Secure Localization Using *Dynamic Verifiers*

Nashad A. Safa, Saikat Sarkar, Reihaneh Safavi-Naini, and Majid Ghaderi

Department of Computer Science
University of Calgary
{nasafa,ssarka,rei,mghaderi}@ucalgary.ca

Abstract. We consider secure positioning in wireless environments where mobile nodes use a trusted infrastructure to prove their location: a node claims a position and wants to prove to the verification infrastructure that it is actually located in that position. We propose a system that uses the notion of *dynamic verifiers* and provides security against *collusion attack* in which the adversary corrupts a set of nodes and its aim is to claim a position where none of the corrupted nodes are located. We give a detailed analysis of the system and show that under reasonable assumptions the protocol will reject false claims and the success probability of the adversary can be made arbitrarily small. We also give the results of our simulation that closely match the analysis. Our protocol is the first secure positioning protocol with security against collusion attack.

1 Introduction

Finding location or range (distance from a fixed node) of nodes in wireless environments has been extensively studied [1,2,3,4,11,13]. These information can be used for network services such as routing [2,8,9], as well as for location based services. Location information has also been used for user authentication [7,20] and access control [21].

Positioning and ranging systems in wireless environments use attributes of wireless signals including time-of-flight, angle of arrival and signal strength [5]. In all cases there is a trusted infrastructure that verifies the claims of nodes (position or range). The infrastructure usually consists of a number of trusted verifiers with fixed, known locations, that have out-of-band communication with each other. Positioning and ranging systems however are vulnerable to malicious nodes that report false information. The strongest attack is when the adversary controls a number of nodes to make false claim (e.g. false location or range). This is called *collusion attack*. In [3] it was shown that an adversary always succeeds if the number of colluding nodes is sufficiently large. This impossibility result however is only valid if the location of verifiers is known to the adversary. A theoretically feasible result for this scenario is in *bounded storage* model that assumes that the adversary is restricted in terms of her storage capacity and can not store large messages broadcasted by verifiers.

A practically attractive solution for secure positioning systems is to use *hidden verifiers* [11]: that is to employ verifiers whose locations are not publicly

V. Atluri and C. Diaz (Eds.): ESORICS 2011, LNCS 6879, pp. 1–20, 2011.

known. This solution however becomes vulnerable if the adversary can adaptively interact with the system over a period of time. To our knowledge there is no positioning protocol that can withstand collusion attack.

Our Work: We consider the problem of secure location verification in the presence of collusion attack. We assume verifiers are connected by out-of-band communication channels and have fixed publicly known locations. Other nodes are mobile and their locations may vary with time. A node can use a positioning protocol to prove a location claim p to this infrastructure. The communication between nodes is wireless and the time-of-flight (TOF) of radio signals is used to calculate distances (the protocol however can be adapted to other localization techniques). We assume any node in the *verification region* can receive the transmission of all verifiers and other nodes. Colluding nodes coordinate their actions by communicating with each other using out-of-band channels and use directional antennas to target a specific receiver. The aim of the adversary is to claim a location that is "far" from all the corrupted nodes. Here "far" is determined by a threshold Δ, which is a system parameter and depends on the accuracy of the positioning system.

The proposed system: We propose a protocol, referred to as *Secure Localization using Dynamic Verifiers (or SLDV for short)* that, with a chosen high probability, will reject false location claims of provers. The protocol is efficient and practical and does not need any unrealistic assumption. Moreover it does not require any additional fixed infrastructure and is particularly applicable to cellular networks, mobile ad hoc networks or vehicular ad hoc networks where system devices have sufficient computing capability. The protocol requires a single public key operation per verification round from the device and so can be easily used for smart-phones, tablets and other similar handheld devices.

We introduce the notion of *dynamic verifiers*: these are verifiers that are 'recruited' in each run of the protocol. The set of dynamic verifiers is a random subset of users that is chosen from the active (connected) nodes in the system, and their role is to help the static verifiers to correctly verify a location claim. The subset is changed in each run of the protocol and so the locations of the dynamic verifiers remain unknown to the adversary. At a high level the protocol between the verifiers and a prover is a basic two phase challenge-response that measures the time-of-flight for the challenge and response, to determine the location. To avoid collusion attack, dynamic verifiers also 'monitor' and collect timing information on the challenge and response and provide it to the static verifiers. Since corrupted and honest nodes are indistinguishable to the verifiers, the set of dynamic verifiers may include corrupted nodes that would provide incorrect information to the verifiers. We will show that as long as sufficiently many dynamic verifiers are honest nodes, the final decision will be correct with high probability.

We give a detailed analysis of the protocol, and calculate the best chance of the adversary in two cases: (i) the adversary does not know the location of the mobile nodes in the system, and (ii) the adversary does know the location of mobile nodes in the system, but does not know which ones are chosen as dynamic verifiers. As long as the majority of nodes in the system are honest, the observations provided by the dynamic verifiers will guarantee a correct decision. In Section B, we give an argument similar to [3], to show that without this assumption it is impossible to provide secure positioning using dynamic verifiers approach.

The rest of the paper is organized as follows. We provide the model and background for our system in Section 3. Section 4 and 5 contain the protocol description and corresponding security analysis. Simulation results and efficiency of the protocol are discussed in Section 6, and Section 7 concluded the paper. An important advantage of dynamic verifier approach is that it does not need infrastructure cost. Correct working of the system requires sufficiently many "good" nodes (roughly more than half of the user population). To overcome this threshold we propose a *hybrid* approach that combines dynamic verifier approach with hidden base station approach [11] which is discussed in Appendix A.

2 Related Work

Protocols for location verification have been studied in the literature from theoretical and practical view points.

In [3], a theoretical framework for location verification was proposed and it was proved that secure positioning against collusion attack is impossible if the locations of the verifiers are public, and this result holds even if verifiers have unbounded computation. The authors then proposed a secure protocol in the Bounded Retrieval Model (BRM), where it was assumed that the verifiers can generate large strings with high minimum entropy that can not be stored by any entity in the system (including adversaries), but can be used by the prover. This work is mainly of theoretical interest.

Secure and practical positioning systems have been well-studied in the field of wireless security. These protocols can be broadly categorized into two groups: one in which the goal is to verify that the prover is within a certain distance from the verifiers, and the second in which a location claim must be verified. Protocols in [14],[12] belong to the first group. The protocol in [14] uses time-of-flight of radio signals while the Echo protocol in [12] uses time-of-flight of ultrasound signals. The protocols in [1,2,4,11,13] consider the problem of verifying location claims. In most cases multilateration (i.e. apply distance bounding strategy from a set of verifiers) based on the time-of-flight of radio signals is used. The 'two tests'-based scheme in [1] can detect false position claims from a single attacker, but is not secure against collusion attack. The system in [22] uses signal intensity of 802.11 card for accurate location determination, but does not consider collusion attack.

The protocols in [2,13] are able to protect against collusion attack but this is achieved using extra assumptions such as the existence of tamper-proof hardware or device fingerprints. The system in [4] provides security against collusion attack with the limitation that the number of colluders is less than the number of fixed verifiers (small number) in the system. In [11], a system with hidden and mobile base stations is considered. For better security, the number of hidden base stations must be high and this incurs significant infrastructure cost. In [3], it is shown that the locations of the hidden base stations can be found by an attacker, assuming that the attacker can run the protocol multiple times and know the result of a failed protocol run. A similar attack is presented in [3] for the case that the base stations are mobile.

In [23] hidden sensors are used to detect fake sensors. Fake sensors are not allowed to exchange messages and so the attack model is weaker than the standard collusion attack. The work of [25] is on geo-localization of wireless sensor nodes where a sensor finds its position from the nodes in its neighbourhood. On the contrary we consider location verification by trusted verifiers. In [24], location is verified by using the feedbacks of beacon nodes. Beacon nodes can be malicious. The authors show an upper bound on the number of malicious nodes that can be tolerated. A possible attack that is not considered by the authors is by positioning malicious nodes close to the honest nodes and so succeeding in false location claims. In comparison to the above previous works, we analyzed security of our protocol when user can provide malicious feedbacks and can position themselves distributively to provide timely responses without being at the claimed location.

3 Model and Background

3.1 Time-of-Flight for Positioning

We will use time-of-flight to determine distance between two nodes. This uses the round trip time of a sent message and the received response, assuming negligible delay in response by the receiver, and the constant speed of the wireless signal during this time. Accuracy of the TOF based schemes depends on the used technology. For example the system in [15] uses ultrasound and can achieve accuracy of 2 cm within the communication range of up to 100 meters. Implementation of [16] (using time-of-flight of radio signals as the underlying mechanism) has an average error of 1.17 m in outdoor environments whereas for indoor environments average error goes up to 2.1 m.

Attacks on the time-of-flight based localization systems aim at changing the timing information of the signals. In extreme cases, the signal is completely blocked (jammed) and the protocol fails. We do not consider these attacks here. In positioning systems with many nodes, the number of nodes that the adversary can corrupt and control (colluding nodes) determines its colluding power. The following Lemma is proved in [3].

Lemma 1. *If the number of colluders scales to the number of verifiers, they can devise a collusion attack to simulate any time-of-flight based location verification protocol to establish a fake position claim in a general setting.*

Readers are referred to [3] for the proof of this lemma.

3.2 Model

There is a set U of users (mobile nodes) that are connected to the system and their locations are registered to the positioning infrastructure. There is a fixed set of trusted static *verifiers*, $V = \{v_1, v_2, \ldots, v_g\}$ with publicly known locations. We refer to these as simply, *verifiers.* Verifiers have a pair (public and private) of encryption keys $((K_e, k_e))$, and a pair of public and private keys $((K_s, k_s))$ for a secure digital signature scheme. The public keys of the encryption and digital signature schemes are assumed to be known by all users. The corresponding secret keys are known only by the verifiers. Verifiers share all key information. Any user can send an encrypted message to the verifiers and can verify a message signed by them. Verifiers may also have out-of-band secure channel among themselves (possibly a wired connection). Data is shared among all the verifiers using this channel.

Verifiers maintain a table that records information about users that have appeared in the system so far. The first successful position verification of a node results in an entry in the table containing information such as pseudonym (Id) of the node, a shared key between the verifier and the node, the registered location of the node and an initialization vector. These information will be updated every time that the node can prove a location.

A *Prover* P_x, is a user who claims a new location p to be verified by V. A prover determines its location, e.g. using a GPS, and then transmits its claimed position p to the verifiers. Verifiers collaboratively execute a protocol with the prover to verify p. During the verification process for a position p, a subset of users, denoted by D_p, is selected, that will serve as the set of *Dynamic Verifiers* for that claim. Time is synchronized among the verifiers and the provers [19,18].

Security: The *Adversary* corrupts and controls one or more nodes with the aim of falsely claiming a position which is not "close" to any of the colluding nodes. Colluders are equipped with omnidirectional and directional antennas, and can use them to broadcast messages or target a particular node. We assume that communications between static verifiers and provers are of broadcast nature.

We assume that majority of nodes in the verification region are honest. We note that the number of static verifiers (e.g. base stations, satellites) that form the positioning infrastructure is in general much less than the number of users in a verification region. For example, in a typical cellular network, a cell-site usually contains from 1 to 3 base stations whereas the number of users may be in the range of hundreds of users.

We say *a location verification protocol Π is (Δ, λ) secure* if a prover at location p' can establish its fake location at p with probability $\leq \lambda$ such that $dist(p, p') > \Delta$. Probability is taken over all random coins of the adversary, and assuming uniform distribution for users in the verification region.

3.3 Notation

We used Π^{SLDV} to denote the localization protocol. DV denotes a dynamic verifier. P_x denotes a new prover who needs to be authenticated. We used the

Table 1. Notations used in Π^{SLDV}

Notation	Description
P_x	New prover
N	Number of users in the system
p_x	Position of user x
V	Set of static (i.e. system) verifiers
v_i	Static verifier i
DV_i	Dynamic verifier i
ID_i, ID_i^r	Id (pseudonym) for user i, and its refreshed form
IV_i	Initialization vector chosen by node i
$k_{V,i}$	Shared symmetric key between V and node i
K_e, k_e	V's Public and private key for public key encryption algorithm.
K_s, k_s	V's Public and private key for digital signature algorithm
$PubE(m, K_e), PubD(c, k_e)$	Public key encryption/decryption of plaintext m using key K_e/k_e.
$SymE(m, k), SymD(c, k)$	Symmetric Encryption/decryption of plaintext m using key k
$Sign(m, k_s), Verify(s, K_s)$	Digital sign/verify of message m using private key k_s/public key K_s
b_{DV_i}, b_{SV_i}	Status bit output of DV_i and v_i
q_+^{DV}, q_-^{DV}	Number of positive/negative verdicts of DV's responses
t_b	Broadcast instance of nonce x
c	Speed of light
$t_{r_j}^{SV}$	Time instance of response received by v_j
$t_{r_i}^{DV}$	Time instance of response received by DV_i

notation $dist(p, q)$ to indicate physical distance between positions p and q. We denote the acceptable error in location measurement by Δ. The notations used in the paper are summarized in Table 1.

4 The Proposed System

Consider a new prover P_x who wants to prove its location p to the system. We assume N users are connected to the network. Upon receiving a location claim message from P_x, verifiers execute a challenge response protocol and measure TOF to determine the location. However before sending the challenge, they increase the number of verifying nodes by selecting a set D_p of $k(\leq N)$ users to play the role of 'dynamic verifiers' in that round. The TOF measurement of static and dynamic verifiers, both will be used by the verifiers to verify the location of P_x. The adversary does not know the location of dynamic verifier nodes and so cannot position the corrupted nodes to control the view of the verifiers. The set D_p changes in every run of the protocol. Moreover the pseudonyms of the nodes, and hence DVs, are refreshed in every run. This ensures that the adversary cannot 'trace' nodes in multiple runs of the protocol. Algorithm 1 describes the main steps of Π^{SLDV}.

4.1 Algorithm Description

Verifiers maintain a user list $UserList$ with one entry for each user that has made a verified claim at one stage in the system. This table is updated after

Algorithm 1. Protocol Π^{SLDV}

```
1. Position claim
   $P_x$ :[1]
     $\rightarrow) : p$ ;   ; if first time prover.
     $\rightarrow) : SymE((p, ID_{P_x}), k_{V,P_x})$ ;   ; if returning prover[2].
2. Verifiers' challenge:
   $v_\ell \in V = \{v_1, v_2, \ldots, v_g\}$; select leader verifier.
   $v_\ell$ :
     $D_p = \{DV_1, DV_2 \ldots DV_k\} \subset UserList.$
     $ch \in_r \{0,1\}^n$;   challenge $ch$ is generated.
     $ID_i^r \leftarrow IDRefresh(ID_i)$, where $IDRefresh(ID_i) := (ID_i \oplus SymE(IV_i, k_{V,i}))$
   $v_\ell \rightarrow D_p : [ID_1^r, \ldots, ID_k^r]$
   $v_\ell \rightarrow) : ch,\ Sign(ch, k_s)$
3. Prover's response
   $P_x$ :
     $k_{V,P_x} \in_r \{0,1\}^{n'}$;   select symmetric key.
     $IV_{P_x} \in_r \{0,1\}^l$;   select initial vector.
     $\rightarrow) : m = (ch, PubE((k_{V,P_x}, IV_{P_x}), K_e))$
4. Dynamic verifiers' response
   $DV_i,\ i = 1, \cdots k$ :
     set $b_i^{DV} = 1$, if $ch$ in response $m$ matches with challenge; $b_i^{DV} = 0$ otherwise.
     record time instance of receiving $m$ ($t_{r_i}^{DV}$) and current position $p_{DV_i}$.
     $\rightarrow v_\ell : SymE((t_{r_i}^{DV}, p_{DV_i}, b_i^{DV}, ID_i^r), k_{V,DV_i})$.
5. Response validation and decision
   $v_\ell$:
   (i) Generate a vote for SV: increment $q_+^{SV}$ if $VerfyRp(t_b, t_{r_i}^{SV}, p, p_{v_i})$ returns
       True ; increment $q_-^{SV}$, if False.
   (ii) Generate a vote for DV: increment $q_+^{DV}$ if $VerfyRp(t_b, t_{r_i}^{DV}, p, p_{DV_i})$ returns
       True and $b_i^{DV}$=1 ; increment $q_-^{DV}$, if False.
   (iii) Accept $p$ if $q_+^{DV} > q_-^{DV}$ and $q_-^{SV} < \theta$; reject otherwise.
   (iv) Update user table: if first time prover: add new entry and send
       $SymE(ID_{P_x}, k_{V,P_x})$ ; if returning: update location.
```

each run of the protocol. The first successful verification of a user (P_x) results in an entry in $UserList$ that includes (i) current position of the user, (ii) a shared key k_{V,P_x}, (iii) an initial vector IV that will be used in symmetric cipher in a chaining mode and (iv) a pseudonym ID. k_{V,P_x} and IV are chosen by the prover and sent to V in encrypted form (public key encryption). ID is generated by V and is sent to the node in encrypted form (symmetric key encryption). ID is used as the user's pseudonym and is refreshed in every usage. The update of ID is done simultaneously by V and the corresponding user using IV for the symmetric key algorithm in a chaining mode. A returning prover with a new rejected claim will fail to establish its new position to verifiers.

[1] '$a : s$' indicates that node a executes statement s

[2] '$a \rightarrow) : m$' is used to indicate sending of a broadcast message m by node a. Similarly '$a \rightarrow b : m$' represents sending of message m from node a to node b.

Update(UserList) algorithm is run after each verification session. Let $V = \{v_1, v_2, \ldots, v_g\}$ denote the set of verifiers.

P_x Claims Location p. A user P_x wants to claim a location p. If this is the first time claiming a location, P_x simply broadcasts p to verifier set V. If it is a returning prover with ID_{P_x} and k_{V,P_x}, it also sends its ID_{P_x} along with p in encrypted form.

Verifiers' Challenge. Verifiers select a lead verifier v_ℓ who coordinates the communication and decision of that protocol run. v_ℓ randomly selects a subset of k users, denoted by D_p, from $UserList$ as dynamic verifiers for that round. v_ℓ refreshes the IDs of the selected subset using $IDRefresh(ID)$; selects a random challenge ch; and broadcasts the list of refreshed IDs and then the signed challenge.

Prover's Response. P_x receives the challenge and verifies the signature. If this is the first time verification (i.e. a new prover), it selects a random key k_{V,P_x} and an initial vector IV_{P_x}; encrypts the key and the IV using the public key of the verifiers[3] and broadcast the message $m = (ch, PubE((k_{V,P_x}, IV_{P_x}), K_e))$. If a returning prover, P_x only broadcasts $m = ch$.

Dynamic Verifiers' Response. A dynamic verifier DV_i is a user who has been verified before and shares a symmetric key k_{V,DV_i} and an IV_{DV_i} with the verifiers. DV_i maintains its current ID in synchrony with the verifiers, and so by monitoring verifiers' broadcast, will be able to determine if it has been selected as a dynamic verifier. It verifies the signature of challenge ch and records the time of receiving P_x's response, denoted by $t_{r_i}^{DV}$. It outputs a bit b_{DV_i} that is 1 if the replayed nonce by P_x is equal to the challenge nonce (ch), and zero otherwise. Each DV_i also determines its position p_{DV_i} using the GPS functionality.

Response of DV_i (i.e. $p_{DV_i}, t_{r_i}^{DV}, b_{DV_i}, ID_i^r$) is encrypted using the shared key with the verifiers using the symmetric key algorithm (used in a secure mode).

Response Validation and Decision. v_ℓ receives a vote $b_{SV_i}, i = 1 \ldots g$ from each static verifier. b_{SV_i} is generated using the time that v_ℓ broadcasts ch (t_b) and v_i receives P_x's response ($t_{r_i}^{SV}$), using algorithm $VerfyRp(t_b, t_{r_i}^{SV}, p, p_{v_i})$. These votes are securely sent to v_ℓ (out-of-band communication, or public key encryption).

v_ℓ also receives $(p_{DV_i}, t_{r_i}^{DV}, b_{DV_i}, ID_i^r), i = 1 \ldots k$. If b_{DV_i} is set to 1, v_ℓ uses algorithm $VerfyRp(t_b, t_{r_i}^{DV}, p, p_{DV_i})$ to determine if P_x's response was within the expected time for the respective DV_i. If not, vote of this DV_i is considered negative directly. The claim is accepted if for the majority of DVs, the response is valid, and less than a threshold θ of static verifiers reject the claim. The threshold θ is chosen according to quality of reception in the verification region and the required level of assurance.

[3] We comment that encryption should be completed before receiving the challenge to reduce processing delay.

Algorithm 2. Update($UserList$)

```
if position claim is accepted then
  if $P_x$ is a new prover then
    $v_\ell : Data_{P_x} \leftarrow \{IV_{P_x}, p, k^{V,P_x}\}$
    $UserList[ID_{P_x}] \leftarrow Data_{P_x}$
  else
    Update position of $P_x$
    in $UserList[ID_{P_x}]$ to $p$
  end if
end if
for every dynamic verifier $DV_i$ do
  Increase $IV_i$ in $UserList[ID_i]$ by 1
end for
```

Algorithm 3. $VerfyRp(t_1, t_2, p_1, p_2)$

```
if $|(t_2 - t_1)c - (dist(p_{v_\ell}, p_1) + dist(p_1, p_2))| \leq \varDelta$ then
  return True
else
  return False
end if
```

List Update. The next steps of the protocol relate to update $UserList$ after a verification scenario and are described in Algorithm 2. Let us assume $Data_{P_x}$ represents the collection of attributes (IV_{P_x}, p, k_{V,P_x}) associated with an accepted prover P_x. If P_x is verified for the first time, its attributes are included in $UserList$ table indexed by ID_{P_x}. Otherwise, its position in $UserList$ is updated to p. Regardless of acceptance or rejection, IV_i of each DV_i participated in this verification session are increased (or updated).

4.2 Reliability of Verification Decisions

Verifiers randomly select k users as dynamic verifiers from the set of all registered users in the network. We use a parameter h_p to show the level of trust that can be put on a randomly selected node and so to its observation. For k randomly selected users, the probability that at least $k/2$ users are dishonest is given by $\sum_{i=\lfloor k/2 \rfloor}^{k} \binom{k}{i}[(1-h_p)^i h_p^{k-i}]$ which is $\mathcal{O}(e^{-2(k-1)(h_p-0.5)^2})$ for $h_p \geq 0.5$. We can denote this probability as P_{dh}. The following Lemma shows that, if $h_p \leq 0.5$, it is impossible to provide a secure localization protocol in the general setting (aka 'vanilla model'). The proof of the lemma is given in Appendix B.

Lemma 2. *If the number of colluding nodes is at least half of the total number of users and verifiers (i.e.$(N+g)/2$), the dynamic verifier system will loose its security. That is colluding nodes can formulate a collusion attack that guarantees success to establish a fake position claim in a general setting.*

5 Security Analysis

An adversary will be successful if a location p can be successfully verified without any colluding node being present at that location (within accuracy parameter $\varDelta$). The best success chance of the adversary is when it initiates a verification session rather than tampering with an existing session. In this case the adversary completely controls the initial location claim and the prover's response to the verifiers' challenge, and will succeed if it can modify (or dominate) the vote of sufficiently many dynamic verifiers. The barriers for this success are two: the location of dynamic verifiers are unknown and so correct timing information cannot be generated, and DVs use secure encryption algorithms to construct their messages in a format that is expected by the verifiers.

We first discuss the cryptographic protection offered by the system, and then analyze the success probability of the adversary in providing correct timing information. The set of DVs is chosen independently for each verification round and so the success chance of finding the DV set is the same as random guessing. Note that refreshing the IDs of DVs in each verification round ensures that long term monitoring of the network cannot be used for tracing users and better guessing the location of DVs. (If IDs are fixed, a very powerful adversary may be able to activate its corrupted nodes that are right next to the selected DVs of a verification round, immediately after seeing the broadcasted IDs.) To construct the response of a DV, the adversary must know the secret key shared by the DV and the verifiers. This is because the messages sent by DVs contain fresh information for the current round (including the refreshed ID of the node) and so can not be reused or correctly guessed which will have a small success chance. Finally one can consider a man-in-the-middle attack where the adversary intercepts m from an honest prover and replaces it with her own chosen key, which needs to be performed within expected response-time for most DV_i. This requires guessing D_p or their locations and adjusting the instance of sending the response accordingly.

So, the viable option remains for the adversary is to control the timing of sending the response message by anticipating the locations of DV or the set D_p itself (when the adversary is able to track the current location of all users). The latter case can arise if the users are not sufficiently mobile and thus a persistent adversary can find all of their locations by observing communications to each other (if allowed in the system) and to dynamic verifiers. In the following we will assume that a colluding node will succeed if it is well positioned. That is we do not use the added layer of security provided by the cryptographic primitives used in the system. In other words we assume that corrupted nodes can construct well formed responses and their success chance is only limited by the unknown locations of DVs.

5.1 Security against Collusion Attacks

Theorem 1 states the security of Π^{SLDV} against collusion attack. Let h_p denote the probability that a randomly selected user be honest and $h'_p = 2h_p - 1$. We assume $2r$ is the diameter (largest dimension) of the verification region, and N denotes the total number of users in the system.

Theorem 1. *Π^{SLDV} is (Δ, λ) secure against collusion attack provided the number of corrupted nodes in the system is less than the number of honest nodes. The value of λ is as follows:*

1. *$\lambda = 1 - P_{SLDV} = P_{dh} + (1-P_{dh})(\frac{0.809}{r/\sqrt{N(1-h_p)}} \times 2\Delta)^{k/2})$, when users locations are unknown to adversaries;*
2. *$\lambda = 1 - P'_{SLDV} = P_{dhm} + (1-P_{dhm})(\frac{0.809}{r/\sqrt{N(1-h_p)}} \times 2\Delta)^{k/2})$, when users locations are available to adversaries.*

Here, $P_{dh} = \sum_{i=\lfloor k/2 \rfloor}^{k} \binom{k}{i}[(1-h_p)^i h_p^{k-i}]$, and $P_{dhm} = \sum_{i=\lfloor k/2 \rfloor}^{k} \binom{k}{i}[(1-h'_p)^i h_p'^{k-i}]$.

Location of Users Are Unknown. To deceive a DV_i in D_p (without knowing its location), the distance between a colluding node and DV_i must be the same as the corresponding distance between the claimed location and DV_i. The adversary succeeds if this is the case for majority of users in D_p. As DV are assumed to be randomly distributed in the network space, the adversary's only strategy would be to guess the approximate locations of D_p and try to adjust the instance of transmitting m based on this guess. The following lemma gives the security of Π^{SLDV} against collusion attack when users are in unknown positions with respect to the adversary.

Lemma 3. *If $N \geq k$ mobile users are available in the system, Π^{SLDV} can detect collusion attack in a two dimensional region with probability, $P_{SLDV} = 1-(P_{dh} + (1-P_{dh})(\frac{0.809}{r/\sqrt{N(1-h_p)}} \times 2\Delta)^{k/2})$, where $P_{dh} = \sum_{i=\lfloor k/2 \rfloor}^{k} \binom{k}{i}[(1-h_p)^i h_p^{k-i}]$ and $\Delta << r$.*

Proof: For simplicity we assume positioning of devices takes place in a circular region of radius r. Here we consider both two-dimensional (2D) and three-dimensional (3D) localization scenarios. Suppose an adversary chooses its position A_p uniformly over the region of radius r and the distance between a DV_i and the claimed prover's location is s. We first determine the success probability in the two-dimensional case. A similar argument can be used for three-dimensional case.

First, we compute the probability that the distance between a single adversarial node and a single DV_i is the same as the distance between the claimed location and that DV_i. Then we will extend the result for multiple colluding nodes and dynamic verifiers. Using the geometric approach of [17], the probability distribution function (pdf) of adversary's distance d_A from the randomly distributed location of a dynamic verifier can be expressed by:

$$Pr_{2D}[d_A = s] = \frac{4s}{\pi r^2} \cos^{-1}(\frac{s}{2r}) - \frac{2s^2}{\pi r^3}\sqrt{1 - \frac{s^2}{4r^2}}$$

And for $3D$,

$$Pr_{3D}[d_A = s] = \frac{3s^2}{r^3} - \frac{9s^3}{4r^4} + \frac{3s^5}{16r^6}$$

Plotting these functions for different values of $s, (0 \leq s \leq 2r)$ establishes that for $s = 0.84r$, $Pr_{2D}[d_A = s]$ achieves its maximum value (which is $0.809/r$) whereas for $s = 1.05r$, $Pr_{3D}[d_A = s]$ achieves its maximum value $0.942/r$. The maximum probability value decreases (which is always achieved at the same s

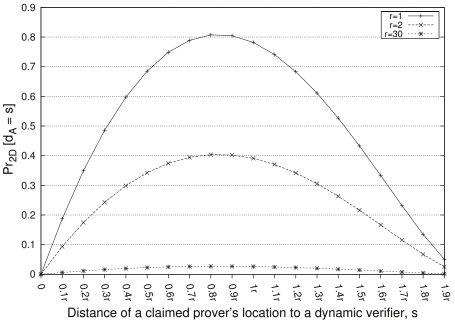


Fig. 1. $Pr_{2D}(d_A = s)$ vs r

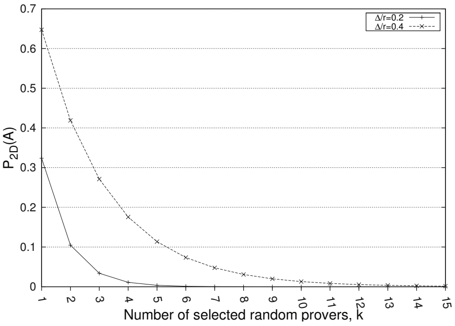


Fig. 2. Value of $Pr_{2D}(A, k)$ As k Increases

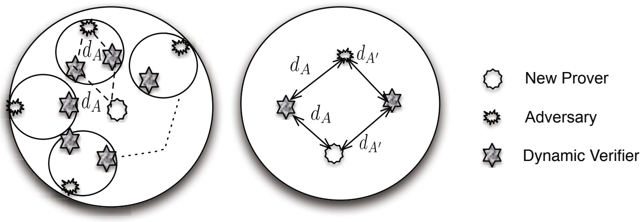


Fig. 3. Colluding Adversarial Scenario

value) as the radius of the network r increases. The relation is depicted in Fig 1 for the two dimensional case. This means that a malicious node has the highest probability of success when it guesses the distance of a random DV_i from its position as $s = 0.84r$ (in 2D case). As we are also allowing an error range Δ to be accepted (to compensate for processing and transmission delay), average success probability of an attacker for a network dimension of radius r can be computed as follows [11]:

$$Pr_{2D}(A) = \int_{0.84r-\Delta}^{0.84r+\Delta} Pr_{2D}[d_A = s] \approx \frac{0.809}{r} \times 2\Delta$$

$$Pr_{3D}(A) = \int_{1.05r-\Delta}^{1.05r+\Delta} Pr_{3D}[d_A = s] \approx \frac{0.942}{r} \times 2\Delta$$

Here, Δ has been considered with respect to radius r and $0 < \Delta < 0.5r$. The above success probabilities have been calculated considering a single DV_i and a lone adversary. We note that, as the radius of the verification region r increases, this probability decreases drastically. In practice, we expect r to be a large value. In Π^{SLDV}, verifiers select k independent random users in D_p for the purpose of response verification. For calculating probability of a successful attack which involves k random DV against a set of colluding nodes, we identify the following two cases (see Fig 3):

(1) A single adversarial node tries to deceive the k DV based on conjectures. In this case the probability of a successful attack would be the probability that distances between every DV_i in D_p and this sole adversary are same as the corresponding distances between these DV_i and the claimed location. So, the average success probability of attack in this scenario would become:

$$Pr_{2D}(A,k) = [Pr_{2D}(A)]^k \approx (\frac{0.809}{r} \times 2\Delta)^k \tag{1}$$

$$Pr_{3D}(A,k) = [Pr_{3D}(A)]^k \approx (\frac{0.942}{r} \times 2\Delta)^k \tag{2}$$

For reasonable small Δ and moderate values of k, success probability of adversarial spoofing of a fake position becomes negligible. Fig 2 shows how $Pr_{2D}(A,k)$ value decreases when we increase k for two different values of Δ.

(2) Alternatively we can assume l colluding nodes are collaborating for the deception. Now if a dynamic verifier receives responses from more than one malicious node who is sending response on behalf of the actual prover, it will invalidate the position claim. Thus, from adversarial view point it will be wise to divide the whole validation zone (of area πr^2) into l subregions where each colluder will broadcast the response using directional antennas or alternatively broadcast with low signal power. Members of D_p are selected uniformly over the verification region. The number of dynamic verifiers (assuming they are uniformly distributed and all the subregions possess equal area) in each of these subregions would be k/l. Without loss of generality we can assume that each subregion is a circle of radius r' where $r' \approx r/\sqrt{l}$ and $r' \geq 1$. Hence, the probability that colluder i would be successful to deceive his part of dynamic verifiers (k/l in number) is given by:

$$Pr_{2D}(A_{single}) = [Pr_{2D}(A)]^{k/l} \approx (\frac{0.809}{r'} \times 2\Delta)^{k/l}$$

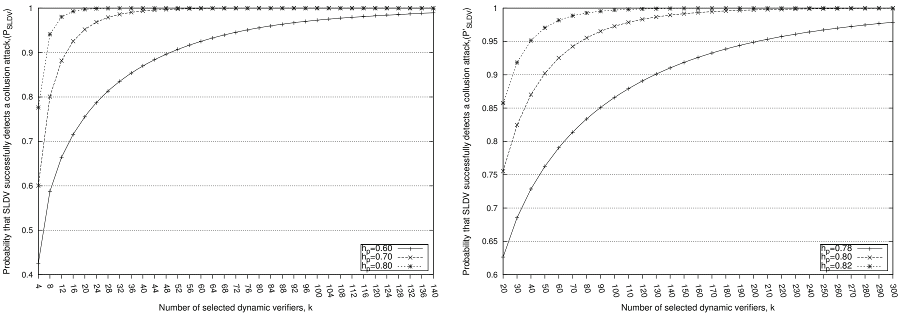
$$Pr_{3D}(A_{single}) = [Pr_{3D}(A)]^{k/l} \approx (\frac{0.942}{r'} \times 2\Delta)^{k/l}$$

Hence the probability that all the l colluders become successful in their attack would be:

$$Pr_{2D}(A_l,k) = [Pr_{2D}(A_{single})]^l \approx (\frac{0.809}{r'} \times 2\Delta)^k \tag{3}$$

$$Pr_{3D}(A_l,k) = [Pr_{3D}(A_{single})]^l \approx (\frac{0.942}{r'} \times 2\Delta)^k \tag{4}$$

Suppose h_p represents the probability with which a random user can be trusted. Now if we select k random users in D_p, the probability that $k/2$ dynamic verifiers or more would be dishonest is $\sum_{i=\lfloor k/2 \rfloor}^{k} \binom{k}{i}[(1-h_p)^i h_p^{k-i}]$. We can denote this probability as P_{dh}. If total number of nodes in the system is N, the number of expected malicious nodes would become $N(1-h_p)$. Therefore, r' in Eq. 3 will be



(a) P_{SLDV} vs k for different h_p when $\Delta/r = .01$ and $N = 1000$

(b) P'_{SLDV} vs k for different h_p when $\Delta/r = .01$ and $N = 1000$

Fig. 4. Probability of Detecting Collusion Attack

$r/\sqrt{N(1-h_p)}$. Consequently, Π^{SLDV} will be able to detect a collusion attack (in this scenario) with probability P_{SLDV}, where

$$P_{SLDV} = 1 - (P_{dh} + (1 - P_{dh})(\frac{0.809}{r/\sqrt{N(1-h_p)}} \times 2\Delta)^{k/2})$$

In typical networks, we expect the value of h_p to be high. It is evident in Fig. 4 that for reasonable values of k, probability of detecting a collusion attack becomes close to one and we claim that Π^{SLDV} is $(\Delta, 1 - P_{SLDV})$ secure against collusion attack under the stated assumptions. This completes the proof of our lemma. □

Location of All Users is Known to the Adversary. The following lemma specifies the security of Π^{SLDV} against collusion attack when users are in known positions with respect to the adversary.

Lemma 4. *If $N \geq k$ users are available in the system (whose locations are known to adversaries), Π^{SLDV} can detect collusion attack in a two dimensional region with probability, $P'_{SLDV} = 1 - (P_{dhm} + (1 - P_{dhm})(\frac{0.809}{r/\sqrt{N(1-h_p)}} \times 2\Delta)^{k/2})$ where $P_{dhm} = \sum_{i=\lfloor k/2 \rfloor}^{k} \binom{k}{i}[(1-h'_p)^i h'^{k-i}_p]$, $\Delta << r$ and $h'_p = 2h_p - 1$.*

Proof: With N users connected to the system and h_p trust assumption, the number of expected attackers is, $l = N(1 - h_p)$. If the size of set D_p is k, out of the k DV, $k(1 - h_p)$ are expected to be corrupted. Now l attackers can choose l of the users randomly, out of total Nh_p honest users, to take appropriate position and perform the distributed time-delayed response attack (as described in Lemma 1). Among these l users, $l\frac{kh_p}{Nh_p}$ of them are expected to be in the set D_p. Consequently, expected number of unmonitored honest DV in set D_p is, $k' = k - k(1 - h_p) - N(1 - h_p)\frac{k}{N}$.

Let h'_p represents the probability that a random DV in set D_p is both honest and unmonitored. Then, $h'_p = k'/k = 2h_p - 1$. From this expression, it is easy to find that for h'_p to be greater than 0.5 (for reliable decision), we need $h_p \geq 0.75$. Now using the same arguments presented in Lemma 3, it can be derived that probability of detecting a collusion attack by Π^{SLDV} is,

$$P'_{SLDV} = 1 - (P_{dhm} + (1 - P_{dhm})(\frac{0.809}{r/\sqrt{N(1-h_p)}} \times 2\Delta)^{k/2})$$

Where, $P_{dhm} = \sum_{i=\lfloor k/2 \rfloor}^{k} \binom{k}{i}[(1-h'_p)^i h'^{k-i}_p]$ and $h'_p = 2h_p - 1$. □

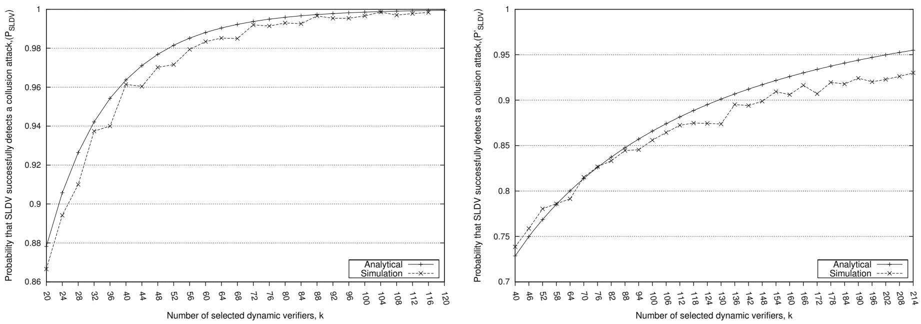
To calculate attacker's success probability in three dimensional regions, we just need to replace 0.809 with 0.942 in both P_{SLDV} and P'_{SLDV}.

We have numerically computed and plotted the probability of collusion attack in Fig 4. In Fig. 4(a) we can see that as long as the users' locations cannot be tracked by the adversary, for reasonable values of k, the probability of detecting a collusion attack, (P_{SLDV}) becomes close to one. In the case of all users' locations are known to the adversary, Fig. 4(b) shows the probability values of detecting a collusion attack (P'_{SLDV}) for different h_p assumptions and $\Delta/r = 0.01$. As expected, when users' locations are compromised to the adversary, we need higher trust assumption and larger number of k to detect a collusion attack with high probability. However, we emphasize that in a typical mobile environment, current locations of very few users will be compromised to the adversary.

6 Simulation Results

We performed simulations to estimate the value of P_{SLDV} for two cases: users' location is unknown, and users' location is known to the adversary. Our simulation program is written in Java and run on an intel 2.66 GHz core 2 duo processor. The number of user nodes, range of co-ordinates for location and trust assumption h_p are given as simulation parameters. To generate random numbers we used *Random* class provided in *java.lang.Math* package. We considered $h_p = 0.65$, i.e. on average 350 of the 1000 users are corrupted in each system run. Co-ordinates of the users are sampled randomly within the given topological range. Users are also randomly selected as honest or corrupted in accordance with the trust assumption h_p.

Let l denote the total number of corrupted nodes and k denotes the number of randomly chosen DVs, t of which are corrupted. For simplicity we assumed that static verifiers are all deceived by the attackers and hence verification decision only depends on the dynamic verifiers. A random position p in the given range is assumed to be the falsely claimed location. We found the number of honest users (denoted by s) among k, such that the Euclidean distance between them and p is approximately equal (with allowed error distance of 0.01) to the distance between them and one of the corrupted user's location. If the sum of s and t is less than $k/2$, we regard this run of the simulation as a successful detection of collusion attack, otherwise not.



(a) P_{SLDV} when $\varDelta/r = .01$, $h_p = 0.65$ and $N = 1000$

(b) P'_{SLDV} when $\varDelta/r = .01$, $h_p = 0.78$ and $N = 1000$

Fig. 5. Comparison of Analytical and Simulated values

For each value of k, we ran this simulation 5000 times and determined during how many times the system is able to detect the collusion attack. If m times out of the 5000 runs, the system is able to detect the attack, then P_{SLDV} is estimated as $\frac{m}{5000}$. Fig 5(a) shows the close match of our simulation results and the analytical results when the locations of users are unknown. For the case that the locations of the users are known to the adversary, we assumed $h_p = 0.78$. Fig. 5(b) compares the values of P'_{SLDV} for analytical and simulated cases. Again the results closely match.

Efficiency. The verification protocol has two rounds of communication for a new prover together with one broadcast message by verifiers intended for all "dynamic verifiers", and k response messages from the DVs. That is the overall communication cost of the protocol is $\mathcal{O}(k)$.

Maintaining $UserList$ requires storage equal to $\mathcal{O}(N)$, where N is the total number of verified nodes in the network. The value k determines the security of the system and can be chosen based on the required level of security and the trust assumption on the users (see Fig 4). The main computation required by the system is constructing messages that use efficient public and symmetric cryptographic primitives. This makes the protocol computationally very efficient.

7 Conclusion

We proposed a protocol for secure localization of mobile devices using the notion of dynamic verifiers, and proved its security against collusion attacks provided there are sufficient number of honest users active in the network. To our knowledge, this is the first protocol with security against large number of colluding nodes (more than the number of static verifiers) without making extra, and in many cases unrealistic, assumptions. A possible future work in this context would be to implement the protocol in a real wireless environment and evaluate the performance in different scenarios.

References

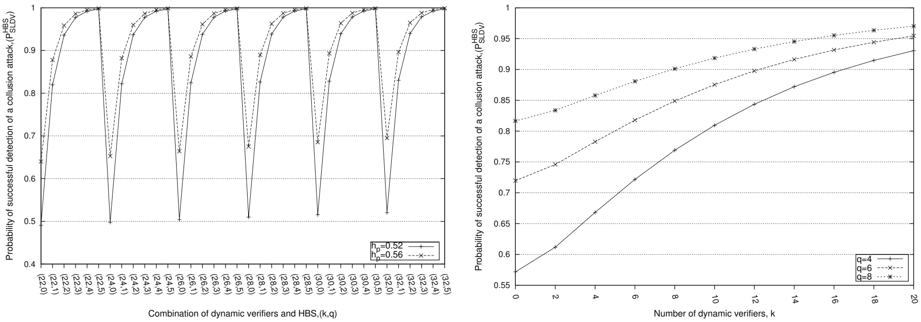
1. Čapkun, S., Hubaux, J.P.: Secure positioning of wireless devices with application to sensor networks. In: IEEE INFOCOM, Miami, USA (2005)
2. Čapkun, S., Hubaux, J.P.: Secure positioning in wireless networks. IEEE Journal on Selected Areas in Communications 24(2), 221–232 (2006)
3. Chandran, N., Goyal, V., Moriarty, R., Ostrovsky, R.: Position based cryptography. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 391–407. Springer, Heidelberg (2009)
4. Chiang, J.T., Haas, J.J., Hu, Y.-C.: Secure and precise location verification using distance bounding and simultaneous multilateration. In: WiSec, Zurich, Switzerland (2009)
5. Ferris, B., Haehnel, D., Fox, D.: Gaussian processes for signal strength-based location estimation. In: Robotics: Science and Systems, Philadelphia, USA (2006)
6. Jain, R., Puri, A., Sengupta, R.: Geographical routing using partial information for wireless ad-hoc networks. IEEE Personal Communications 8(1), 48–57 (2001)
7. Jansen, W., Korolev, V.: A location-based mechanism for mobile device security. In: World Congress on Computer Science and Information Engineering, Los Angeles, USA (2009)
8. Navas, J.C., Imielinski, T.: Geocast-geographic addressing and routing. In: MobiCom, Budapest, Hungary (1997)
9. Rasmussen, K.B., Čapkun, S.: Implications of radio fingerprinting on the security of sensor networks. In: SecureComm, Nice, France (2007)
10. Maurer, U.M.: Conditionally-perfect secrecy and a provably-secure randomized cipher. Journal Cryptology 5(1), 53–66 (1992)
11. Čapkun, S., Čagalj, M., Srivastava, M.: Secure localization with hidden and mobile base stations. In: INFOCOM, Barcelona, Spain (2006)
12. Sastry, N., Shankar, U., Wagner, D.: Secure verification of location claims. In: WiSE, Rome, Italy (2003)
13. Singelee, D., Preneel, B.: Location verification using secure distance bounding protocols. In: MASS, Washington, DC, USA (2005)
14. Brands, S., Chaum, D.: Distance-bounding protocols (extended abstract). In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 344–359. Springer, Heidelberg (1994)
15. Savvides, A., Han, C.-C., Srivastava, M.B.: Dynamic fine-grained localization in ad-hoc networks of sensors. In: Mobicom, Rome, Italy (2001)
16. Wibowo, S.B., Klepal, M., Pesch, D.: Time of Flight Ranging using Off-the-self IEEE802.11 WiFi Tags. In: POCA, Antwerp, Belgium (2009)
17. Tu, S.-J., Fischbach, E.: A New Geometric Probability Technique for an N-dimensional Sphere and Its Applications to Physics. arXiv:math-ph/0004021 (2000)
18. Elson, J., Estrin, D.: Time Synchronization for Wireless Sensor Networks. In: IPDPS, San Francisco, USA (2001)
19. Song, H., Zhu, S., Cao, G.: Attack-resilient Time Synchronization for Wireless Sensor Networks. In: MASS, Washington, DC, USA (2005)
20. Bardram, J.E., Kjær, R.E., Pedersen, M.Ø.: Context-aware user authentication – supporting proximity-based login in pervasive computing. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864, pp. 107–123. Springer, Heidelberg (2003)

21. Ardagna, C.A., Cremonini, M., Damiani, E., De, S., di Vimercati, C., Samarati, P.: Supporting Location-based Conditions in Access Control Policies. In: ASIACCS, Taipei, Taiwan (2006)
22. Traynor, P., Schiffman, J., La Porta, T., McDaniel, P., Ghosh, A.: Constructing Secure Localization Systems with Adjustable Granularity Using Commodity Hardware. In: IEEE GLOBECOM, Miami, USA (2010)
23. Delaët, S., Mandal, P.S., Rokicki, M.A., Tixeuil, S.: Deterministic Secure Positioning in Wireless Sensor Networks. In: IEEE DCOSS, Santorini Island, Greece (2008)
24. Jadliwala, M., Zhong, S., Upadhyaya, S., Qiao, C., Hubaux, J.: Secure Distance-Based Localization in the Presence of Cheating Beacon Nodes. IEEE Transactions on Mobile Computing 9(6), 810–823 (2010)
25. Garcia-Alfaro, J., Barbeau, M., Kranakis, E.: Secure Geolocalization of Wireless Sensor Nodes in the Presence of Misbehaving Anchor Nodes. Annals of Telecommunications (2011)

A Hybrid Approach with Hidden-Base Stations

In a Hybrid system the two approaches to secure positioning, dynamic verifier approach and hidden base station (HBS) approach, are combined to save on infrastructure, while providing protection against location tracking attack on these latter systems as proposed in [3]. In this hybrid scheme a set $HiddenV$ of fixed hidden verifiers work as part of the positioning infrastructure along with the set of dynamic verifiers. We provide a modification to the basic HBS approach of [11]. Instead of using all the static hidden verifiers from set $HiddenV$, we select a random subset S_{HV} (of size q) from it. Randomized selection of HBS ensures that the attack mentioned in [3] can not be executed as stated. This is because the outcome of the protocol now depends on the selection of HBS which resides in different locations at different executions. However, to achieve sufficient randomness for S_{HV}, the size of $HiddenV$ (denoted by z) needs to be much larger than q so that $\frac{1}{\binom{z}{q}}$ becomes small.

When HBS are used in conjunction with DV, z can be smaller as verification decision will also rely on D_p which is randomly selected from a large set of nodes. Suppose, according to [3], adversary needs to execute the localization protocol c (i.e. $\mathcal{O}(\log(1/\delta))$) times to find out the locations of HBS, where δ is the precision of location. To make the attack successful, selection of S_{HV} needs to be same for all these c rounds, which is proportional to $(\frac{1}{\binom{z}{q}})^{c-1}$. Moreover, observations of DV either can not have impact on these verifications or D_p needs to be static for these c rounds, which leads to a success chance proportional to $\left(\frac{1}{\binom{z}{q}}\left(P_{dh} + (1 - P_{dh})\frac{1}{\binom{N}{k}}\right)\right)^{c-1}$ for the adversary to determine locations of HBS. Thus by increasing z with respect to q, it is possible to achieve security in this hybrid approach even when h_p is less than 0.5 (i.e. P_{dh} is close to 1). Specifically, when $h_p \geq 0.5$ (i.e. P_{dh} is small), randomization of S_{HV} is not necessary and z can be equal to q. Now, if z hidden base stations are used (along with k dynamic verifiers) as part of infrastructure, the probability that any q



(a) Values of P_{SLDV}^{HBS} vs (k, q) for different h_p when $\varDelta/r = .01$ and $N = 1000$

(b) Values of P_{SLDV}^{HBS} vs k for different q when $\varDelta/r = .05$, $h_p = 0.7$ and $N = 1000$

Fig. 6. Values of P_{SLDV}^{HBS}

Table 2. Dynamic Verifiers vs Hidden Base Stations

	HBS-only System	Hybrid System	
$\varDelta/r$	# of HBS	# of HBS	# of DV
0.05	6	4	6
	8	4	11
	10	6	11
0.07	8	4	11
	10	4	15
	10	6	11

of them are deceived by colluding attackers is bounded by $(\frac{0.809}{r/\sqrt{N(1-h_p)}} \times 2\varDelta)^q$. Hence, the probability of detecting a collusion attack (i.e. P_{SLDV}^{HBS}) would become $1 - (P_{dh} + (1 - P_{dh})(\frac{0.809}{r/\sqrt{N(1-h_p)}} \times 2\varDelta)^{k/2})(\frac{0.809}{r/\sqrt{N(1-h_p)}} \times 2\varDelta)^q$, where $P_{dh} = \sum_{i=\lfloor k/2 \rfloor}^{k} \binom{k}{i}[(1-h_p)^i h_p^{k-i}]$. Here we assume z is appropriately chosen according to h_p. HBS are particularly useful when we have a comparatively low trust assumption for users. With $h_p = 0.52$ and 22 dynamic verifiers alone, we can achieve P_{SLDV} around 0.5 where as the combination of 22 dynamic verifiers and 5 HBS give P_{SLDV}^{HBS} of approximately 0.998 (see Fig. 6(a)).

Even if we consider that assumptions stated in the attack of [3] do not hold, this hybrid system can help to save on infrastructure cost required for setting up multiple HBS in the verification region. From Fig. 6(b) we can see that instead of using 8 HBS we can incorporate a combination of 4 HBS and 11 dynamic verifiers (with $h_p = 0.7$) to achieve similar security. Table 2 shows few comparisons to illustrate the equivalency between a HBS-only system and a hybrid system containing both DV and HBS (for different values of $\varDelta/r$ and $h_p = 0.7$).

B Impossibility of a Secure Protocol with Less Honest Users

Proof of Lemma 2. A valid user can lie about its location and share all of its cryptographic credentials with other collaborators. Moreover, a user can be honest for a certain period of time and then can attempt to establish a false location. Hence it's not possible for verifiers to determine whether a specific user is honest or dishonest before an authentication session with certainty. If the location verification protocol does not involve users to delegate verifying duty, we can devise a collusion attack similar to Lemma 1. Now, if the location verification protocol involves users to perform verifying duty and the number of colluding nodes scales up to them, we can form a collusion attack in any of the following ways:

1. In any selection of the users (D_p) to perform verifying duty, the number of expected malicious nodes will be as much as the honest nodes. Thus, taking a majority decision will not provide correct verification result regardless of the protocol. If unanimous decision is adopted for positive verification, it will always be possible for adversary to undermine a valid verification case.
2. If the network model allows users to communicate to each other which are not mobile, persistent adversary by observing communications of the users over a long period of time can determine their positions. Hence colluding nodes can take positions (with respect to the false location) in such a way that they can provide the correct response-time ($t^u_{r_i}$) for each user u. Consequently, regardless of which users are selected as DV, adversary becomes successful to establish the false location.

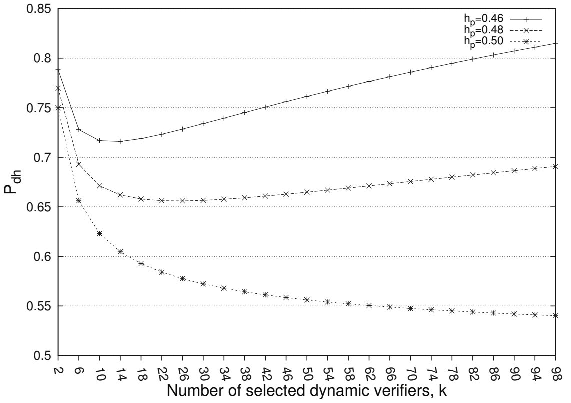


Fig. 7. Change in P_{dh} Value as k Increases

Figure 7 illustrates this fact by showing how reliability decreases (i.e. P_{dh} increases) when h_p is assumed to be ≤ 0.5 and we increase k indefinitely.