Revisiting Two-Hop Distance-Bounding Protocols: Are You Really Close Enough?

Nektaria Kaloudi, Aikaterini Mitrokotsa

Revisiting Two-hop Distance-Bounding Protocols: Are you really close enough?

Nektaria Kaloudi[1], and Aikaterini Mitrokotsa[2]

[1]University of the Aegean, Samos, Greece
n.kaloudi033@gmail.com
[2]Chalmers University of Technology, Gothenburg, Sweden
aikmitr@chalmers.se

Abstract. The emergence of ubiquitous computing has led to multiple heterogeneous devices with increased connectivity. In this communication paradigm everything is inter-connected and proximity-based authentication is an indispensable requirement in multiple applications including contactless payments and access control to restricted services/places. Distance-bounding (DB) protocols is the main approach employed to achieve accurate proximity-based authentication. Traditional distance-bounding requires that the prover and the verifier are in each other's communication range. Recently, Pagnin *et al.* have proposed a two-hop DB protocol that allows proximity-based authentication, when the prover and the verifier need to rely on an intermediate untrusted party (linker). In this paper, we investigate further the topic of two-hop distance-bounding. We analyse the security of the Pagnin *et al.* protocol for internal adversaries and we investigate the impact of the position of the linker in the distance-bounding process. We propose a new two-hop DB protocol that is more lightweight and avoids the identified problems. Finally, we extend the protocol to the multi-hop setting and we provide a detailed security analysis for internal adversaries.

keywords distance-bounding, authentication, relay attacks

1 Introduction

Ubiquitous computing technologies have affected radically our communications. Multiple heterogeneous devices are inter-connected and proximity-based authentication has been adopted in a wide range of applications *e.g.,* remote unlocking, contactless payments, proximity cards for access control in services/places. Distance-bounding (DB) protocols [1, 2] is a valuable tool in this ubiquitous computing paradigm, since they determine an upper bound on the physical distance between two communicating parties by measuring the round-trip-time between the exchanged challenges and responses. Distance-bounding protocols have received a

lot of attention in the literature and multiple works have been published focusing on the selection of optimal parameters for DB protocols [3–5], on related privacy issues [6, 7], as well as multiple attacks [8–11] against existing DB protocols and proposed solutions [12–14] that combat identified weaknesses. Furthermore, the concept of grouping proof distance-bounding protocols [15] has been recently introduced in order to provide not only a proof of the presence of multiple provers at the same time but also assurance regarding the physical proximity of the provers. However, in many cases (*e.g.,* communication in Vehicular ad-hoc Networks (VANETs) and Wireless Sensor Networks (WSNs)), we need to have a proof of proximity even when the two main parties prover and verifier are not in the direct communication range of each other, but have to rely on an untrusted intermediate node (linker). Recently, Pagnin *et al.* [16] proposed an extension of traditional DB protocols that can verify the proximity of both next-hop and two-hop neighbours.

In this article, we investigate the security of the Pagnin *et al.* [16] protocol for internal adversaries and we identify three weaknesses. We discuss how the location of intermediate untrusted linkers affects the DB process and we propose a new two-hop DB protocol that overcomes the identified problems, requires no computation from the linker, and provides lower attack success probabilities in comparison to the Pagnin *et al.* protocol. Finally, we investigate how the proposed two-hop DB protocol can be extended to the multi-hop setting [17] *i.e.,* when multiple intermediate linkers are used to reach the prover and we discuss how the attack success probabilities are affected.

2 Two-hop distance bounding

Traditional one-hop DB protocols consider two "legitimate" parties: a single verifier and a single prover within one-hop communication range. Pagnin *et al.* [16] introduced the concept of two-hop DB which involves three entities: a trusted verifier ($\mathcal{V}$), an untrusted prover ($\mathcal{P}$) and an untrusted in-between linker ($\mathcal{L}$). The goal is to bound the distance between $\mathcal{V}$ and $\mathcal{P}$, while relying on the intermediate untrusted linker $\mathcal{L}$. More precisely, $\mathcal{P}$ wants to be authenticated by $\mathcal{V}$, while $\mathcal{V}$ wants to bound the distance of $\mathcal{P}$. The prover $\mathcal{P}$ is not in the communication range of the verifier $\mathcal{V}$ (two-hop neighbours), while $\mathcal{L}$ is within the communication range (one-hop) of both $\mathcal{V}$ and $\mathcal{P}$. All nodes ($\mathcal{P}$, $\mathcal{V}$ and $\mathcal{L}$) use the same communication channel.

The two-hop DB protocol proposed by Pagnin *et al.* is depicted in Fig. 1 and employs two secret keys: $x_{\mathcal{VL}}$ shared between $\mathcal{V}$ and $\mathcal{L}$, and $x_{\mathcal{VP}}$ shared between $\mathcal{V}$ and $\mathcal{P}$. $N_{\mathcal{V}}$, $N_{\mathcal{P}}$ and $N_{\mathcal{L}}$ denote random nonces, $f_y()$ a pseudorandom function, $\mathsf{Enc}_y()$ an encryption function and $\mathsf{MAC}_y()$ a function that computes a message authentication code. All three functions use an appropriate key here denoted as y. More precisely, it is composed of the following phases:

–**Initialization phase:** $\mathcal{V}$, $\mathcal{L}$, $\mathcal{P}$ generate randomly their corresponding nonces. Both $\mathcal{V}$ and $\mathcal{L}$ use the pseudorandom function h and the secret key $x_{\mathcal{VL}}$ in order to calculate two n-bit sequences a_0 and a_1, while $\mathcal{P}$ uses the pseudorandom function h with the key $x_{\mathcal{VP}}$ to generate the d_0 and d_1 registers.

–**Distance-bounding phase:** In this phase, a fast bit exchange phase takes place, where $\mathcal{V}$ sends out one bit c_i to $\mathcal{L}$ and starts two clocks $t_{\mathcal{L}}$ and $t_{\mathcal{P}}$. $\mathcal{L}$ responds to both $\mathcal{V}$ and $\mathcal{P}$ with one calculated bit $\ell_i = (a_{c_i})_i$ and $\mathcal{V}$ stops the first timer $t_{\mathcal{L}}$. The delay time of the responses enables $\mathcal{V}$ to compute an upper-bound on the next-hop distance with $\mathcal{L}$ and after observing the second timer $t_{\mathcal{P}}$, it computes the other bound on the two-hop distance with $\mathcal{P}$. This phase is time-critical and is separated by n rounds of timed challenge/response exchanges.

–**Verification phase:** After $\mathcal{V}$ receives $\mathcal{P}$*'s* nonce, ℓ_i from $\mathcal{L}$ and the responses r_i, it computes the d_0, d_1 registers in order to verify r_i and ℓ_i. According to the result, $\mathcal{V}$ measures the bound on the physical distance to $\mathcal{V}$ using the clocks $t_{\mathcal{L}}$, $t_{\mathcal{P}}$.

2.1 Security Analysis

We describe what are the effects of a malicious linker $\mathcal{L}$ in the different phases of the Pagnin *et al.* protocol and analyse malicious behaviour that has not been considered before. We need to stress that the first two described security issues are related to weaknesses of the Pagnin *et al.* protocol to identify possible modifications of the transmitted messages that may subsequently lead to denial of service (DoS) attacks.

– *Malicious* $\mathcal{L}$ *in the initialisation phase:* A malicious $\mathcal{L}$ that receives $N_{\mathcal{V}}$ from $\mathcal{V}$, could send different nonces $N_{\mathcal{L}_1}$ and $N_{\mathcal{L}_2}$ to $\mathcal{V}$ and $\mathcal{P}$ correspondingly (Fig. 2), thus, disrupting the whole protocol and leading $\mathcal{P}$ and $\mathcal{V}$ to compute different registers $d_0 = f_{x_{\mathcal{VP}}}(N_{\mathcal{L}_2}, N_{\mathcal{P}})$ *i.e.,* computed by $\mathcal{P}$ in the initialisation, and $d'_0 = f_{x_{\mathcal{VP}}}(N_{\mathcal{L}_1}, N_{\mathcal{P}})$ *i.e.,* computed by $\mathcal{V}$ in the verification phase. Consequently, the computation will be completely different and as a result the protocol will fail.

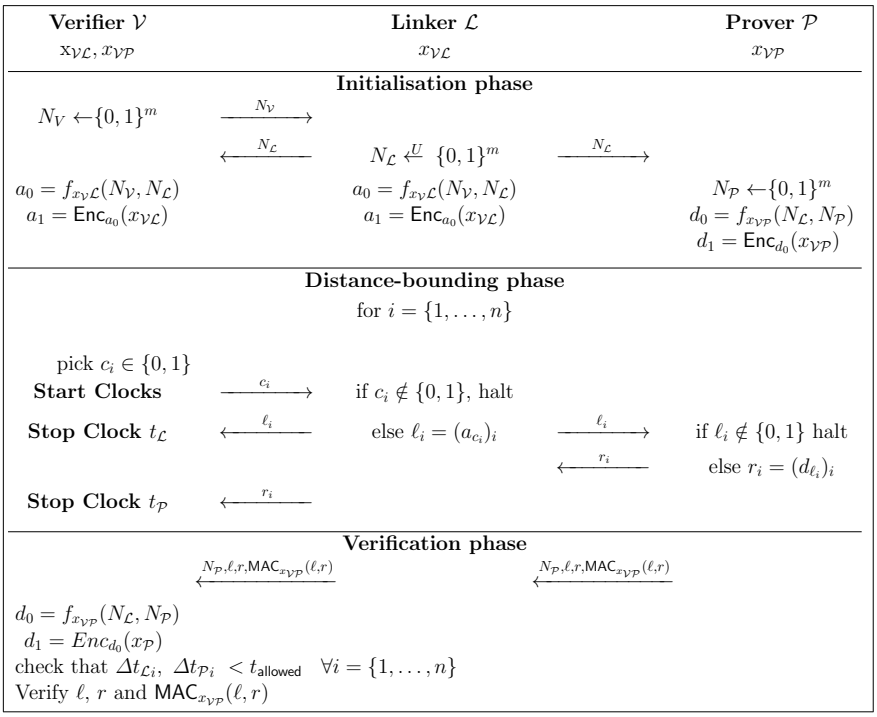

Fig. 1. The Pagnin *et al.* [16] two-hop distance-bounding protocol.

– *Malicious $\mathcal{L}$ in the verification phase:* Similarly to the previous attack, a malicious $\mathcal{L}$ could modify the transmitted $N_{\mathcal{P}}$, thus disrupting the computation of d_0 by $\mathcal{V}$ and consequently leading to the failure of the protocol. This can be easily detected if a MAC of the nonce $N_{\mathcal{P}}$ is also sent. Although this would not stop the DoS attack, it would definitely be useful to detect on time the misbehaviour of $\mathcal{L}$.

– *Malicious $\mathcal{L}$ in the distance-bounding phase:* A malicious $\mathcal{L}$ can act as a man-in-the-middle and perform the attack that was initially described by Kim *et al.* [18] against one-hop DB protocols [19]– against the Pagnin *et al.* [16] protocol in order to recover bits of the key $x_{\mathcal{VP}}$, when as Enc function is employed the one time pad *i.e.,* $d_1 = d_0 \oplus x_{\mathcal{VP}}$. More precisely, $\mathcal{L}$ can during the DB phase, toggle the value of one bit ℓ_i *i.e.,* $\ell_i' \neq \ell_i$ transmit the same ℓ_i' to $\mathcal{V}$ and $\mathcal{P}$ and leave all other messages untouched. Then, $\mathcal{L}$ can observe the verifier's reaction. If $\mathcal{V}$ accepts $\mathcal{P}$, it means that $\mathcal{P}$'s answer r_i was correct. Thus, the bit of the key $x_{\mathcal{VP}i}$ will be 0, because d_{0i}=d_{1i}. If $\mathcal{V}$ does not accept then $d_{0i} \neq d_{1i}$, thus, $x_{\mathcal{VP}i}$=1. We should note that in the Pagnin *et al.* protocol is stated: "$\mathcal{V}$ computes d_0 and d_1 and

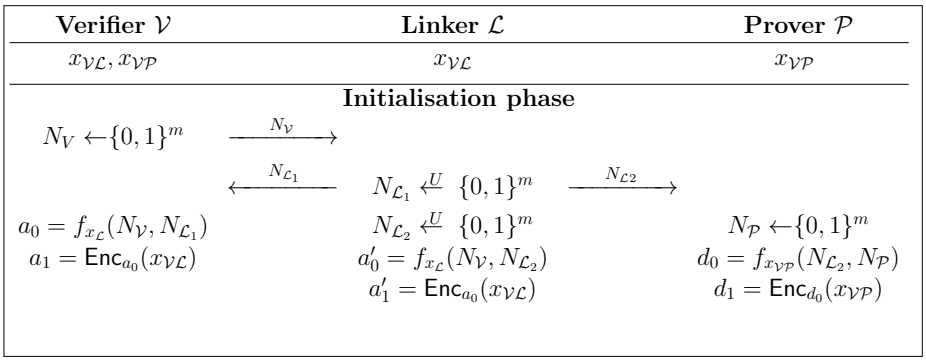

Fig. 2. Attack using different nonces.

verifies that all received ℓ_i and r_i, $\forall i \in \{1, ..., n\}$ are correct." However, this leaves rather unclear if $\mathcal{V}$ actually computes ℓ_i or simply verifies that the ℓ_is and r_is received during the distance-bounding phase match the ones received during the verification phase. To avoid this attack, the necessary condition is that $\mathcal{V}$ should recompute ℓ_i using $\ell_i = (a_{c_i})_i$ in the verification phase and verify the received ℓ_i's. If this recomputation is performed, then the attack will not be successful and the protocol will be aborted because $\mathcal{V}$ will see the differences in ℓ_i. However, if $\mathcal{V}$ simply verifies that the values of ℓ and r received in the DB phase match the ones received in the verification phase and the corresponding $\mathsf{MAC}_{x_{\mathcal{VP}}}(\ell, r)$ the attack can be performed successfully.

2.2 Effects of possible positions of the Linker

It is easy to see that the estimated distance between $\mathcal{V}$ and $\mathcal{P}$ in two-hop DB mainly depends on the position of $\mathcal{L}$. More precisely, $\mathcal{L}$ can be located either on the same line with the other two entities or anywhere else between them. In the second case, a triangle is formed. Let us denote by t_1 the estimated time required to transmit a message from $\mathcal{V}$ to $\mathcal{L}$ and t_2 the corresponding time required to transmit a message from $\mathcal{L}$ to $\mathcal{P}$. t_1 and t_2 can be easily estimated using the $\Delta t_{\mathcal{P}_i}$ and $\Delta t_{\mathcal{L}_i}$ in a two-hop DB protocol [16]. Let us denote by $d(A, B)$ the actual distance between two entities A and B.

If we construct a segment $d(\mathcal{V}, \mathcal{L})$ and a circle with centre $\mathcal{L}$ (Fig. 3, where $\mathcal{P}$ and $\mathcal{P}'$ denote possible locations of the prover), then $\mathcal{P}$ can be any point inside or on this circle. We would like to estimate the third side of the formed triangle $d(\mathcal{V}, \mathcal{P})$. If we knew the included angle between

$d(\mathcal{V}, \mathcal{L})$ and $d(\mathcal{L}, \mathcal{P})$, then we could determine the length of the third side. For instance, when the angle is equal to 180 degrees *i.e.,* $\mathcal{V}$, $\mathcal{P}$ and $\mathcal{L}$ are all in the diameter of the circle with centre $\mathcal{L}$ then $d(\mathcal{V}, \mathcal{P}) = d(\mathcal{V}, \mathcal{L}) + d(\mathcal{L}, \mathcal{P}) \approx c(t_1 + t_2)$ where c denotes the speed of light. Thus, in that case we have minimal error in estimating $d(\mathcal{V}, \mathcal{P})$ using a two-hop DB.

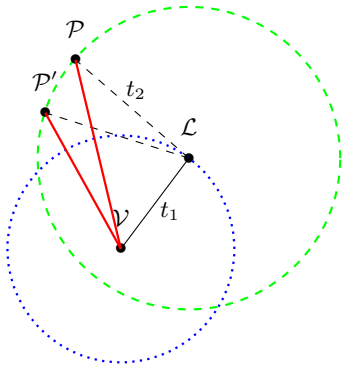

Fig. 3. Depiction of possible locations of a linker, a verifier, and a prover.

We may distinguish two cases in determining the estimation error ϵ on the physical distance between $\mathcal{V}$ and $\mathcal{P}$:

$\mathcal{L}$'s position is unknown Using the triangular inequality, we have an upper and lower bound for the distance: $c \mid t_1 - t_2 \mid -\epsilon \; < d(\mathcal{V}, \mathcal{P}) \; < c(t_1 + t_2) - \epsilon$ where ϵ denotes the error in the distance-bounding process. If we have multiple linkers $\mathcal{L}_j$, $j \in \{1, \ldots, m\}$ in the communication range of $\mathcal{V}$ and $\mathcal{P}$, we can run multiple times a two-hop DB protocol. We denote by $t_{j,1}$ the estimated time required to transmit a message from $\mathcal{V}$ to $\mathcal{L}_j$ and by $t_{j,2}$ the estimated time required to transmit a message from $\mathcal{P}$ to $\mathcal{L}_j$. By observing the different sums $(t_{j,1} + t_{j,2})$, we can deduce which linker is closer (*i.e.,* produces the smallest sum) and thus, which $\mathcal{L}_j$ has the smallest error in the distance estimation, but we cannot find its exact position.

$\mathcal{L}$'s position is known In this case, if we know the exact position of $\mathcal{V}$ and $\mathcal{L}$, we can find the position of $\mathcal{P}$ and consequently $d(\mathcal{V}, \mathcal{P})$. However, to achieve this with high accuracy, we need to have at least

three linkers in the communication range of $\mathcal{V}$ and $\mathcal{P}$. We may run a DB protocol three times each for a different linker $\mathcal{L}_j$, where $j \in \{1, 2, 3\}$ to get a good estimate of the three distances $d(\mathcal{L}_j, \mathcal{P})$. If we consider that $d_{\mathcal{L}_j\mathcal{P}}$ denotes the estimated distance via the DB protocol we can consider $d_{\mathcal{L}_j\mathcal{P}} \approx d(\mathcal{L}_j, \mathcal{P})$. Then, using trilateration [20] we can determine the exact location of $\mathcal{P}$. We can compute $d(\mathcal{V}, \mathcal{P}) = \sqrt{(x_\mathcal{V} - x)^2 + (y_\mathcal{V} - y)^2}$, where $x_\mathcal{V}$, $y_\mathcal{V}$ are the coordinates of $\mathcal{V}$ and x, y the coordinates of $\mathcal{P}$. If we know the angle between $d(\mathcal{V}, \mathcal{L}_j)$ and $d(\mathcal{L}_j, \mathcal{P})$ this computation is simplified, *i.e.,* if the triangle is orthogonal we only need to know the locations of two linkers. Alternatively, using the *Received Signal Strength Indicator* (RSSI) method [21] by employing multiple reference points and using the strength of the transmitted signals, we are able to estimate the distance between $\mathcal{V}$ and $\mathcal{P}$. After computing $d(\mathcal{V}, \mathcal{P})$, it is easy to see that the estimation error of the distance $d_{\mathcal{V}\mathcal{P}}$ computed via a two-hop DB protocol is $\epsilon = d_{\mathcal{V}\mathcal{P}} - d(\mathcal{V}, \mathcal{P})$.

Selection of the best Common Neighbour: In any case, the linker $\mathcal{L}$, should be a common neighbour of both $\mathcal{V}$ and $\mathcal{P}$. Several linkers may be located in the range of $\mathcal{V}$. For every linker $\mathcal{L}_j$ in the communication range δ of the verifier $\mathcal{V}$ it holds $d(\mathcal{V}, \mathcal{L}_j) \leqslant \delta$. Obviously, if in order to reach $\mathcal{P}$, the verifier $\mathcal{V}$ has to rely on multiple linkers (*e.g.,* $\mathcal{L}_1$ and $\mathcal{L}_2$), the error estimation will increase. Optimally, we want to choose the linker $\mathcal{L}_j$ that satisfies the condition $\max\{d(V, \mathcal{L}_j)\} \leq \delta$, while at the same time gives the lowest estimation error ϵ among the possible linkers.

3 The Proposed Two-Hop DB Protocol

In this section, we describe a novel two-hop DB protocol (depicted in Fig. 4), that overcomes the problems identified in Section 2.1, while requires no computation from the intermediate linker *i.e.,* $\mathcal{L}$ simply relays messages between $\mathcal{V}$ and $\mathcal{P}$. Furthermore, the proposed protocol as we will show in the security analysis presents higher resistance to the attacks considered by Pagnin *et al.* [16] for internal adversaries.

We consider the same setting of a verifier $\mathcal{V}$, an untrusted linker $\mathcal{L}$, and an untrusted prover $\mathcal{P}$ that wants to be authenticated and prove its proximity. We also consider that there is only one secret key $x_{\mathcal{V}\mathcal{P}}$ that is shared between $\mathcal{V}$ and $\mathcal{P}$. In the *initialization phase*, we would like to be sure that $\mathcal{L}$ transfers the correct nonces. A simple solution is that each node uses the shared key to compute a MAC on the nonce it sends. As a result, $\mathcal{P}$ and $\mathcal{V}$ can understand if the received nonce from $\mathcal{L}$ has been altered or not. Subsequently, $\mathcal{V}$ and $\mathcal{P}$ compute the values a_0 and a_1 using

a pseudorandom function and an encryption function correspondingly. In the *distance-bounding phase*, $\mathcal{V}$ chooses a random challenge bit, which is forwarded by $\mathcal{L}$ to $\mathcal{P}$, while $\mathcal{P}$ computes and sends back the responses $r_i = a(c_i)_i \; \forall i \in \{1 \ldots, n\}$. Finally, in the *verification phase*, $\mathcal{P}$ sends c, r and the corresponding $\mathsf{MAC}_{x_{\mathcal{V},\mathcal{P}}}(c, r)$ that is forwarded by $\mathcal{L}$ and finally verified by $\mathcal{V}$.

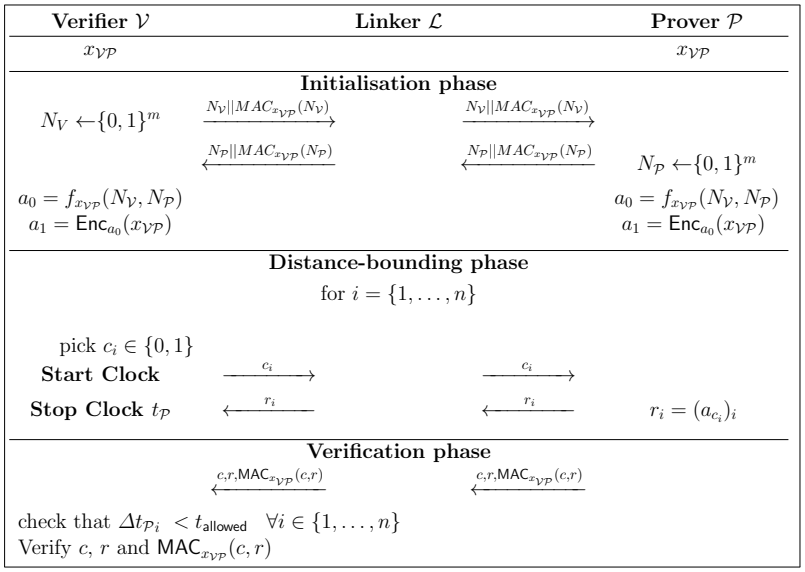

Fig. 4. Efficient two-hop distance-bounding protocol.

3.1 Security Analysis

In this section, we analyse the security of the proposed two-hop DB protocol, considering internal adversaries as in [16].

Case A - Dishonest prover $\tilde{\mathcal{P}}$, honest linker $\mathcal{L}$: A dishonest prover $\tilde{\mathcal{P}}$ might be located far away from $\mathcal{L}$ and may want to appear closer. So, in the distance-bounding phase $\tilde{\mathcal{P}}$ has to send the wrong response $\tilde{r}_i$ before it receives the challenge c_i from $\mathcal{L}$. Since r_i is determined by the two response registers a_0 and a_1, $\tilde{\mathcal{P}}$ knows r_i if $a_{0i} = a_{1i}$. If $a_{0i} \neq a_{1i}$ then $\tilde{\mathcal{P}}$ has to guess the response r_i. Thus, the success probability is $\left(\frac{3}{4}\right)^n$.

Case B - Honest prover $\mathcal{P}$, dishonest linker $\tilde{\mathcal{L}}$: $\tilde{\mathcal{L}}$ may want to shorten the distance between $\mathcal{P}$ and $\mathcal{V}$. We may consider two main strate-

gies: In the *first* strategy, $\tilde{\mathcal{L}}$ waits for c_i from $\mathcal{V}$ and sends it to $\mathcal{P}$. Then, $\tilde{\mathcal{L}}$ sends a random response before receiving r_i from $\mathcal{P}$. Since r_i is determined by a_0, a_1, and c_i, the success probability is equal to $\left(\frac{1}{2}\right)$ per round. In the *second* strategy, $\tilde{\mathcal{L}}$ sends a random challenge before receiving c_i from $\mathcal{V}$. Then, $\tilde{\mathcal{L}}$ waits for the response from $\mathcal{P}$ and forwards it to $\mathcal{V}$ when it sends c_i. The success probability is again equal to $\left(\frac{1}{2}\right)$ per round. Thus, the *overall* success probability is $\left(\frac{1}{2}\right)^n$.

We need to note here that the problems identified in Section 2.1, when $\mathcal{L}$ is malicious do not apply in the new protocol. By computing the MAC of the nonces in the initialisation phase, $\mathcal{L}$ cannot modify the nonces without being detected, while there is no need to transfer $N_{\mathcal{P}}$ in the verification phase. Also the Kim *et al.* [18] attack cannot be applied since a MAC of all transmitted c_i's and r_i's is verified at the end.

Case C - Dishonest prover $\tilde{\mathcal{P}}$, dishonest linker $\tilde{\mathcal{L}}$: We may discriminate into two sub-cases.

– *$\tilde{\mathcal{P}}$ and $\tilde{\mathcal{L}}$ do not collaborate:* In this sub-case, the success probability depends on whether Case A or Case B succeeds. More precisely, the success probability depends on whether $\tilde{\mathcal{P}}$ can guess r_i correctly with probability $\left(\frac{3}{4}\right)^n$ (Case A). For a dishonest $\tilde{\mathcal{L}}$ (Case B), the success depends on guessing c_i and r_i correctly (*i.e.,* $(\frac{1}{2})^n$). Thus, the overall success probability is $\left(\frac{5}{8}\right)^n$, (*i.e.,* $\left(\frac{3}{4}\right)$ for Case A or $\left(\frac{1}{2}\right)$ for Case B).

– *$\tilde{\mathcal{P}}$ and $\tilde{\mathcal{L}}$ collaborate:* In this sub-case, $\tilde{\mathcal{P}}$ collaborates with $\tilde{\mathcal{L}}$ in order to appear within the allowed distance bound. $\tilde{\mathcal{P}}$ has two options: to reveal some information about his secret key $x_{\mathcal{VP}}$ (something he does not want to do) or to send one register a_0 or a_1 to $\tilde{\mathcal{L}}$. Thus, in the latter case $\tilde{\mathcal{L}}$ can compute the half of responses r_i correctly and send them in time to $\mathcal{V}$. The other half, have to be guessed by $\tilde{\mathcal{L}}$. So, the overall success probability of the attack in this sub-case is $\left(\frac{3}{4}\right)^n$.

We should point out that in the above security analysis the focus is mainly on distance-shortening attacks since DB protocols are mainly employed in proximity-based authentication settings. In case that a malicious linker on purpose delays to relay information between $\mathcal{P}$ and $\mathcal{V}$ this would lead to failure of the protocol when the condition $\Delta t_{\mathcal{P}_i} < t_{\mathsf{allowed}}$ does not hold.

3.2 Extension to the multi-hop DB setting

Up to now, we have considered two-hop DB protocols where $\mathcal{P}$ and $\mathcal{V}$ need to rely on a single intermediate linker. In this section, we investigate how the proposed two-hop DB protocol can be extended to a multi-hop

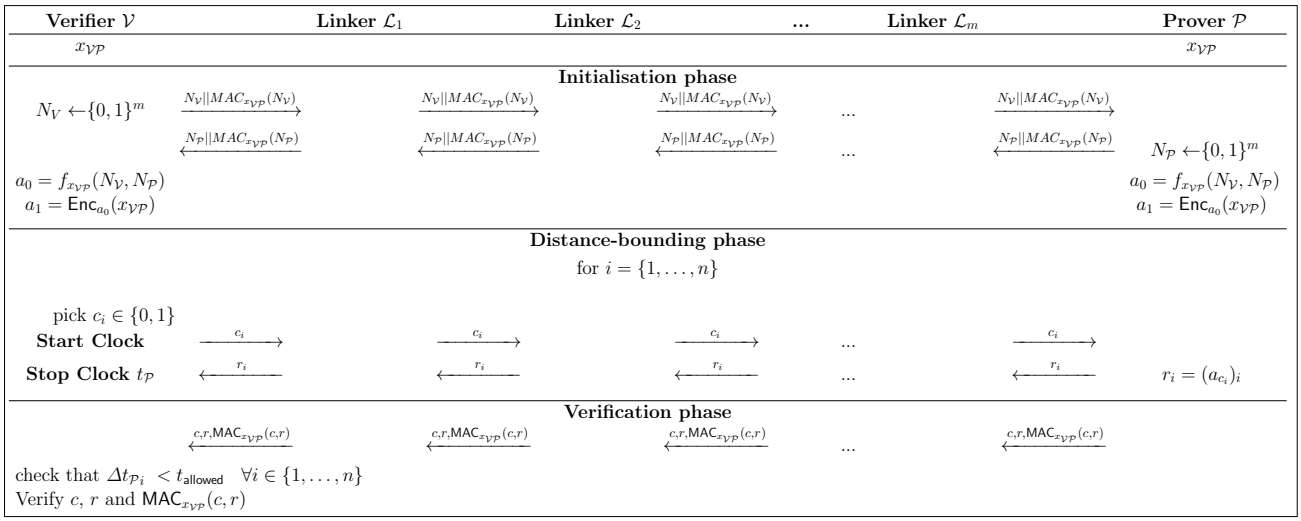

Fig. 5. Extended multi-hop distance-bounding protocol

setting *i.e.,* when $\mathcal{V}$ and $\mathcal{P}$ have to rely on multiple intermediate linkers $\mathcal{L}_j$ where $j \in \{i, \ldots, m\}$. Multi-hop distance estimation as a generalisation of distance-bounding was proposed for the first time by Mitrokotsa *et al.* [17]. In our proposed multi-hop DB protocol, similarly as before, the goal of $\mathcal{V}$ is to determine an upper-bound of the distance to $\mathcal{P}$ and the role of intermediates $\mathcal{L}_j$ is to forward the messages to $\mathcal{P}$. Our proposed multi-hop DB protocol (depicted in Fig. 5) is a natural extension of the proposed two-hop DB protocol; we discuss briefly its security in this section. It is easy to see that the protocol is similar to the proposed two-hop DB protocol.

The security analysis of the multi-hop DB protocol is similar to the one for the two-hop DB protocol.

Case A - Dishonest prover $\tilde{\mathcal{P}}$, honest linkers $\mathcal{L}$'s: The overall success probability remains $\left(\frac{3}{4}\right)^n$ since only $\mathcal{P}$ is dishonest.

Case B - Honest prover $\mathcal{P}$, dishonest linkers $\tilde{\mathcal{L}}$'s: When we have one dishonest linker the success probability is $(\frac{1}{2})^n$ as we have shown. It is easy to see that even if we have k dishonest linkers the overall success probability is again $(\frac{1}{2})^n$ since there is no dependency on the forwarded messages.

Case C - Dishonest prover $\tilde{\mathcal{P}}$, dishonest linkers $\tilde{\mathcal{L}}$'s: We have two sub-cases:

– *$\tilde{\mathcal{P}}$ and $\tilde{\mathcal{L}}$ do not collaborate:* This sub-case depends on the Cases A and

B. Either $\tilde{\mathcal{P}}$ (Case A) will succeed with probability $\left(\frac{3}{4}\right)$ or k $\tilde{\mathcal{L}}$'s (Case B) will succeed with probability $\left(\frac{1}{2}\right)$. Thus, the overall success probability is $\left(\frac{5}{8}\right)^n$.

– $\tilde{\mathcal{P}}$ *and* $\tilde{\mathcal{L}}$*'s collaborate:* If $\tilde{\mathcal{P}}$ collaborates with k $\tilde{\mathcal{L}}$'s *i.e.,* reveals either a_0 or a_1 (thus, half of the responses) then, the overall success probability will be $\left(\frac{3}{4}\right)^n$. Either $\tilde{\mathcal{P}}$ (Case A) will succeed with probability $\left(\frac{3}{4}\right)$ or k $\tilde{\mathcal{L}}$'s (Case B) will succeed with probability $\left(\frac{1}{2}\right)$. Thus, the overall success probability is $\left(\frac{5}{8}\right)^n$.

Table 1 summarises the best case attack success probabilities for the three protocols that we studied in this paper. It covers all the cases of malicious internal participants.

Table 1. Best Case Attack Success Probabilities

Protocol	$\tilde{\mathcal{P}}$,$\mathcal{L}$	$\mathcal{P}$,$\tilde{\mathcal{L}}$	$\tilde{\mathcal{P}}$,$\tilde{\mathcal{L}}$ collaboration	$\tilde{\mathcal{P}}$,$\tilde{\mathcal{L}}$ no collaboration
two-hop DB	$\left(\frac{3}{4}\right)^n$	$\left(\frac{3}{4}\right)^n$	$\left(\frac{3}{4}\right)^n$	$\left(\frac{3}{4}\right)^n$
new two-hop DB	$\left(\frac{3}{4}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{3}{4}\right)^n$	$\left(\frac{5}{8}\right)^n$
multi-hop DB	$\left(\frac{3}{4}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{3}{4}\right)^n$	$\left(\frac{5}{8}\right)^n$

4 Conclusion

In this paper, we investigated the problem of two-hop DB protocols. More precisely, we examined the security of the first two-hop DB protocol [16] and we identified some weaknesses. We also discussed how the location of the linker may affect the distance-bounding process. Furthermore, we proposed a novel two-hop DB protocol, that does not require any computation from the intermediate linker, does not suffer from the identified weaknesses and reduces the attack success probabilities for internal adversaries. Finally, we investigated how the proposed two-hop DB protocol can be extended to the multi-hop DB setting and how the attack success probabilities are affected.

5 Acknowledgements

This work was partially supported by the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no 608743, the VR grant

"PRECIS: Privacy and Security in Wearable Computing Devices" no 621-2014-4845, the STINT grant "Secure, Private & Efficient Healthcare with wearable computing no IB2015-6001 and the ERASMUS+HE2015 project.

References

1. Dimitrakakis, C., Mitrokotsa, A.: Distance-bounding protocols: Are you close enough? IEEE Security & Privacy **13**(4) (2015) 47–51
2. Mitrokotsa, A.: Authentication in constrained settings. In: Cryptography and Information Security in the Balkans - First International Conference, BalkanCryptSec 2014, Istanbul, Turkey, October 16-17, 2014, Revised Selected Papers. (2014) 3–12
3. Dimitrakakis, C., Mitrokotsa, A., Vaudenay, S.: Expected loss bounds for authentication in constrained channels, Orlando, Florida (March 2012)
4. Dimitrakakis, C., Mitrokotsa, A., Vaudenay, S.: Expected loss analysis for authentication in constrained channels. Journal of Computer Security **23**(3) (2015) 309–329
5. Mitrokotsa, A., Peris-Lopez, P., Dimitrakakis, C., Vaudenay, S.: On selecting the nonce length in distance-bounding protocols. The Computer Journal (2013)
6. Mitrokotsa, A., Onete, C., Vaudenay, S.: Location leakage in distance bounding: Why location privacy does not work. Computers & Security **45** (2014) 199–209
7. Aumasson, J.P., Mitrokotsa, A., Peris-Lopez, P.: A note on a privacy-preserving distance-bounding protocol. In: Proceedings of the 13th International Conference on Information and Communications Security (ICICS 2011). LNCS, Beijing, China (November 2011) 78–92
8. Pagnin, E., Yang, A., Hu, Q., Hancke, G., Mitrokotsa, A.: : Distance bounding meets human based authentication. Future Generation Computer Systems (2016) –
9. Mitrokotsa, A., Dimitrakakis, C., Peris-Lopez, P., Castro, J.C.H.: Reid et al.'s distance bounding protocol and mafia fraud attacks over noisy channels. IEEE Communications Letters **14**(2) (February 2010) 121–123
10. Bay, A., Boureanu, I., Mitrokotsa, A., Spulber, I., Vaudenay, S.: The bussard-bagga and other distance bounding protocols under man-in-the-middle attacks. In: Proceedings of Inscrypt'2012, 8th China International Conference on Information Security and Cryptology. Lecture Notes in Computer Science, Beijing, China, Springer (2012)
11. Mitrokotsa, A., Onete, C., Vaudenay, S.: Mafia fraud attack against the rc distance-bounding protocol. In: Proceedings of the 2012 IEEE RFID Technology and Applications (IEEE RFID T-A), Nice, France, IEEE Press (November 2012) 74–79
12. Pagnin, E., Yang, A., Hancke, G.P., Mitrokotsa, A.: Hb+db, mitigating man-in-the-middle attacks against HB+ with distance bounding. In: Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, New York, NY, USA, June 22-26, 2015. (2015) 3:1–3:6
13. Boureanu, I., Mitrokotsa, A., Vaudenay, S.: Practical and provably secure distance-bounding. Journal of Computer Security **23**(2) (2015) 229–257
14. Boureanu, I., Mitrokotsa, A., Vaudenay, S.: Practical and provably secure distance-bounding. In: Proceedings of the 16th Information Security Conference (ISC), Dallas, Texas, USA (November 2013)

15. Karlsson, C., Mitrokotsa, A.: Grouping-proof-distance-bounding protocols: Keep all your friends close. IEEE Communications Letters **20**(7) (July 2016) 1365–1368
16. Pagnin, E., Hancke, G., Mitrokotsa, A.: Using distance-bounding protocols to securely verify the proximity of two-hop neighbours. Communications Letters, IEEE **19**(7) (2015) 1173–1176
17. Mitrokotsa, A., Onete, C., Pagnin, E., Perera, M.: Multi-hop distance estimation: How far are you? Cryptology ePrint Archive, Report 2017/705 (2017) `http://eprint.iacr.org/2017/705`.
18. Kim, C.H., Avoine, G., Koeune, F., Standaert, F.X., Pereira, O.: The swiss-knife RFID distance bounding protocol. In: Information Security and Cryptology–ICISC 2008. Springer (2008) 98–115
19. Tu, Y.J., Piramuthu, S.: RFID Distance Bounding Protocols. In: Proc. of 1st Intern. EURASIP Workshop on RFID Technology. (2007)
20. Shih, C.Y., Marrón, P.J.: Cola: Complexity-reduced trilateration approach for 3d localization in wireless sensor networks. In: Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on. (July 2010) 24–32
21. Papamanthou, C., Preparata, F.P., Tamassia, R.: Algorithms for location estimation based on rssi sampling. In: Proceedings of Algorithmic Aspects of Wireless Sensor Networks, Springer-Verlag (2008) 72–86