before, both in the Hidden-Base Model and the Moving-Base Model, the security of position-verification protocols could increase with a decreased localisation and ranging error Δ. This error is taken into account in the first place, because localisation and ranging hardware is never completely accurate. However, in time this localisation and ranging hardware could improve, allowing for smaller Δ.

For the Moving Base Model, a smaller Δ potentially improves the security of position-verification protocols. This is because from a very small Δ on, the number of required aversaries would increase fast, see Figures 4.1 to 4.2. If improved future hardware allows for a Δ small enough, position-verification protocols based on the Moving Base Station could be secure.

For the Hidden Base Model, a smaller Δ does not necessarily improve the security of position-verification protocols. When the limited number of attempts to connect to the verification infrastructure, discussed in Section 4.1, are implemented, a smaller Δ improves the security. However protocols using this implementation are not suited for application, as seen before. When unlimited attempts to connect can be done by the adversaries, the security improvement caused by smaller Δ is not sufficient. This is because smaller Δ in this case results in possibly smaller δ. As we have shown in Section 3.1, the effort that the adversaries have to make to locate a hidden verifier with

δ-precision relates to δ with $O(\log(\frac{1}{\delta}))$. Clearly, even for extremely small δ, this effort doesn't increase much.

Chapter 5

Conclusion

In this article we have reviewed possibilities for secure position-verification in the Vanilla Model, the Hidden-Base Model and the Moving-Base Model. The Vanilla Model is the most basic model. Here, no limitations on the abilities or the knowledge of the adversaries are assumed. We have proven that the adversaries can always make it look as if some device is present at a certain position P, regardless the protocol run by the verifiers. To do this, the adversaries have to position themselves at equal distance to P, in a way such that an adversary is positioned on every line from a verifier to P.

The Hidden-Base Model is similar to the Vanilla Model, but here the position of one of the verifiers is unknown to the adversaries. We have shown that a position-verification protocol suggested by Capkun et al. can only be secure when assumptions are made that would make the protocol impractical for applications. Since without these assumptions, the position of any hidden verifier can efficiently be determined by the verifiers, still no practical secure position-verification protocols exist that make use of hidden verifiers.

In the Moving-Base Model, instead of hidden verifiers, there are moving verifiers. Each moving verifier moves uniformly at random in a disk with radius R' and with centre P, the position where the moving verifiers want to check for the presence of a device. We have shown that even with moving verifiers adversaries can successfully pretend a device to be present at this position with chances high enough to say that the moving position-protocol given by Capkun et al. is insecure. This is because, if moving verifiers execute a mobile position-verification protocol and adversaries position themselves around position P, they can cover for moving verifiers in a significantly high percentage of this area and do not need many adversaries to achieve successful spoofing.

Based on our research on these three models, we conclude that secure position-verification based on either hidden or moving verifiers is not possible. We do recommend research into secure position-verification in other alterative models, such as but not limited to the Bounded-Retrieval Model, the Noisy-Channel Model and models making use of quantum information.

Chapter 6

Popular Summary

Doing homework is often a boring activity. Especially when the sun is shining and your parents want you to sit at school and do boring math exercises, instead of going outside to play football. Would it not be nice if you, with the help of your friends, could always make your parents believe that you are doing your sitting at school and doing your homework when in fact you are playing football outside? In this article about *Position-Based Cryptography*, the field of cryptography that deals with the question whether or not position can be part of an encryption, we have examined just that. We have shown that for the most basic model with which one could check this and for some obvious alterations on this basic model, it is impossible to securely determine someones position.

Suppose you are studying mathematics at the Amsterdam Science Park and your mother wants check if you are doing your homework. She has three radio masts and wants to verify if your at your desk (see Figure 6.1). She wants to accomplish this by letting the masts execute a publicly known protocol in which the first two masts on the left side sent a message to you that you should combine and sent to the mast on the right. We call these masts *verifiers*, the position where you should be sitting the position P and you and your friends outside the *adversaries*.

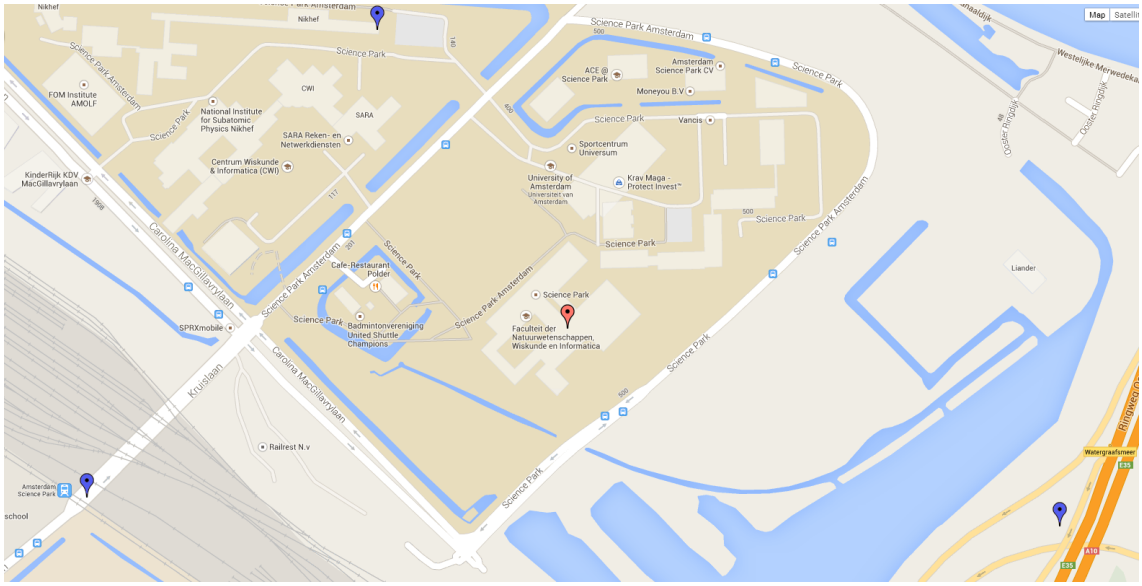


Figure 6.1: Three masts and the place where you should be doing your homework.

If the adversaries now position themselves on a circle with radius α around position P each on the straight line between the mast and position P, then you are always able to let the verifiers (and thus your mom) believe that you are on position P, when in fact you and your friends are just 'chilling' outside. This, because by triangle inequality the distance from one adversary A_1 to another A_2 is always shorter than 2α; the distance α from adversary A_1 to P plus again the distance α from P to the other adversary A_2 (see Figure 6.2). So if adversary A_1 holds it for a time of $delay(1,2) = 2\alpha - dist(1,2)$, which is the time it takes for the message to go from A_1 to P and from P to A_2 (so 2α) minus the time $dist(1,2)$ it takes for a message to go from A_1 to A_2, and then sends it, then a message arrives at each adversary at the same time as when it has been sent by the verifier. Because after a message has reached adversary A_1 it arrives $delay(1,2)+dist(1,2) = 2\alpha - dist(1,2)+dist(1,2) = 2\alpha$ time later at adversary A_2 and when he sends it through to the verifier directly the message will arrive there at the exact same time as when it has been sent by a verifier directly.

Now since the protocol is publicly known, the adversaries that are positioned in between the two left verifiers and P both know that they need to send their intercepted message to the adversary positioned in between the right verifier and P, which needs to send the two messages to the right verifier. The same message arrives at the same time at the right verifier when it has been intercepted by the adversaries as when it has been sent by the verifiers and so your mom cannot see the difference between you chilling outside and you sitting at your desk studying. Therefore you can always make your mom believe that you are studying when in fact you are not.

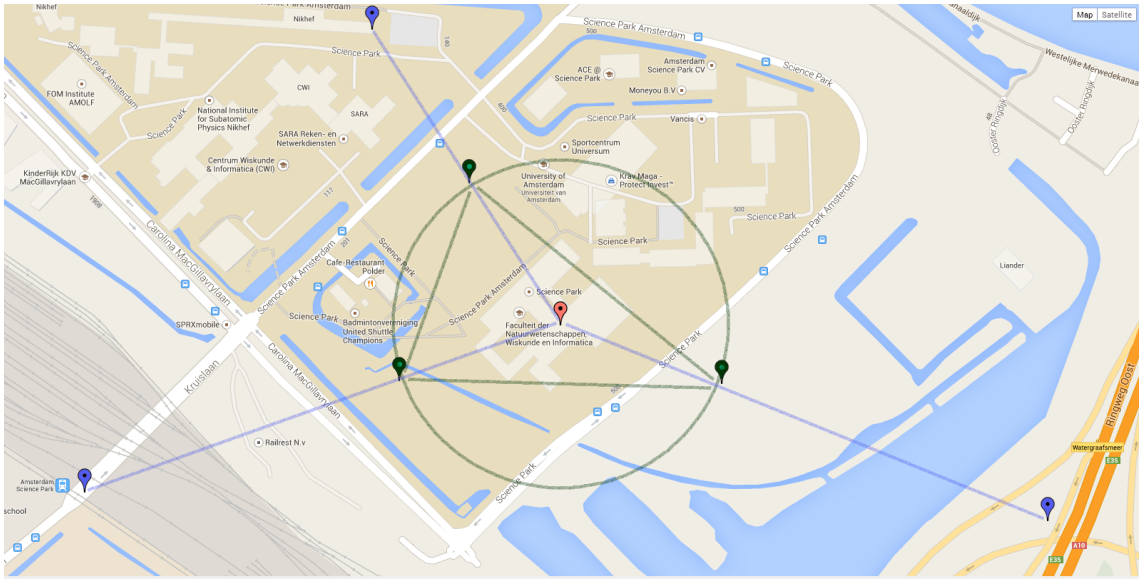


Figure 6.2: By triangle inequality, the distance from one adversary to the other is always shorter than the distance from the adversary to P and from P to the other adversary.

We now know that when you know the location of all verifiers, you and your friends can pretend to be studying at Science Park. But what if the location of one of the verifiers is unknown to you? The chance of this verifier coincidentally being at a location for which your plan still works, is

very small. This chance depends on the total size of the area enclosed by the verifiers. When coincidentally fooling your mother does not have a high chance of success, you try to take coincidence out of the equation. If you could locate that hidden verifier, you would know where to position yourself and your friends in such a way that you could again successfully pretend to be studying at Science Park. Suppose that prior to the day that your mother wants you to go study, you go to Science Park to prepare for the location-fooling operation. This time, you do send your actual position to the verifiers. However, you only send your location to half the area that the verifiers can be in. Now, when they accept your position, you know that the hidden verifier must be somewhere in the area that you sent your location to. When they do not accept your position, you know that the hidden verifier must be somewhere in the area that you did not send your position to. With every time you repeat this, the area in which the hidden verifier could be is divided into halves. In no time, you know the location of the verifier precise enough to, when the day comes that your mother wants you to study at Science Park, be able to successfully pretend to be doing just that, whilst actually being outside of Science Park playing football.

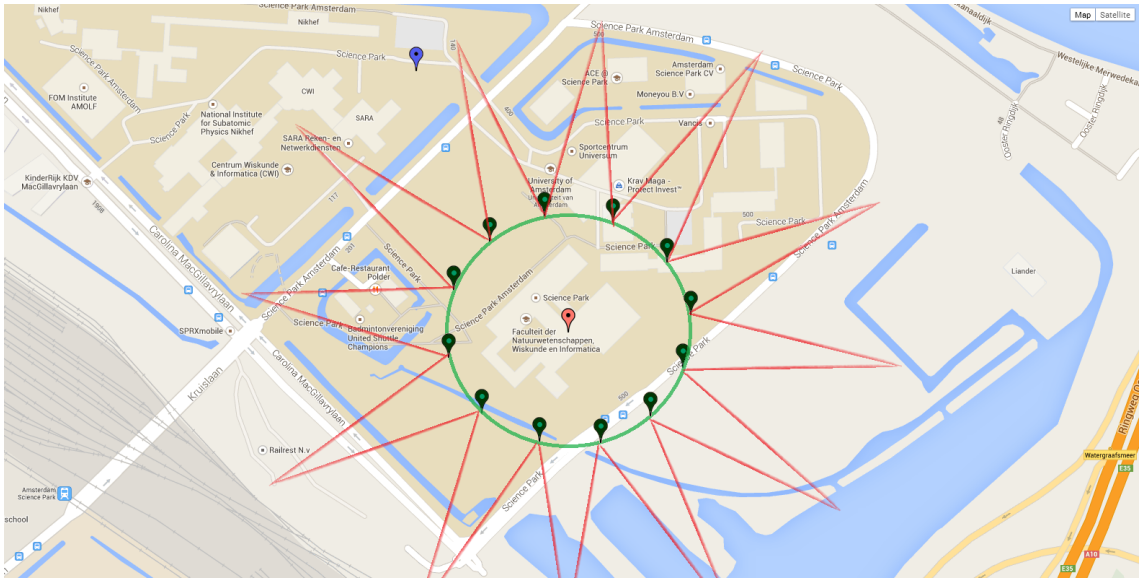


Figure 6.3: Method for the verifiers to position themselves around position P with one moving verifier, so they can successfully pretend someone to be present at position P.

Since it does not matter whether the verifiers are hidden or are not, your mother needs an other method to successfully check if you are studying at Science Park. She now tries using moving verifiers that broadcast their signal when checking if you are studying at position P. Because they are moving, you and your friends (the adversaries) cannot position themselves on the straight line between the verifiers and the position P where you should be studying. In this article we have shown that even with moving verifiers it is still possible to pretend that someone is at position P when he is not. Since verifiers (which are in our example radio masts) are measuring devices they always have an error bound when they measure something. Adversaries can make good use of this and still successfully make it seem that someone is at position P. If verifiers position themselves all on the circle around P and each adversaries takes care for an area of the space where the moving verifiers could be as shown in Figure 6.3, then they can successfully make it seem that someone is

present at position P. This because the more adversaries you have the more each adversary comes close to the straight line between the verifier and P. We have shown in our article that you do not many adversaries to accomplish this. The only problem is, as you can see in Figure 6.3, that there is an area (the triangle in between two adversaries) in which they cannot pretend that someone is present at position P. But we have also shown in our article that the chance that a verifier is in this area is so small that it does not give problems and therefore it is even with moving verifiers always possible to make your mother believe that you are studying.

Despite her efforts, you can always fool your mother into thinking you are studying at Science Park, as long as you and your friends together position yourselves correctly. For you this is a great thing. For Position-Based Cryptography, it means that no encryptions based on someones location is secure, for the location itself cannot be known with certainty.

Bibliography

Buhrman, H., Chandran, N., Fehr, S., Gelles, R., Goyal, V., Ostrovsky, R., and Schaffner, C. (2014). Position-based quantum cryptography: Impossibility and constructions. *Society for Industial and Applied Mathematics*, 43:150–178.

Capkun, S., Rasmussen, K. B., Cagal, M., and Srivastava, M. (2008). Secure location verification with hidden and mobile base stations. *IEEE TRANSACTIONS ON MOBILE COMPUTING*, 7:1–14.

Chandran, N., Goyal, V., Moriarty, R., and Ostrovsky, R. (2009). Position based cryptography. *Lecture Notes in Computer Science*, 5677:391–407.

Dziembowski, S. and Zdanowicz, M. (2014). Position-based cryptography from noisy channels. *Lecture Notes in Computer Science*, 8469:300–317.

Wootters, W. K. and Zurek, W. H. (1982). A single quantum cannot be cloned. *Nature*, 299:802–803.