On the optimal resistance against mafia and distance fraud in distance-bounding protocols*

Reynaldo Gil-Pons[a], Sjouke Mauw[a], Rolando Trujillo-Rasua[b]

[a] *University of Luxembourg, 6 Av. de la Fonte, 4364, Belval, Luxembourg*
[b] *Universitat Rovira i Virgili, 26 Campus Sescelades, 43007, Tarragona, Spain*

Abstract

Distance-bounding protocols are security protocols with a time measurement phase used to detect relay attacks, whose security is typically measured against mafia-fraud and distance-fraud attacks. A prominent subclass of distance-bounding protocols, known as *lookup-based protocols*, use simple lookup operations to diminish the impact of the computation time in the distance calculation. Independent results have found theoretical lower bounds $\frac{1}{2^n}\left(\frac{n}{2}+1\right)$ and $\frac{1}{2^n}$, where n is the number of time measurement rounds, on the security of lookup-based protocols against mafia and distance-fraud attacks, respectively. However, it is still an open question whether there exists a protocol achieving both security bounds. This article closes this question in two ways. First, we prove that the two lower bounds are mutually exclusive, meaning that there does not exist a lookup-based protocol that provides optimal protection against both types of attacks. Second, we provide a lookup-based protocol that approximates those bounds by a small constant factor. Our experiments show that, restricted to a memory size that linearly grows with n, our protocol offers strictly better security than previous lookup-based protocols against both types of fraud.

Keywords: Distance bounding, Mafia fraud, Distance fraud, Security

1. Introduction

Distance-bounding protocols aim to counteract relay attacks against wireless authentication protocols by ensuring proximity between communicating parties. Because the speed of light represents a hard limit on the speed of radio waves, the distance from a transmitter to a receiver is (at most) half the round-trip time multiplied by the speed of light, where round-trip time is the time it takes for a signal to travel from the transmitter to the receiver and back. This physical

*This document is the results of the research project funded by the Luxembourg National Research Fund, Luxembourg, under the grant AFR-PhD-14565947.

Email addresses: reynaldo.gilpons@uni.lu (Reynaldo Gil-Pons), sjouke.mauw@uni.lu (Sjouke Mauw), rolando.trujillo@urv.cat (Rolando Trujillo-Rasua)

law allows distance-bounding protocols to verify proximity by measuring the round-trip-time of a message exchange between a prover and a verifier [12].

Many protocols have suffered relay attacks in the past [5]. Famously, the EMV standard for contactless payments [13] now offers some protection against this type of attacks, and NXP's MIFARE Plus cards support proximity checks [25]. Both protocols use distance bounding as their main protection mechanism against relay attacks and have been extensively studied.

In general, a distance-bounding protocol consists of several rounds, in each of which a verifier challenges a prover to answer a query based on a shared secret key. Each round is timed by the verifier to check whether the prover is close. As a case in point, consider the protocol introduced by Hancke and Kuhn [18] depicted in Figure 1. First, the parties exchange one nonce each (N_v and N_p). The two nonces and a shared secret key k are used by both parties as input to a pseudo-random function PRF, which outputs a $2n$-bit sequence $T_1 \cdots T_{2n}$[1]. Next, prover and verifier engage in n challenge-response rounds. At the i-th round, the verifier generates and sends a random bit-challenge c_i. If $c_i = 0$, the prover replies with T_{2i-1}, otherwise with T_{2i}. The protocol succeeds if all responses are correct and all round-trip times are below a predefined threshold $\Delta_{\max}$. If $\Delta_{\max}$ is sufficiently small, then the protocol ensures proximity of the prover to the verifier, because the correct responses can only be generated by a nearby prover with the correct key k.

Hancke and Khun's protocol has the virtue of relying on a simple lookup operation by the prover during the time-measurement phase, minimising the impact that the prover's computational cost has on the round-trip-time measurement. Moreover, it makes a single call to a keyed pseudo-random function, making it suitable for resource-constrained devices. Hancke and Khun's protocol, however, offers suboptimal security. That is why distance-bounding protocols improving upon Hancke and Khun's design have been extensively studied [18, 3, 16, 19, 21, 24, 23, 26, 27, 28], and given the name of *lookup-based protocols* [23]. These protocols usually include a precomputation phase and a fast phase. This is different from more general distance-bounding protocols that may include a verification phase. Moreover, during each round of the fast phase the prover executes a single lookup operation, i.e. the response is already computed in memory.

The analysis of lookup-based protocols has mainly focused on two potential threats: *mafia fraud* and *distance fraud* [4, 6]. The former is a man-in-the-middle attack aimed at convincing the verifier that an honest far-away prover is close (see Figure 2a); the latter is an attack executed by a corrupt prover to pass the protocol from far-away (see Figure 2b). It was proven in [23] that the optimal resistance against mafia fraud is $\frac{1}{2^n}\left(\frac{n}{2}+1\right)$, where n is the number of round-trip-time measurements, and in [26] that the optimal resistance against distance fraud is $\frac{1}{2^n}$. It remains an open question, however, whether there exists

[1] For the sake of rigorousness, we should point out that [18] describes the protocol by using two registers of length n rather than a single bit-sequence of length $2n$.

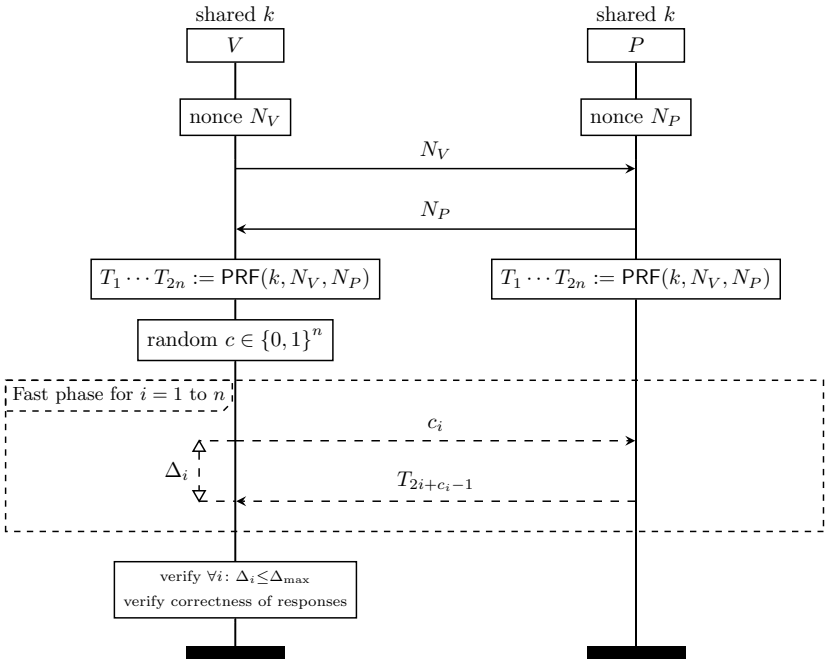


Figure 1: A lookup-based protocol by Hancke and Kuhn.

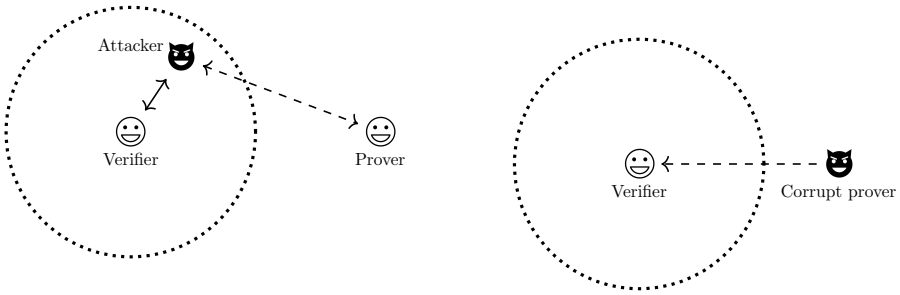
a lookup-based protocol achieving both optimal security bounds.

Contributions and structure. This article addresses the open question of whether lookup-based protocols can achieve optimal resistance against mafia fraud and distance fraud. We do so by extending the security model used in [23] to model lookup-based protocols with a game-based definition of security (Section 3). The resulting model is more general, in the sense that it makes fewer security assumptions, and allows us to prove that optimal security against mafia fraud and distance fraud are conflicting goals (Section 4). That is, optimality against one fraud moves the protocol away from optimality against the other fraud.

The second contribution of this work is the introduction of the first lookup-based protocol whose security bounds are either optimal or close to the optimal by a small constant factor (Section 5). More precisely, our protocol achieves $\frac{1}{2^n}$ security against distance fraud, which is optimal, and $\frac{1}{2^n}(n+1)$ security against mafia fraud, which is less than twice the optimal. By comparing those bounds with the security offered by existing lookup-based protocols (Section 6), we show that our protocol represents a significant improvement over the state-of-the-art.

2. Related work

The earliest distance-bounding protocol was proposed by Brands and Chaum [9]. This protocol achieves optimal security against mafia fraud and dis-



(a) **Mafia fraud:** the verifier believes the prover is close.

(b) **Distance fraud:** the verifier believes the corrupt prover is close.

Figure 2: The difference between mafia fraud and distance fraud. The encircled area denotes proximity to the verifier.

tance fraud by cryptographically signing the protocol transcripts after the time-measurement phase. Yet, it is vulnerable to a distance-hijacking attack [10]. With the years it has been improved [6], extending its security goals to the resistance of terrorist fraud [7]. One such improvement is the Swiss-knife protocol [22], whose main features are reaching optimal security against mafia fraud and resisting *terrorist fraud* to some extent. Terrorist fraud is an exotic type of fraud whereby a dishonest prover helps an adversary to defeat a distance-bounding protocol without allowing the adversary to impersonate him in future sessions of the protocol. The Swiss-knife protocol, while computationally more costly, does not offer better resistance to distance fraud than Hancke and Kuhn's protocol.

In 2005, Hancke and Kuhn [18] pointed out that using cryptographic signatures, or other expensive primitives, may cause many false failures of the protocol in noisy environments. Furthermore, it could hinder the deployment of distance-bounding on resource-constrained devices, such as RFID tags. They advocated for simplicity and computational efficiency when designing distance-bounding protocols and proposed the design displayed in Figure 1. Notably, the use of lookup operations during the time-measurement phase improves the precision of the distance estimation and prevents attacks in which the adversary overclocks the prover. Following [23], we refer to protocols adhering to the design principles outlined by Hancke and Kuhn as lookup-based protocols, which are the focus of this article. For a comprehensive overview of terrorist fraud and other classes of distance-bounding protocols, we refer the interested reader to [6].

Despite the performance virtues of lookup-based protocols, they have struggled to offer strong security against mafia and distance fraud attacks. This is a major drawback, as several scenarios exist where those attacks are of practical significance. For example, Kfir and Wool documented how relay attacks can be executed with NFC devices [20], which was later put into practice in [15]. The abuse of proximity claims through radio channels was first shown by Hancke [17], and later successfully implemented to attack the remote keyless entry system of

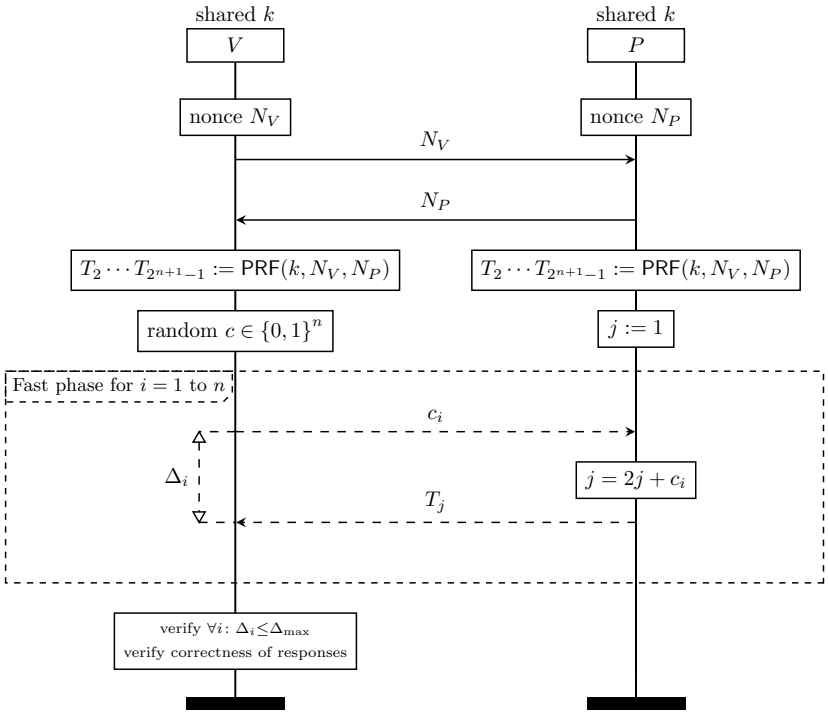


Figure 3: The tree-based protocol by Avoine and Tchamkerten. The tree is encoded in the binary sequence $T_2 \cdots T_{2^{n+1}-1}$.

ten car models from eight different manufacturers [14].

The first lookup-based protocol whose security against mafia fraud is close to the optimum value $\frac{1}{2^n}$, when $n \to \infty$, was introduced by Avoine and Tchamkerten in [3]. The protocol starts with an exchange of nonces between the prover and the verifier. The nonces and a shared secret key are used to secretly agree on the labels of a binary tree of depth n (see Figure 3). The labels of such binary tree are encoded within a sequence $T_1 T_2 \cdots T_{2^{n+1}-1}$ of size $2^{n+1} - 1$, where T_1 is the label of the root node and, for every $i \in \{1, \ldots, 2^n - 1\}$, T_i's left and right children have labels T_{2i} and T_{2i+1}, respectively. To bound its distance to the prover, the verifier executes n time measurements by sending n binary challenges. Starting from the root node, each binary challenge determines the child node whose label (also binary) is sent as reply to the challenge. The challenge-response game continues recursively until a leaf node is reached. The verifier authenticates the prover if all the responses are correct and arrive on time.

Avoine and Tchamkerten's protocol has a major drawback, though. The size of the tree grows exponentially with the number of RTT measurements. Note that it requires an encoding of size $2^{n+1} - 2$, while Hancke and Kuhn's protocol merely needs a sequence of length $2n$. This problem has been treated

by subsequent works, notably by [27] and [23], which resorted into different graph structures, rather than trees, to store the outcome of the precomputation phase. Avoine and Tchamkerten themselves offer a trade-off where a linear number trees of depth $d < n$ are used at the cost of downgraded security. Kim and Avoine proposed a different trade-off [21], one which degrades security against distance fraud to improve resistance to mafia fraud. Lastly, the SKI protocol [8] traded resistance to mafia fraud in exchange for some resistance to terrorist fraud. We observe that, while these works managed to find interesting trade-offs between memory size and security, none has solved the problem of finding a lightweight distance-bounding protocol with resistance to mafia fraud and distance fraud equal or close to their optimal values. That is the goal of this article. We show in Table 1 a quick comparison between lookup-based protocols. The first entry for our protocol is marked with an asterisk (*) because its mafia-fraud resistance is optimal, conditional to the optimality of resistance against distance fraud, as will be shown in Sections 4 and 5.

	Mafia Fraud	Distance Fraud	Space
HK [18]	no	no	linear
Tree Based [3]	yes	no	exponential
Poulidor [27]	no	no	linear
Modular [23]	no	no	linear
Our protocol	yes*	yes	linear

Table 1: Comparison of lookup-based protocols with respect to Mafia Fraud resistance optimality, Distance Fraud resistance optimality, and space requirements

We conclude this section by remarking that many security models have been proposed to analyse distance-bounding protocols, such as [8] and [11]. Notably, in [8], Boureanu and Vaudenay study the impossibility of resisting to all types of fraud efficiently. We are interested in a different question, though. Rather than merely resisting an attack, we are interested in *optimally* resisting an attack. For that, we need a dedicated security model particularly tailored to lookup-based protocols, which we introduce next.

3. The security model

Our security model takes the specification and execution model of [23], which defines a lookup-based protocol as a set of automata, but defines the security properties of interest in a more general way.

Lookup-based protocols contain two phases: a precomputation phase and a fast phase. During the first phase the prover and verifier exchange messages containing cryptographic information. As a result, both agents generate the same fresh random automaton, which is then used during the fast phase by the prover to compute the responses to the challenges generated randomly by the verifier. Both challenges and responses are single bits. For the security analysis of this type of protocol, it is sufficient to analyse the set of all possible automata

that can be generated during the precomputation phase, as in most cases the probability distribution used is (or is assumed to be) uniform.

3.1. Specification

We use an automata-based representation of lookup-based protocols as introduced in [23]. An automaton, i.e., a state-labelled deterministic finite automaton (DFA), is of the form $(\Sigma, \Gamma, Q, q_0, \delta, \ell)$, where Σ is a set of input symbols, Γ is a set of output symbols, Q is a set of states, $q_0 \in Q$ is the initial state, $\delta\colon Q \times \Sigma \to Q$ is a transition function, and $\ell\colon Q \to \Gamma$ is a labelling function. Given input and output symbol sets Σ and Γ, respectively, we use $\mathbf{U}_{\Sigma,\Gamma}$ to denote the universe of all DFAs over Σ and Γ.

Definition 1 (Lookup-based protocol [23])**.** *A* lookup-based protocol *with input set Σ and output set Γ is a finite non-empty subset of $\mathbf{U}_{\Sigma,\Gamma}$.*

Given an automaton $G = (\Sigma, \Gamma, Q, q_0, \delta, \ell)$ and a current state $q \in Q$, a lookup operation is regarded as a transition to a new state $q' = \delta(q, c)$ where $c \in \Sigma$ is a verifier's challenge. The corresponding response for such a challenge c_0 is the output symbol attached to the new state q', i.e., $\ell(q')$. This specification abstracts away from various protocol details that are present, for example, in the graphical descriptions given in Figure 1 and Figure 3, such as the precomputation phase and the round-trip-time measurements. These features of the protocol will be carefully considered when defining the execution model and security properties. For that, we need to introduce additional notation. For a sequence of input symbols $c = c_1 \cdots c_i \in \Sigma^i$:

- $\hat{\delta}(c) = \delta(\hat{\delta}(c_1 \dots c_{i-1}), c_i)$ if $i > 1$, otherwise $\hat{\delta}(c_1) = \delta(q_0, c_1)$. In a nutshell, $\hat{\delta}(c)$ returns the state reached by the input sequence c.
- $\hat{\ell}(c) = \ell(\hat{\delta}(c))$, which is the output symbol attached to the state reached by the sequence c.
- $\Omega(c)$ is used to represent the sequence of labels attached to the states $\hat{\delta}(c_1), \hat{\delta}(c_1 c_2), \ldots, \hat{\delta}(c_1 c_2 \cdots c_i)$ in that order, i.e., $\Omega(c) = r_1 \cdots r_i \in \Gamma^i$, where $r_j = \hat{\ell}(c_1 \cdots c_j)$ for $1 \le j \le i$.

As a running example we model the tree-based protocol [3] described in Figure 3. In this protocol, each tree (representing the automaton) is the full binary tree of depth n, which contains $2^{n+1} - 1$ nodes (representing the states of the automaton), where each node except the root is labelled randomly with either 0 or 1. This gives a total of $2^{2^{n+1}-2}$ different labelled trees.

Definition 2. *The tree-based protocol is the set of automata $\{G_1, \ldots, G_{2^{2^{n+1}-2}}\}$ where each automaton $G_i = (\Sigma, \Gamma, Q, q_0, \delta, \ell)$ satisfies:*

- $\Sigma = \Gamma = \{0, 1\}$,
- $Q = \{1, \ldots, 2^{n+1} - 1\}$ *and* $q_0 = 1$,

- *For every $j \in Q$ and $c \in \Sigma$, $\delta(j, c) = 2j + c$*
- *For every $1 \leq j \leq 2^{n+1} - 2$, $\ell(j+1)$ is the j-th least significant bit (rightmost bit) of the binary representation of the integer number i, which is the index of the automaton G_i.*

Observe that, for $n = 2$, the automaton G_{17} corresponds to the tree displayed in Figure 4. The labels follow from the binary representation of 17 (010001). Further, observe that the automaton G_{17} on input sequence $c = 01$ (represented in dashed arrows in Figure 4) gives:

- $\hat{\delta}(c) = \delta(\hat{\delta}(1), 1) = \delta(\delta(1, 0), 1) = \delta(2, 1) = 5$.
- $\hat{\ell}(c) = \ell(\hat{\delta}(01)) = 0$.
- $\Omega(c) = \hat{\ell}(0)\hat{\ell}(01) = 10$.

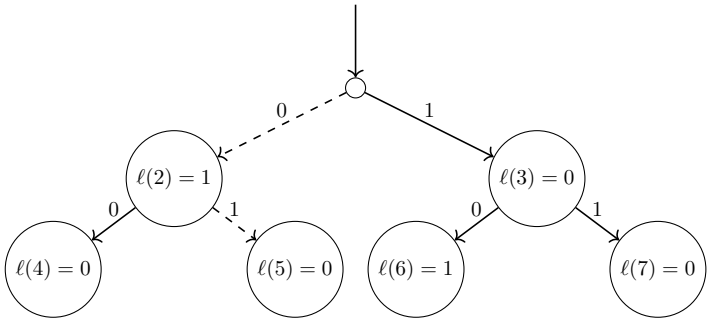


Figure 4: Example of a tree for an execution with two rounds. Dashed arrows indicate the path traversed on the challenges 0 and 1 (edge labels), resulting in the responses 1 and 0 (node labels), respectively.

Remark 1. *Like in [23], we argue informally that Definition 2 is a correct formalization of the protocol depicted in Figure 3, as the theory does not allow for a formal proof of equivalence between the two specifications. Throughout the paper we shall resort to a similar informal argumentation whenever we describe a protocol in graphical notation, as in Figure 3, and formalize it within the automata-based model, as in Definition 2.*

Using the above formalism to describe lookup-based protocols, instead of, for example, Figure 3, makes it possible to formalize properties of the protocol, such as the following.

Proposition 1. *Let P_{tree} be the tree-based protocol as in Definition 2. For any input sequence x, it holds that $\Pr\left[\hat{\ell}(x) = 0\right] = \Pr\left[\hat{\ell}(x) = 1\right] = \frac{1}{2}$, where the probability is taken over a random automaton sampled from P_{tree}. That is to say, if we look at $\hat{\ell}(x)$ as a random variable, then $\hat{\ell}(x)$ is uniformly distributed.*

We will use this proposition later in our security proofs.

3.2. Execution

The model in [23] abstracts away the precomputation phase in lookup-based protocols by considering the following execution model.

Definition 3 (Execution model)**.** *A* correct execution*, up to $n > 0$ rounds, of a lookup-based protocol P is a triple (G, C, R), where G is an automaton randomly selected from P, C is an input sequence randomly selected from Σ^n and R is an output sequence from Γ^n such that $R = \Omega(C)$.*

The outcome of the precomputation phase is considered to be a random automaton $G \in_R P$ within the set of automata defining the lookup-based protocol P. The input sequence $C = c_1 \cdots c_n$ corresponds to the verifier's challenges, and the correct replies $R = r_1 \cdots r_n$ must satisfy that $\hat{\ell}(c_1) = r_1, \hat{\ell}(c_1 c_2) = r_2, \ldots, \hat{\ell}(c_1 \cdots c_n) = r_n$.

Up to this point, the model has not established the role of the round-trip time of the exchanged messages and the location of prover and verifier. These features will be made part of the security properties by not allowing far-away participants to receive a verifier's challenge *in time*.

3.3. Security properties

In this article, we focus on the two most common and successful attack-strategies against lookup-based protocols: mafia fraud and distance fraud. Our focus on these two properties is because previous work [4, 6] have shown that terrorist fraud and distance-hijacking [10] pose no threat to lookup-based protocols.

3.3.1. Distance-fraud attack.

A distance-fraud attack consists of a corrupt prover trying to pass the protocol with a far-away verifier. To be successful, the corrupt prover cannot wait for the verifier's challenge to produce a response. Instead, the prover has to send its responses sufficiently in advance so that it passes the round-trip time condition.

Figure 5 shows the security experiment we use to define distance fraud, where we use $\mathcal{A}.Resp$ to denote the attacker's response algorithm (which in general may be randomized) and $\leftarrow\!\!\$$ to uniformly sample from a set. The experiment gives the adversary *white-box* access to an automaton G, representing the outcome of the precomputation phase of the protocol, and to the challenges of the verifier $c_1, \ldots, c_n$ in sequential order. At round i, the attacker knows all challenges sent by the verifier up to round $i - 1$, but not the challenge at the current round. This makes our model more conservative than previous models, such as [23], which assumes the corrupt prover is unaware of all the verifier's challenges.

Observe that the restriction of not letting the attacker receive the current challenge c_i, follows from the assumption that the corrupt prover (i.e. the attacker) is far away.

$\mathrm{Dist}^{\mathcal{A}}_{G,c_1\cdots c_n}(n)$

1 : **for** $i = 1$ **to** n
2 : $\quad r_i \leftarrow \mathcal{A}.Resp(G, c_1 \cdots c_{i-1})$
3 : **return** $r_1 \ldots r_n$

Figure 5: A security experiment for distance fraud.

Definition 4 (Distance-fraud resistance)**.** *The security of a lookup-based protocol P against a distance-fraud attacker $\mathcal{A}$, denoted $\mathsf{Adv}^{\mathrm{dist}}_{\mathcal{A},P}(n)$, is defined by,*

$$\Pr_{G \leftarrow\$ P, c_1 \cdots c_n \leftarrow\$ \{0,1\}^n}\left[\mathrm{Dist}^{\mathcal{A}}_{G,c_1\cdots c_n}(n) = \Omega(c_1 \cdots c_n)\right]$$

where the probability is over the coins (random source) of the adversary and the random choices of the automaton G and the verifier's challenges $c_1 \cdots c_n$.

In our security definition we do not let the attacker influence the choice of the automaton G. This may seem counter-intuitive, as the corrupt prover could, in practice, halt the precomputation phase various times looking for an automaton G with *good* properties for the attack. However, here we are being consistent with previous work on distance fraud [23, 26, 6, 4, 1, 16, 18], which considers an idealized precomputation phase whose outcome cannot be influenced by the attacker. We will informally discuss in our security analyses whether influencing the choice of G increases the adversary's probability of success.

3.3.2. Mafia fraud attack.

A mafia fraud attack is a man-in-the-middle attack in which the adversary, who is close to the verifier, interacts with a far-away prover to improve the odds on passing the time measurement phase with a legitimate verifier. This attack is fully defined by the choice of the adversary's challenges sent to the prover and the response function used by the adversary when trying to pass the time measurement phase. Therefore, we model the adversary $\mathcal{A}$ by two algorithms: $\mathcal{A}.Chall$ and $\mathcal{A}.Resp$. The former chooses the challenges the adversary sends to the prover, the latter chooses the responses the adversary provides to the verifier's challenges.

Figure 6 shows the security experiment we use to define mafia fraud. The experiment gives the adversary black-box access to an automaton G, representing the outcome of the precomputation phase of the protocol, and to the challenges of the verifier $c_1, \ldots, c_n$ in sequential order. At each round i, the adversary is let to choose a challenge y for the prover executing the protocol with automaton G by calling the algorithm $\mathcal{A}.Chall$ on input all previous challenges sent by the attacker, denoted x, the correct responses from the prover to those challenges, denoted $\Omega(x)$, and all challenges sent by the verifier in previous rounds. Note that both x and y can be empty sequences. The only restriction in this step is that the adversary cannot query the prover having knowledge about c_i, the verifier's challenge in the current round, because the prover is far-away. Lastly,

$\mathrm{Mafia}^{\mathcal{A}}_{G,c_1\cdots c_n}(n)$

1 : $x = \varepsilon$
2 : **for** $i = 1$ **to** $i = n$
3 : $\quad y \leftarrow \mathcal{A}.Chall(x, \Omega(x), c_1 \cdots c_{i-1})$
4 : $\quad x = x \| y$
5 : $\quad r_i \leftarrow \mathcal{A}.Resp(x, \Omega(x), c_1 \cdots c_i)$
6 : **return** $r_1 \ldots r_n$

Figure 6: A security experiment for mafia fraud.

the adversary produces the response r_i at round i by seeing the challenge c_i and the pair $(x, \Omega(x))$ collected during the pre-ask step. The output of the experiment is the sequence of responses from the adversary.

Definition 5 (Mafia fraud resistance)**.** *The security of a lookup-based protocol P against a mafia fraud attacker $\mathcal{A}$, denoted $\mathsf{Adv}^{\mathrm{mafia}}_{\mathcal{A},P}(n)$, is defined by,*

$$\Pr_{G \leftarrow\$ P, c_1 \cdots c_n \leftarrow\$ \{0,1\}^n}\left[\mathrm{Mafia}^{\mathcal{A}}_{G,c_1\cdots c_n}(n) = \Omega(c_1 \cdots c_n)\right]$$

where the probability is over the coins of the adversary and the random choices of the automaton G and the verifier's challenges $c_1 \cdots c_n$.

Unlike in distance fraud, where the attacker has white-box access to the prover, in mafia fraud both prover and verifier are assumed to be honest. Hence the attacker has no influence on the prover's choice with respect to the automaton. At best, the adversary could desynchronise prover and verifier in such a way that the prover's automaton is different from the verifier's automaton. Such a strategy, however, does not give the attacker an advantage and, therefore, is not considered in Definition 5.

4. Impossibility result towards optimal protocols

The goal of this section is to prove that no lookup-based protocol exists that can simultaneously achieve optimal resistance to mafia fraud and distance fraud. We do so in three steps. First, we prove that $\frac{1}{2^n}\left(\frac{n}{2}+1\right)$ is a tight lower bound on the security of lookup-based protocols against mafia fraud. Second, we provide sufficient and necessary conditions for a lookup-based protocol to resist distance fraud with probability $\frac{1}{2^n}$, which is trivially optimal. Third, we prove that meeting the optimal bound with respect to one attack-strategy moves the protocol away from the optimal bound with respect to the other attack-strategy.

Theorem 1. *Let P_{tree} be the tree-based protocol given in Definition 2. Then for all attackers $\mathcal{A}$ we have $\mathsf{Adv}^{\mathrm{mafia}}_{\mathcal{A},P_{tree}}(n) \leq \frac{1}{2^n}\left(\frac{n}{2}+1\right)$. Moreover, the bound is tight.*

Proof. We start proving tightness of the bound. Consider the attacker $\mathcal{A}$ defined as follows, for every $i \in \{1, \ldots, n\}$, $c_1 \cdots c_i$, and an arbitrary sequence x (determined by $\mathcal{A}$),

$$\mathcal{A}.Resp(x, \Omega(x), c_1 \cdots c_i) = \begin{cases} \hat{\ell}(x_1 \cdots x_i) & \text{if } x_1 \cdots x_i = c_1 \cdots c_i \\ \text{random} & \text{otherwise.} \end{cases}$$

This attacker succeeds with certainty in round i as long as $x_1 \cdots x_i = c_1 \cdots c_i$, otherwise he succeeds with probability $1/2$. Let X be the random variable giving the round number i where the attacker does not correctly guess c_i. That is, $\Pr[X = i] = \Pr[x_1 \cdots x_{i-1} = c_1 \cdots c_{i-1} \wedge x_i \neq c_i] = \frac{1}{2^i}$, where the last equality follows from $c_1 \cdots c_i$ being a random sequence independently generated. Then,

$$\Pr\left[\text{Mafia}^{\mathcal{A}}_{G,c_1\cdots c_n}(n) = \Omega(c_1 \cdots c_n)\right] = \sum_{i=1}^{i=n} \Pr\left[\text{Mafia}^{\mathcal{A}}_{G,c_1\cdots c_n}(n) = \Omega(c_1 \cdots c_n) \;\middle|\; X = i\right] \Pr[X = i] + \frac{1}{2^n}$$

The last term in the sum $\frac{1}{2^n}$ accounts for the case where $x_1 \cdots x_n = c_1 \cdots c_n$. Now, $\Pr\left[\text{Mafia}^{\mathcal{A}}_{G,c_1\cdots c_n}(n) = \Omega(c_1 \cdots c_n) \;\middle|\; X = i\right] = \frac{1}{2^{n-i+1}}$. This gives,

$$\Pr\left[\text{Mafia}^{\mathcal{A}}_{G,c_1\cdots c_n}(n) = \Omega(c_1 \cdots c_n)\right] = \sum_{i=1}^{i=n} \frac{1}{2^{n-i+1}} \times \frac{1}{2^i} + \frac{1}{2^n} = \frac{1}{2^n}\left(\frac{n}{2} + 1\right)$$

Observe that the above advantage is the same for every automaton G, which finishes this part of the proof. To prove that this is an optimal bound, consider the random variable $\hat{\ell}(c_1 \cdots c_i)$. According to Proposition 1, $\hat{\ell}(c_1 \cdots c_i)$ is uniform and, unless $\mathcal{A}.Chall$ outputs $x_1 \cdots x_i = c_1 \cdots c_i$, then the adversary cannot guess $\hat{\ell}(c_1 \cdots c_i)$ with probability better than $\frac{1}{2}$. As we established earlier, the advantage in this case for the adversary is $\frac{1}{2^n}\left(\frac{n}{2} + 1\right)$. □

Once we have established that the tree-based protocol optimally resists mafia fraud with probability $\frac{1}{2^n}\left(\frac{n}{2} + 1\right)$, we move our attention to the analysis of distance fraud.

Theorem 2. *A lookup-based protocol P satisfies $\mathsf{Adv}^{\text{dist}}_{\mathcal{A},P}(n) \leq \frac{1}{2^n}$ for all attackers $\mathcal{A}$ if and only if for every $G \in P$ and every (possibly empty) sequence x of size lower than n it holds that $\hat{\ell}(x\|0) \neq \hat{\ell}(x\|1)$.*

Proof. If for every $G \in P$ and every (possibly empty) sequence x of size lower than n it holds that $\hat{\ell}(x\|0) \neq \hat{\ell}(x\|1)$, then,

$$\Pr_{c \leftarrow\$ \{0,1\}}\left[\hat{\ell}(x\|c) = 0\right] = \Pr_{c \leftarrow\$ \{0,1\}}\left[\hat{\ell}(x\|c) = 1\right] = \frac{1}{2}$$

This means $r_i \leftarrow \mathcal{A}.Resp(G, c_1 \cdots c_{i-1})$ has probability $1/2$ to be equal to $\hat{\ell}(c_1 \cdots c_{i-1}, c_i)$, because the adversary does not have c_i. This proves one direction of the double implication. To prove the other direction we proceed via contradiction.

Assume there exists $G \in P$ and a (possibly empty) sequence x such that $\hat{\ell}(x\|0) = \hat{\ell}(x\|1)$. We build the following adversary, for every G',

$$\mathcal{A}.Resp(G', c_1 \cdots c_{i-1}) = \begin{cases} \hat{\ell}(x\|0) & \text{if } G = G' \text{ and } x = c_1 \cdots c_{i-1} \\ \text{random} & \text{otherwise.} \end{cases}$$

Note that such an adversary can be built because the adversary has white-box access to G'. Further note that we are considering syntactical equality between G and G', although the proof works with equality modulo isomorphism too.

If $G' = G$ and $x = c_1 \cdots c_{i-1}$, then the probability of success of this adversary is 1 at round i, because $\hat{\ell}(x\|0) = \hat{\ell}(x\|1)$. In any other situation, the adversary wins with probability $1/2$. Because the event $G' = G$ and $x = c_1 \cdots c_{i-1}$ has non-zero probability, it follows that the probability of success of the adversary is strictly larger than $\frac{1}{2^n}$. □

It is worth noting that the result above holds even if the adversary is allowed to choose the automaton G. The reason is that the necessary and sufficient condition stated in Theorem 2 has to be satisfied by all automata in the protocol, thereby giving the adversary the same success probability for all automata.

We now proceed to prove the main result of this section.

Theorem 3. *For every lookup-based protocol P there exists a pair of adversaries $(\mathcal{A}_m, \mathcal{A}_d)$ such that either $\mathsf{Adv}^{\text{mafia}}_{\mathcal{A}_m,P}(n) > \frac{1}{2^n}(n+1)$ or $\mathsf{Adv}^{\text{dist}}_{\mathcal{A}_d,P}(n) > \frac{1}{2^n}$.*

Proof. As established earlier, optimality in terms of distance fraud implies that, for every $G \in P$ and every (possibly empty) sequence x of size lower than n it holds that $\hat{\ell}(x\|0) \neq \hat{\ell}(x\|1)$. This allows us to build the following mafia fraud adversary, for every $i \in \{1, \ldots, n\}$, $c_1 \cdots c_i$, and sequence x,

$$\mathcal{A}.Resp(x, \Omega(x), c_1 \cdots c_i) = \begin{cases} \hat{\ell}(x_1 \cdots x_i) & \text{if } x_1 \cdots x_i = c_1 \cdots c_i \\ 1 - \hat{\ell}(x_1 \cdots x_i) & \text{if } x_1 \cdots x_{i-1} = c_1 \cdots c_{i-1} \wedge x_i \neq c_i \\ \text{random} & \text{otherwise.} \end{cases}$$

Because $\hat{\ell}(c_1 \cdots c_{i-1}\|0) \neq \hat{\ell}(c_1 \cdots c_{i-1}\|1)$, it follows that the attacker succeeds with certainty if $x_1 \cdots x_{i-1} = c_1 \cdots c_{i-1}$, otherwise he succeeds with probability $\frac{1}{2}$. The rest of this proof is similar to the proof of Theorem 1. Let X be the random variable giving the round number i where the attacker does not correctly guess c_i. Then,

$$\Pr\Big[\text{Mafia}^{\mathcal{A}}_{G,c_1\cdots c_n}(n) = \Omega(c_1\cdots c_n)\Big] =$$
$$\sum_{i=1}^{i=n} \Pr\Big[\text{Mafia}^{\mathcal{A}}_{G,c_1\cdots c_n}(n) = \Omega(c_1\cdots c_n) \;\Big|\; X = i\Big] \Pr[X = i] + \frac{1}{2^n}$$

Now, $\Pr\Big[\text{Mafia}^{\mathcal{A}}_{G,c_1\cdots c_n}(n) = \Omega(c_1\cdots c_n) \;\Big|\; X = i\Big] = \frac{1}{2^{n-i}}$. This gives,

$$\Pr\Big[\text{Mafia}^{\mathcal{A}}_{G,c_1\cdots c_n}(n) = \Omega(c_1\cdots c_n)\Big] = \sum_{i=1}^{i=n} \frac{1}{2^{n-i}} \times \frac{1}{2^i} + \frac{1}{2^n} = \frac{1}{2^n}(n+1)$$

□

In addition to the impossibility result above, we obtain two main insights from the theoretical results in this section. First, there is a simple strategy to achieve optimality in terms of distance fraud resistance (see Theorem 2). Second, the variant of the tree-based protocol implementing such strategy seems to achieve a $\frac{1}{2^n}(n+1)$ security value against mafia fraud, which approximates the optimal value by a small constant factor lower than 2. Thus, our next step is to build a memory-efficient protocol with resistance to mafia fraud and distance fraud, respectively, equal to $\frac{1}{2^n}(n+1)$ and $\frac{1}{2^n}$.

5. A lookup-based protocol with nearly optimal security bounds

This section provides a protocol design that is memory efficient and nearly optimal in terms of mafia fraud and distance fraud.

5.1. Protocol specification in graphical notation

To specify our protocol in graphical notation we will use explicit lookup tables with domain the naturals and co-domain $\{0,1\}$. We write T_i instead of $T(i)$ to denote the output of T on index i, with i a natural number, and $T_i = b$ to denote the update of T with the value b on input the index i. Given a sequence of bits $b_1\cdots b_n$ and an ordered set of keys $\Psi = \{i_1,\ldots,i_n\}$, we use $T_\Psi = b_1\cdots b_n$ to denote the sequence of updates $T_{i_1} = b_1,\ldots,T_{i_n} = b_n$.

Our protocol (depicted in Figure 7) relies on a parameter d, which determines the maximum depth of the tree used throughout the protocol. Its most prominent feature is that it keeps a lookup table $T_2,\ldots,T_{2^{d+1}-1}$ of size $2^{d+1}-2$, which exponentially decreases with d. All bits $T_2, T_4,\ldots,T_{2^{d+1}-2}$ are randomly sampled, while $T_i = 1 - T_{i-1}$ for every odd index $i \in [3, 2^{d+1}-1]$. During the fast phase, rounds are divided in blocks of exactly d rounds each. Throughout the protocol, the variable *seed* stores the index of the current node in the tree, which corresponds to the number whose binary representation is $c_1\cdots c_i$, where i is the round number and $c_1\cdots c_i$ the challenges sent by the verifier up to the current round. When the depth of *seed* is a multiple of d, which coincides with

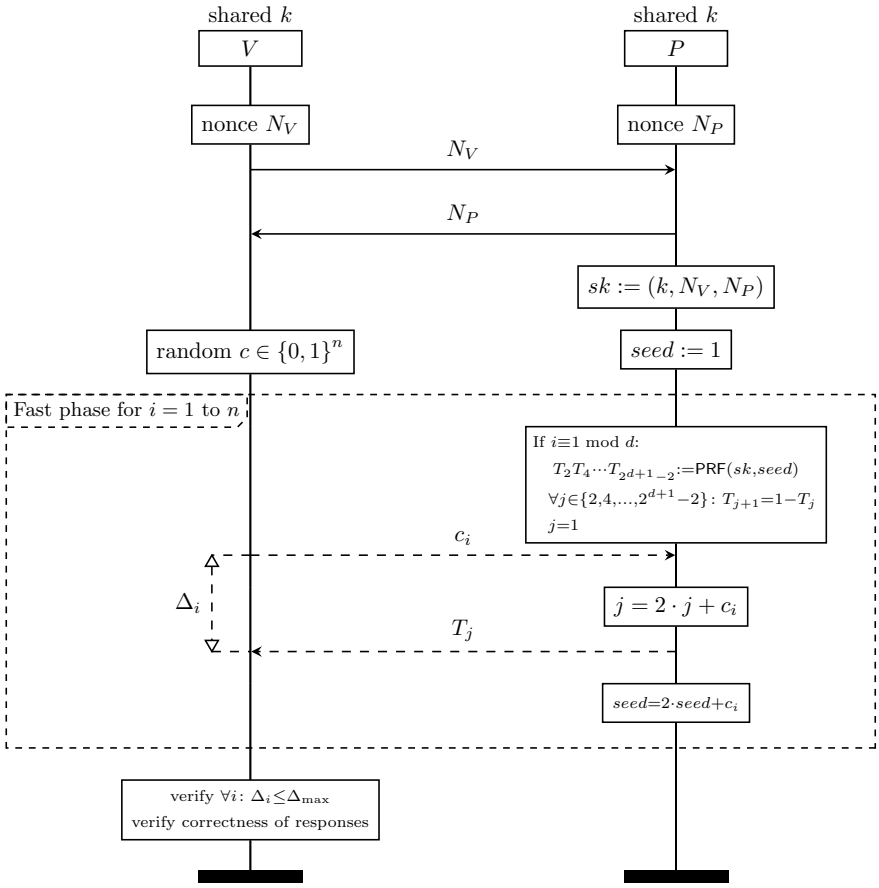


Figure 7: Our nearly optimal protocol. The parameter d determines the output length of the pseudo-random function.

the condition $i \equiv 1 \bmod d$, i.e. the first round in each block, the PRF is used to compute the $2^d - 1$ random bits necessary to run the protocol for the next d rounds. These bits will be assigned to the left child of each non-leaf node in the subtree of depth d rooted in the node *seed*. The right child will get the opposite value. Inside each block, the variable j represents the relative index of the current node with respect to the root node of the subtree rooted in *seed*.

Observe that, for $n = 4$, $d = 3$, the automaton used by our protocol corresponds to the tree displayed (partially) in Figure 8. The input sequence $c = 0101$ is represented in dashed arrows. The outputs of the pseudorandom function are: $\mathsf{PRF}(sk, 1) = T_2T_4T_6 = 110$ and $\mathsf{PRF}(sk, 5) = T_2T_4T_6 = 011$.

At first sight, the way bits are sampled in the protocol might look unnecessarily complex in comparison to the way bits are sampled in the tree-based protocol shown in Figure 3. While the tree-based protocol uses a single call

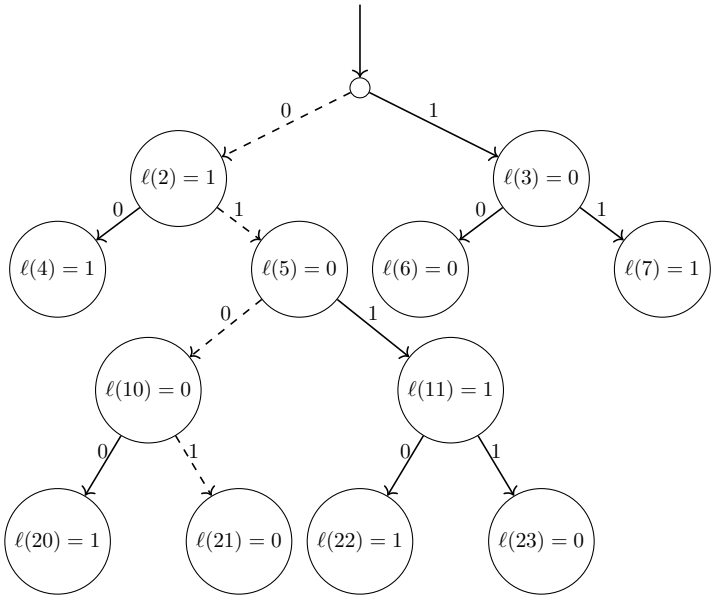


Figure 8: Example tree for an execution with 4 rounds. Dashed arrows indicate the path traversed on the challenges 0101 (edge labels), resulting in the responses 1000 (node labels), respectively.

to a pseudo-random function with output length $2^{n+1} - 1$, ours uses several calls to a pseudo-random function with output length $2^{d+1} - 1$, where d is a protocol parameter and divisor of n. Concretely, our protocol makes $\frac{n}{d}$ calls to the pseudo-random function. The advantage of our approach, however, is that we keep a lookup table of size $2^{d+1} - 1$ as opposed to the lookup-table of size $2^{n+1} - 1$ needed by the tree-based protocol. That is to say, for small values of d, say $d = \log_2 n$, our protocol requires little memory to be executed. Next, we address the question of whether this improvement in terms of memory size is traded-off by a decrease in resistance to mafia-fraud and distance-fraud attacks.

5.2. Security analysis

We analyse our protocol by reducing it to the protocol specified in Figure 9, which does not optimise memory and it is simpler to analyse. Rather than calculating trees of depth $2^{d+1} - 1$ every d rounds, the protocol in Figure 9 calculates the entire tree prior the execution of the fast phase, just like the original tree-based protocol does. The protocol identifies a node by a sequence of bits $c_1 \cdots c_i$, denoting the path from the root to the node. Because lookup tables have domain the naturals, we use the auxiliary function $dec(.)$ to obtain the decimal representation of a binary sequence and identify a node $c_1 \cdots c_i$ with its decimal representation $dec(1 \| c_1 \cdots c_i)$ (see Figure 4 for an example). Based on this mapping from nodes in the tree to positive integers, the protocol samples bits from a pseudo-random function of fixed output length equal to $2^{d+1} - 1$

bits as follows. For every node $c_1 \cdots c_i$ with i a multiple of d smaller than n, the protocol randomly sample half of the labels of the subtree with depth d rooted in $c_1 \cdots c_i$. Concretely, it samples the nodes whose identifiers are even numbers, assigning the opposite value to their siblings.

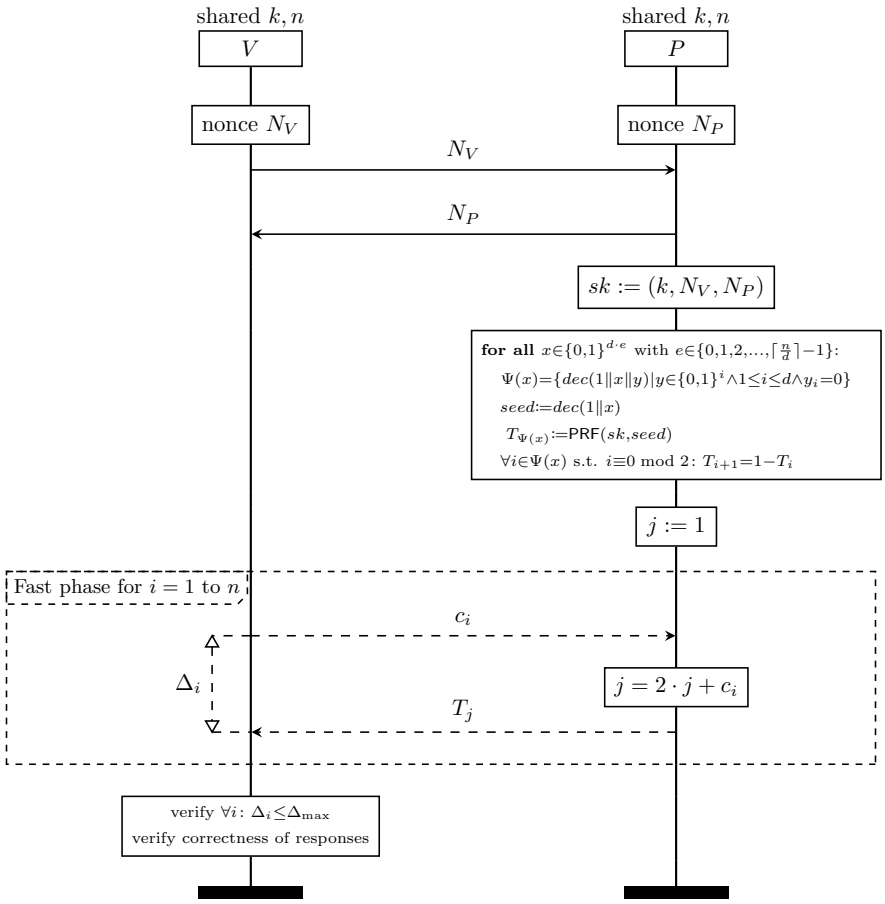


Figure 9: An alternative specification of our protocol used to analyse its security properties.

Lemma 1. *For fixed nonces N_V and N_P, the protocols in Figure 9 and Figure 7 gives the same output on any sequence of challenges.*

Proof. We proceed with the proof assuming that d is a divisor of n. Let $c_1 \cdots c_i$ be a sequence of challenges and $e = \lfloor \frac{i}{d} \rfloor$. Observe that the variable *seed* is identical in both protocols, in the sense that it stores the decimal representation of $1 \| c_1 \cdots c_{d \cdot e + 1}$. This means that the sequence $T_2 \cdots T_{2^{d+1}-2}$ produced by $\mathsf{PRF}(sk, seed)$ in Figure 7 is equal to the sequence $T_{\Psi_x} = PRF(sk, seed)$ produced in Figure 9, where x is the binary representation of *seed* and $\Psi(x) =$

$\{dec(1\|x\|y)|y \in \{0,1\}^i \wedge 1 \leq i \leq d \wedge y_i = 0\}$. Although not explicitly mentioned earlier, we are assuming that $\Psi(x)$ is sorted in ascending order. This means that the position of j, as calculated in Figure 9, within $\Psi(x)$ is equal to the position of j, as calculated in Figure 7, within $T_2 \cdots T_{2^{d+1}-2}$, which implies that T_j stores the same value in both protocols. □

The lemma above proves that the protocols in Figure 7 and Figure 9 produce identical outputs after fixing the nonces N_V and N_P. This means that any adversary advantage when given black-box access to one protocol becomes an identical advantage when given black-box access to the other protocol. Therefore, it is sufficient for us to analyse the adversary advantage over Figure 9.

To analyse the security properties of the protocol in Figure 9, we provide its formalisation in Definition 6 using the automata-based model defined earlier. The formalisation is identical to the tree-based protocol, except that it removes all automata that do not satisfy $\ell(s) \neq \ell(s+1)$ for every state s in $\{2, 4, \ldots, 2^{n+1} - 2\}$. That is to say, we make the labelling function guarantee that for every node in the tree the labels of its children are different. We argue that Definition 6 is a correct formalisation of the protocol in Figure 9 by noting that all calls to the pseudo-random function in Figure 9 use a different seed. Therefore, the sequence $T_2 T_4 \cdots T_{2^{n+1}-2}$ is indistinguishable from a random sequence, just like in the tree-based protocol.

Definition 6 (Specification of our protocol as a set of automata)**.** *Our protocol is modelled by the set of automata* $\{G_1, \ldots, G_{2^{2^{n+1}-2}}\}$ *where each automaton* $G_i = (\Sigma, \Gamma, Q, q_0, \delta, \ell)$ *satisfies:*

- $\Sigma = \Gamma = \{0, 1\}$,
- $Q = \{1, \ldots, 2^{n+1} - 1\}$ *and* $q_0 = 1$,
- *For every* $j \in \{1, \ldots, 2^n - 1\}$, *and* $c \in \Sigma$,
  - $\delta(j, c) = 2j + c$
  - $\ell(2 \cdot j)$ *is the* j*-th least significant digit of the binary representation of the integer number* i
  - $\ell(2 \cdot j + 1) = 1 - \ell(2 \cdot j)$

It follows directly from Theorem 2 that this protocol is optimal in terms of distance fraud. Next we prove that the protocol is nearly optimal in terms of mafia fraud.

Theorem 4. *Let* P_{opt} *be the protocol given in Definition 6. Then for all attackers* $\mathcal{A}$ *we have* $\mathsf{Adv}^{\text{mafia}}_{\mathcal{A},P_{opt}}(n) \leq \frac{1}{2^n}(n+1)$, *and* $\mathsf{Adv}^{\text{dist}}_{\mathcal{A},P_{opt}}(n) \leq \frac{1}{2^n}$.

Proof. It remains to prove the upper bound for mafia fraud attackers. The proof follows closely the one from Theorem 1. Consider an attacker $\mathcal{A}$. Let X be the random variable giving the round number i where the attacker does not correctly guess c_i. Then,

$$\Pr\Big[\mathrm{Mafia}^{\mathcal{A}}_{G,c_1\cdots c_n}(n) = \Omega(c_1\cdots c_n)\Big] =$$
$$\sum_{i=1}^{i=n} \Pr\Big[\mathrm{Mafia}^{\mathcal{A}}_{G,c_1\cdots c_n}(n) = \Omega(c_1\cdots c_n) \;\Big|\; X = i\Big] \Pr[X = i] + \frac{1}{2^n}$$

Now, $\Pr[X = i] = \frac{1}{2^i}$ because the challenges chosen independently and uniformly at random. The value $\Pr\Big[\mathrm{Mafia}^{\mathcal{A}}_{G,c_1\cdots c_n}(n) = \Omega(c_1\cdots c_n) \;\Big|\; X = i\Big]$ is upper bounded by $\frac{1}{2^{n-i}}$ because if the adversary guesses incorrectly c_i, then from the value of $\Omega(x_1\cdots x_n)$ it gets no information about the labels of the nodes in the subtree determined by the path $c_1\cdots c_{i+1}$, so the best it can do is randomly guess these values, and there are at least $n-i$ such values. We conclude:

$$\Pr\Big[\mathrm{Mafia}^{\mathcal{A}}_{G,c_1\cdots c_n}(n) = \Omega(c_1\cdots c_n)\Big] \leq \sum_{i=1}^{i=n} \frac{1}{2^{n-i}} \times \frac{1}{2^i} + \frac{1}{2^n} = \frac{1}{2^n}(n+1)$$

□

6. Evaluation

The goal of this section is to compare the security bounds of our protocol against previous lookup-based protocols. To make our comparison comprehensive, yet concise, we only consider the protocols regarded relevant by the decision-making methodology due to Avoine et al. [2], while adding the Hancke and Kuhn protocol (HK protocol for short) as a baseline. Concretely, we consider the protocols HK [18], Tree based [3], Modular [23] and Poulidor [27]. The security of these protocols in terms of mafia and distance fraud are taken from the framework available at `https://github.com/rolandotr/db_comparison`.

6.1. Evaluation setting

Because most distance-bounding protocols require a memory size that linearly grows with n, we shall restrict protocols to a linear memory size. This means that, for protocols whose memory requirement depends on protocol parameters, we set them such that the memory usage scales linearly with the number of rounds. In particular, for:

- The tree-based protocol: the depth of each tree is set to 6, which gives a memory size of approximately $21 \cdot n$ bits.
- The Modular protocol: the width of the graph is set to $u = 4$, which gives a memory size of $16 \cdot n$ bits.
- Our protocol: the depth of each tree is set to $\lfloor \log(n) \rfloor$, which gives a memory size of n bits.

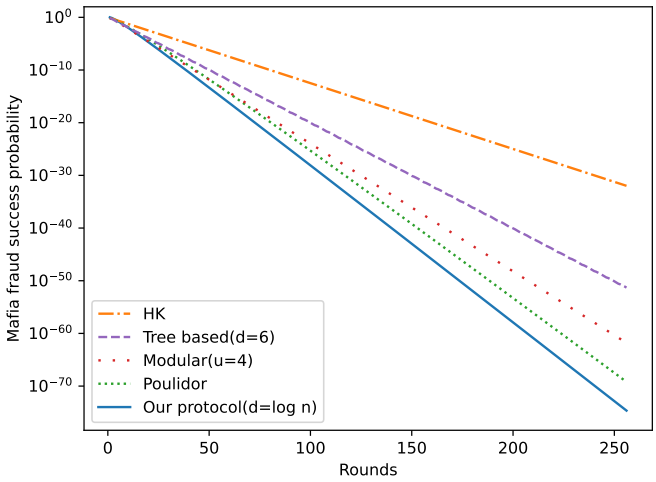


Figure 10: Mafia fraud success probability.

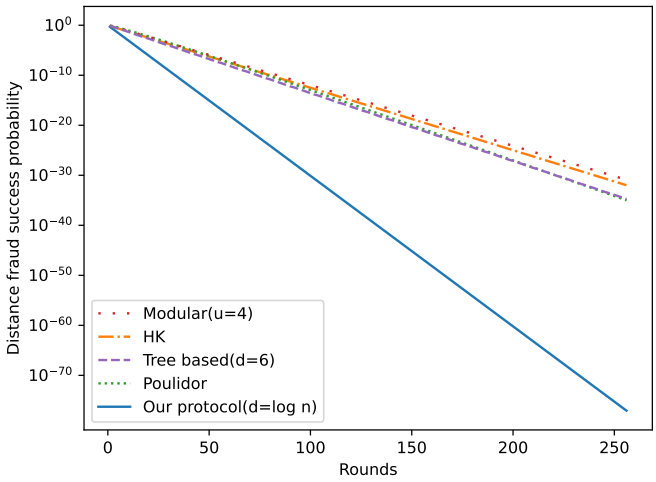


Figure 11: Distance fraud success probability.

6.2. Comparison results

In Figure 10 and Figure 11 we compare the protocols in terms of their resistance to mafia fraud and distance fraud, respectively. Our protocol clearly outperforms the rest in terms of both types of fraud, with Poulidor being its closest competitor. In terms of distance fraud, our protocol outperforms the others

by a somewhat larger margin. This is not so surprising, as it was specifically designed to be optimal with respect to this fraud.

7. Conclusions

We presented a novel lookup-based distance-bounding protocol that provides (close to) optimal protection against distance fraud and mafia fraud attacks. We demonstrated the impossibility of achieving optimal protection against both types of fraud, and derived a tight lower bound for mafia fraud when distance fraud resistance is optimal. Furthermore, we conducted a comparative analysis of our protocol against previous lookup-based protocols in terms of their resistance to mafia fraud and distance fraud. The results show that our protocol outperforms others in mitigating both types of fraud.

As a future direction, we intend to implement our proposed protocol using currently available hardware to obtain real-world measurements of its speed and energy consumption.

References

[1] Gildas Avoine, Christian Floerkemeier, and Benjamin Martin. "RFID Distance Bounding Multistate Enhancement". *Proc. Progress in Cryptology (INDOCRYPT'09)*. Vol. 5922. LNCS. Springer, 2009.

[2] Gildas Avoine, Sjouke Mauw, and Rolando Trujillo-Rasua. "Comparing distance bounding protocols: A critical mission supported by decision theory". *Computer Communications* 67 (2015).

[3] Gildas Avoine and Aslan Tchamkerten. "An efficient distance bounding RFID authentication protocol: Balancing false-acceptance rate and memory requirement". *Proc. 12th International Conference on Information Security (ISC'09)*. Vol. 5735. LNCS. Pisa, Italy: Springer, 2009.

[4] Gildas Avoine et al. "A framework for analyzing RFID distance bounding protocols". *Journal of Computer Security* 19.2 (2011). DOI: 10.3233/JCS-2010-0408.

[5] Gildas Avoine et al. "From relay attacks to distance-bounding protocols". *Security of Ubiquitous Computing Systems: Selected Topics* (2021).

[6] Gildas Avoine et al. "Security of distance-bounding: A survey". *ACM Comput. Surv.* 51.5 (2019). DOI: 10.1145/3264628.

[7] Samy Bengio et al. "Secure implementation of identification systems". *Journal of Cryptology* 4.3 (1991).

[8] Ioana Boureanu, Aikaterini Mitrokotsa, and Serge Vaudenay. "Practical and Provably Secure Distance-Bounding". *Information Security*. Ed. by Yvo Desmedt. Cham: Springer International Publishing, 2015. ISBN: 978-3-319-27659-5.

[9] Stefan Brands and David Chaum. "Distance-bounding protocols". *Proc. Advances in Cryptology (EUROCRYPT'93)*. Vol. 765. LNCS. Lofthus, Norway: Springer, 1994.

[10] Cas Cremers et al. "Distance hijacking attacks on distance bounding protocols". *2012 IEEE Symposium on Security and Privacy*. 2012. DOI: 10.1109/SP.2012.17.

[11] Alexandre Debant, Stéphanie Delaune, and Cyrille Wiedling. "So near and yet so far - Symbolic verification of distance-bounding protocols". *ACM Trans. Priv. Secur.* 25.2 (2022). DOI: 10.1145/3501402.

[12] Yvo Desmedt, Claude Goutier, and Samy Bengio. "Special uses and abuses of the Fiat-Shamir passport protocol". *CRYPTO'87*. Vol. 293. LNCS. Springer, 1988.

[13] E. M. V. EMVCo. "Contactless Specifications for Payment Systems". *Book C-2, Kernel* 2 (2021).

[14] Aurélien Francillon, Boris Danev, and Srdjan Capkun. "Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars". *Proceedings of the Network and Distributed System Security Symposium, NDSS 2011, San Diego, California, USA, 6th February - 9th February 2011*. 2011.

[15] Lishoy Francis et al. "Practical NFC peer-to-peer relay attack using mobile phones". *Radio Frequency Identification: Security and Privacy Issues - 6th International Workshop, RFIDSec 2010, Istanbul, Turkey, June 8-9, 2010, Revised Selected Papers*. 2010.

[16] Ali Özhan Gürel, Atakan Arslan, and Mete Akgün. "Non-uniform stepping approach to RFID distance bounding problem". *Proc. 5th International Workshop on Data Privacy Management (DPM'10), and 3rd International Conference on Autonomous Spontaneous Security (SETOP'10)*. Vol. 6514. LNCS. Springer, 2011.

[17] Gerhard P. Hancke. "Practical attacks on proximity identification systems (Short paper)". *Proceedings of the 2006 IEEE Symposium on Security and Privacy*. SP '06. Washington, DC, USA: IEEE Computer Society, 2006. ISBN: 0-7695-2574-1. DOI: 10.1109/SP.2006.30.

[18] Gerhard P. Hancke and Markus G. Kuhn. "An RFID distance bounding protocol". *Proc. First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm-2005)*. Washington, DC, USA: IEEE Computer Society, 2005.

[19] Süleyman Kardas et al. "A novel RFID distance bounding protocol based on physically unclonable functions". *International Conference on Radio Frequency Identification: Security and Privacy Issues*. Springer, 2012.

[20] Ziv Kfir and Avishai Wool. "Picking virtual pockets using relay attacks on contactless smartcard". *First International Conference on Security and Privacy for Emerging Areas in Communications Networks, (SecureComm-2005)*. 2005.

[21] Chong Hee Kim and Gildas Avoine. "RFID distance bounding protocols with mixed challenges". *IEEE Transactions on Wireless Communications* 10.5 (2011).

[22] Chong Hee Kim et al. "The Swiss-Knife RFID distance bounding protocol". *The 11th International Conference on Information Security and Cryptology - ICISC 2008*. Vol. 5461. Lecture Notes in Computer Science. Springer-Verlag, Dec. 2008.

[23] Sjouke Mauw, Jorge Toro-Pozo, and Rolando Trujillo-Rasua. "A class of precomputation-based distance-bounding protocols". *Proc. 1st IEEE European Symposium on Security and Privacy – EuroS&P*. Saarbrücken, Germany, 2016.

[24] Jorge Munilla and Alberto Peinado. "Distance bounding protocols for RFID enhanced by using void-challenges and analysis in noisy channels". *Wireless Communications and Mobile Computing* 8.9 (2008).

[25] Peter Thueringer et al. *Decoupling of measuring the response time of a transponder and its authentication*. US Patent 10,044,512. Aug. 2018.

[26] Rolando Trujillo-Rasua. "Complexity of distance fraud attacks in graph-based distance bounding". *Proc. 10th International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (MOBIQUITOUS'13)*. Vol. 131. LNCS. Springer, 2013.

[27] Rolando Trujillo-Rasua, Benjamin Martin, and Gildas Avoine. "The Poulidor distance-bounding protocol". *Proc. 6th International Conference on Radio Frequency Identification: Security and Privacy Issues (RFIDSec'10)*. Vol. 6370. LNCS. Istanbul, Turkey: Springer, 2010.

[28] Yu-Ju Tu and Selwyn Piramuthu. "RFID distance bounding protocols". *Proc. First International EURASIP Workshop on RFID Technology*. 2007.