A Class of Precomputation-Based Distance-Bounding Protocols*

Sjouke Mauw[1,2] Jorge Toro-Pozo[1] Rolando Trujillo-Rasua[2]

[1]CSC / [2]SnT, University of Luxembourg
6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg
{sjouke.mauw, jorge.toro, rolando.trujillo}@uni.lu

Abstract

Distance-bounding protocols serve to thwart various types of proximity-based attacks, such as relay attacks. A particular class of distance-bounding protocols measures round trip times of a series of one-bit challenge-response cycles, during which the proving party must have minimal computational overhead. This can be achieved by precomputing the responses to the various possible challenges.

In this paper we study this class of precomputation-based distance-bounding protocols. By designing an abstract model for these protocols, we can study their generic properties, such as security lower bounds in relation to space complexity. Further, we develop a novel family of protocols in this class that resists well to mafia fraud attacks.

1 Introduction

Physical proximity is a common requirement in many access control policies, particularly in those involving physical access. Most real-world security systems would rise a security alert if a door has been opened remotely. An electronic toll payment made by a user whose car is parked in front of his home can also hardly be considered an expected behavior. Some access control mechanisms have been designed in such a way that physical proximity is enforced easily, e.g., mechanical locks or biometric identification. However, due to the open nature of wireless channels, providing the same kind of guarantee in wireless systems is far from trivial.

Simple proximity enforcing techniques, such as setting up small communication timeouts or short-range communication channels, can be easily circumvented in practice by a variety of attacks [1]. Perhaps, the most popular and devastating of such attacks is *mafia fraud* [2], also known as *relay attack* [3]. This fraud simply consists in relaying all communication between two wireless devices, making them believe that they have a direct communication.

*This work was accepted for publication at the proceedings of the *1st IEEE European Symposium on Security and Privacy, EuroS&P, 2016*. This is the authors' preprint version. For reference, use the published version with DOI 10.1109/EuroSP.2016.19. This work was supported by the Luxembourg National Research Fund (FNR) under the grants AFR-PhD-10188265 and C15-IS-10428112.

Case in point, let's assume that Mallory wants to get unauthorized access to Alice's office, and that Alice opens the door of her office by simply swiping her personal contactless token over the door's card reader. Mallory can achieve her goal by executing a mafia fraud attack as follows. First, a friend of Mallory approaches Alice while she is away from the office. At the same time, Mallory, who is in front of the door of Alice's office, uses a wireless device that pretends to be Alice's contactless token. All messages from the door are relayed by Mallory's wireless device to Mallory's friend, who also uses a wireless device to send these messages to the contactless token of Alice. Similarly, all messages from Alice's contactless token are relayed back to the door. Even though Alice nor her token is near the door, her relayed credentials will eventually be accepted by the door and Mallory will get access to Alice's office.

The most reliable countermeasures against proximity-based attacks, such as mafia and distance fraud [4], are *distance-bounding protocols*. These protocol employ mechanisms measuring the Round Trip Time (RTT) of a message exchange, i.e. the time a message takes to travel from a verifier to a prover and back [5]. If we denote the propagation speed of the communication channel by c, the round trip time by Δ_t and the processing time taken by the prover to send back the message by t_d, then the distance between the verifier and the prover is computed by the equation $d = c \times (\Delta_t - t_d)$.

The first design of a distance-bounding protocol based on RTT measurements dates back to 1993 [4]. Since then, more than 30 distance-bounding protocols have been proposed[1], each of them bringing improvements over their predecessors or adding new features. Amongst them, we can find a large class of protocols (e.g., [6, 7, 8, 9, 10, 11]) following two core principles raised by Hancke and Kuhn in [3].

- RTT measurements should exchange single-bit messages. This reduces the processing time by allowing the prover to instantly reply upon reception of a single bit message.
- Each RTT measurement ought to be based on a challenge-response authentication scheme so that, even if the protocol stops after a few RTT measurements, some guarantees of proximity can be provided.

Distance-bounding protocols adhering to the principles of

[1]`http://www.avoine.net/rfid/index.php`

Hancke and Kuhn's protocol (HK, see a representation in Figure 1) [3] normally consist of two phases. The first phase is the *slow phase*, where the verifier and the prover exchange nonces and use a shared key to secretly *pre-compute* a lookup table with potential responses for the next phase. The second phase, known as *fast phase*, consists of n RTT measurements. At the i-th RTT measurement round, the verifier sends a random bit-challenge c_i to the prover and starts a clock. The prover replies instantly to the challenge c_i by using the precomputed lookup table. Upon reception of the prover's reply, the verifier stops the clock and computes the RTT (Δ_i). The protocol finishes correctly if all responses are correct, and if $\Delta_i \leq \Delta$ for every $i \in \{1,\ldots,n\}$ given some time threshold Δ.

The simplicity of distance-bounding protocols based on the above scheme makes it attractive for ubiquitous wireless technologies such as RFID systems. However, to the best of our knowledge, all these protocols fall short in terms of resistance to mafia fraud attacks in comparison to cryptographically more expensive approaches, such as Brands and Chaum's protocol (BC) [4]. More precisely, given a fixed number of RTT measurements n, a mafia fraud attack to the BC protocol has probability of success $\frac{1}{2^n}$ [4], while no protocol following the above principles reduces the success probability of the same attack below $\frac{1}{2^n}(1+\frac{n}{2})$ [7]. This good performance of Avoine and Tchamkerten's protocol (Tree) [7] comes at the price of an exponential space requirement. It is not known if the same performance can be achieved at lower memory costs. Further, it is also an open question whether this lower bound can be reduced under the two core principles mentioned above.

Because all protocols based on these principles have a similar shape, it would be natural and useful to consider them as instantiations of the same protocol scheme, with slight variations. That would provide us with a mathematical model, allowing us to study theoretical properties that hold for a large class of protocols. In this paper, we propose such a model based on deterministic finite automata (DFAs). The contributions of this paper are summarized as follows:

- We propose an abstract model for distance-bounding protocols that use pre-computation and lookup operations. The proposed model is based on DFAs and captures several state-of-the-art distance-bounding protocols, such as [3, 6, 7, 10, 11, 12].
- Considering n to be the number of RTT measurements performed during a single execution of the protocol, we prove that $\frac{1}{2^n}(1+\frac{n}{2})$ is a tight lower bound on the security of this type of protocols against mafia fraud. This result indicates that, within our model, the Tree protocol [7] is optimal in terms of mafia fraud resistance.
- We study theoretical properties of a subclass of protocols within the proposed model, such as its resistance to pre-ask attacks and its space complexity.
- We define a novel family of protocols within our model that has good security properties in terms of resistance to mafia fraud in relation to its memory.

2 Related Work

2.1 Distance-Bounding Protocols

Distance-bounding protocols are authentication protocols that, in addition, compute an upper bound on the distance between the two parties involved in the protocol. The distance estimation process relies on a concrete physical law, stating that Radio Frequency (RF) signals travel at the speed of light. By using a communication channel whose propagation speed is close to the physical limit, a distance-bounding protocol ensures that an adversary interfering in an RTT measurement by relaying messages cannot decrease the estimated distance. In the same vein, most distance-bounding protocols aim to minimize the processing time on the prover side. A small and deterministic processing time will improve the precision of the distance estimation and prevents attacks where the adversary overclocks the prover [13, 4, 14].

A large variety of distance-bounding protocols exist. Some are based on expensive cryptographic operations such as signatures and commitment schemes [4, 15, 16]. Others use error detection and correction techniques to deal with potential noise during the RTT measurements [17]. We can even find approaches addressing location-privacy concerns [18]. Nevertheless, there are common features shared by most distance-bounding protocols. For example, they typically exchange 1-bit messages for RTT measurements, which reduces the processing time by allowing the prover to instantly reply upon reception of a single bit message. Exceptions to this rule are, for example, Munilla and Peinado's protocol [19] and its generalization [20].

Another characteristic that clearly splits the set of distance-bounding protocols into two classes is the presence or not of a so-called *final slow phase*. We recall that a fast phase consists of consecutive message exchanges intended for RTT measurement, while a slow phase is formed by any other type of message. Hence, a slow phase is said to be final if it represents the end of the protocol execution. It was in 2005, twelve years after the pioneering Brands and Chaum's protocol (BC) [4], when Hancke and Kuhn proposed the first distance-bounding protocol without final slow phase [3]. Given its relevance and impact on many recent protocols, we next detail this protocol in extenso (see also Figure 1).

Hancke and Kuhn's protocol (HK) consists of a slow phase followed by a fast phase. In the slow phase, the verifier and the prover exchange nonces $N_{\mathcal{V}}$ and $N_{\mathcal{P}}$ and use a keyed pseudo-random function (PRF) to agree on a $2n$-bit sequence $B = b_1 \ldots b_{2n}$, where n is a parameter representing the number of RTT measurements. The fast phase consists of n consecutive rounds. At the i-th round, the verifier sends a random challenge bit c_i to the prover and starts a clock. Upon reception of c_i, the prover replies with the bit b_{2i+c_i-1}. Immediately after receiving the prover's answer, the verifier stops the clock and computes the RTT, $\Delta_i = t_f - t_s$. The protocol finishes correctly if all responses are correct according to the challenges and bit-sequence B, and if $\Delta_i \leq \Delta$ for every $i \in \{1,\ldots,n\}$ given some time threshold Δ.

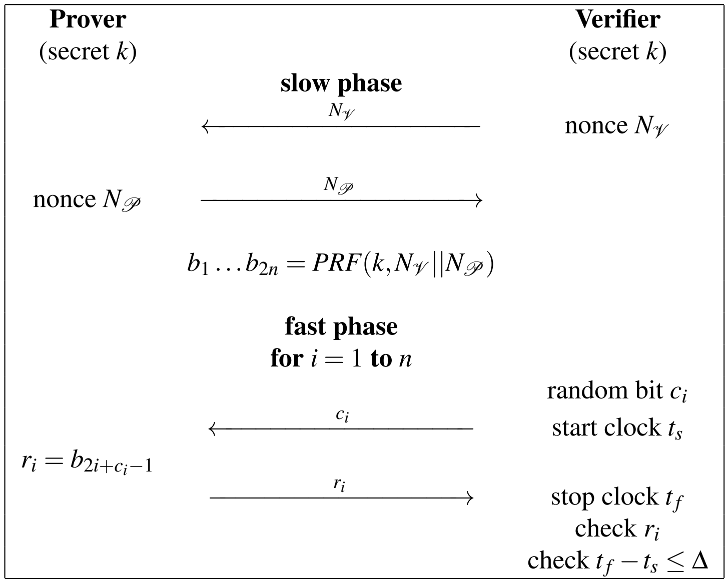

Figure 1: The HK protocol, by Hancke and Kuhn [3].

Because Hancke and Kuhn's protocol relies on a single pseudo-random function, it is considered to be the first distance-bounding protocol suitable for resource-constrained technologies such as RFID systems. An additional feature of this protocol is that, during the fast phase, the prover computes the correct reply to the verifier's challenge via a simple lookup operation. This significantly reduces the processing time, ergo it sticks to the basic principles of RTT measurements. The drawback, however, of the HK protocol is its low resistance to mafia fraud. An adversary could execute a so-called *pre-ask* [21] attack as follows. First, the adversary relays all the communication between the prover and the verifier during the slow phase. Before the beginning of the fast phase, the adversary claims to be the legitimate verifier and queries the prover n times with the same challenge 0. As a result, the adversary receives from the prover the values $b_1, b_3, \dots, b_{2n-1}$. Finally, the adversary uses this knowledge to reply to the verifier's challenges during the fast phase. The probability of success of an adversary executing such an attack is $(3/4)^n$. This can be easily seen as follows. At every round of the fast phase, the adversary is challenged with a random bit. If this bit is a 0, she will answer with the correct (pre-asked) reply. Otherwise, if the challenge is 1, she can reply with a random bit, which gives her a chance of $\frac{1}{2}$ to reply correctly. In total, this gives her a success probability of $\frac{1}{2} \times 1 + \frac{1}{2} \times \frac{1}{2} = \frac{3}{4}$ per round, leading to $(3/4)^n$ for n rounds. For an ideal protocol, the adversary would not be able to achieve any advantage over randomly guessing the reply, leading to a success probability of $(1/2)^n$. This probability is achieved by Brands and Chaum's protocol (BC) [4], which uses a final slow phase for authentication of the exchanged bits.

In [3], Hancke and Kuhn explain the advantage of avoiding a final slow phase, even at the cost of an apparent decrease in the resistance to mafia fraud, as follows. In terms of execution time and computational complexity, the cost of executing a couple of additional rounds during the fast phase is significantly lower than the cost of performing expensive cryptographic operations and message exchanges over a traditional communication channel. Hence, for practical values of n, there exists $m > n$ such that the HK protocol with m rounds is more efficient than the BC protocol with n rounds, i.e., $\left(\frac{3}{4}\right)^m < \left(\frac{1}{2}\right)^n$.

The precomputation approach started by Hancke and Kuhn has been extended and improved by many recent distance-bounding protocols [6, 7, 8, 9, 10, 11], which we refer to as HK-like protocols. Even though they all have their own peculiarities, most of them perform simple lookup operations during the fast phase in order to reply to the verifier's challenges. Exceptions to this rule are the protocols introduced in [8], which uses XOR operations, and [9], which requires the prover to generate random bits during the fast phase.

To the best of our knowledge, the best HK-like protocol in terms of resistance to mafia fraud is the Tree-based protocol (Tree) proposed by Avoine and Tchamkerten [7] in 2009. Their protocol considers an edge-labeled full-binary tree of depth n with labels taken from the set $\{0, 1\}$, and satisfying that every two edges with a common parent node have different labels. Thus, any bit-sequence $c_1 c_2 \dots c_i$ with $1 \leq i \leq n$, defines a unique path in the tree. At each session, the prover and the verifier securely agree on a vertex-labeling function over the considered tree by using two nonces and a secret key as input to a pseudo-random function. As in Hancke and Kuhn's protocol, the fast phase consists of n challenges $c_1, \dots, c_n$. The prover replies to these challenges with the node's label of the unique path defined by $c_1 \dots c_n$. A sketch of this protocol is shown in Figure 2.

In the Tree protocol, the probability of success of an adversary executing a pre-ask attack is $\frac{1}{2^n}(1 + \frac{n}{2})$ [7], which has not been improved yet by any HK-like protocol. The problem is, however, that pre-computing a full-binary tree of depth n is ex-

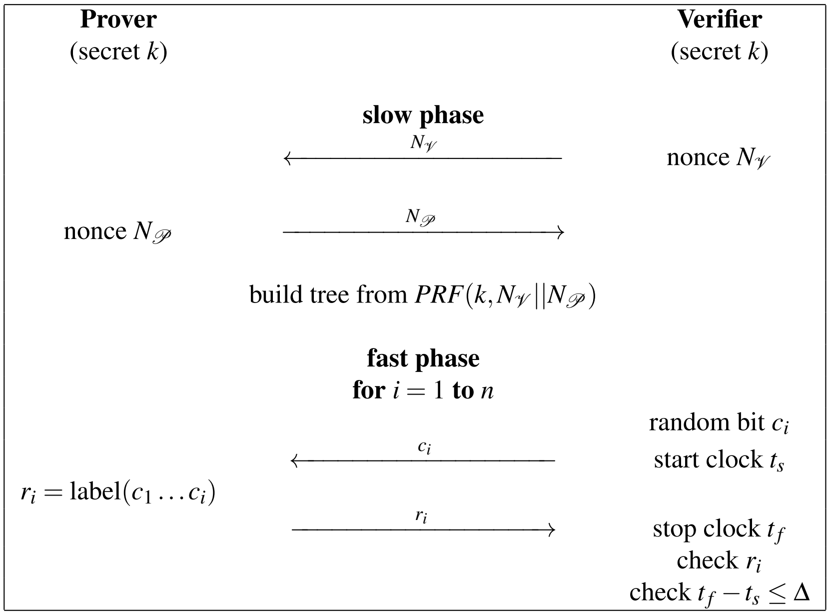

Figure 2: The Tree protocol, by Avoine and Tchamkerten [7].

ponential in terms of n. Avoine and Tchamkerten propose to split the tree into several trees to overcome this problem [7], while in [6] the tree-based approach is generalized to graphs of arbitrary size. From a decision-making point of view and according to the framework proposed in [22], graph-based protocols provide relevant features not provided by other types of distance-bounding protocols.

The formal model proposed in this article actually captures the notion of graph-based distance-bounding protocols, as introduced in [6, 23]. By studying the proposed model we solve the open question of whether there exists a graph-based protocol with security $\frac{1}{2^n}(1+\frac{n}{2})$ against a pre-ask attack and whose graph contains a polynomial number of vertices in terms of n. It is worth remarking that our model is not a simple abstraction and formalization of the notion of graph-based distance-bounding. As we show later, there exists a novel class of distance-bounding protocols within our model with resistance to mafia fraud asymptotically close to $\frac{1}{2^n}(1+\frac{n}{2})$.

2.2 Modeling Distance-Bounding Protocols

In 2010, Kara *et al.* [24] defined a class of distance-bounding protocols named *current challenge-dependent protocol*. Similar to the class of protocols that we consider in this article, *current challenge-dependent protocols* do not have a final slow phase. However, there are two main aspects that make Kara *et al.*'s model significantly different from ours. First, they consider stateless protocols only, in the sense that all RTT measurements are independent of previous ones. Therefore, their model cannot capture, for example, the class of graph-based distance-bounding protocols, which is captured by our model. Second, Kara *et al.*'s model allows protocols to perform arbitrary computation during the fast phase, as long as this computation depends exclusively on the current challenge and the *session secrets* [24]. Differently, we make the requirement explicit that only lookup operations are allowed, implying that there exist protocols within Kara *et al.*'s model that are not captured by ours.

3 A Model Based on Deterministic Finite Automata

In this section we provide a simple model that captures a prominent class of distance-bounding protocols, for which the following two properties hold:

- The responses to the fast phase challenges are a the result of lookup operation on a table built prior to fast phase. The entry for the lookup is is the fast phase challenge.
- They do not have a final crypto-based verification phase.

We model distance-bounding protocols via a particular class of Deterministic Finite Automata (DFA).

Definition 1 (State-Labeled DFA)**.** *A State-Labeled Deterministic Finite Automata is a tuple of the form* $(\Sigma, \Gamma, Q, q_0, \delta, \ell)$ *where:*

- Σ *is a finite set of input symbols,*
- Γ *is a finite set of output symbols*
- Q *is a finite set of states,*
- $q_0 \in Q$ *is the initial state,*
- $\delta : Q \times \Sigma \to Q$ *is a state-transition function,*

- $\ell\colon Q \to \Gamma$ *is a labeling function on the states.*

Definition 1 above differs from traditional DFAs in two main aspects. First, it does not define final states. This is because we use it for modeling the execution of a security protocol, which might halt at any state. Second, it includes a labeling function on the states whose output ranges over the set of output symbols Γ. While transition labels will be used to express the challenges exchanged in the protocol, the state labels will define the corresponding responses. We will make this precise in Definition 5 below.

Similar to traditional DFAs, we assume that the state-transition function is total. When defining or drawing DFAs we will only specify the relevant transitions and, again similar to normal DFAs, we assume that specifications are completed with an implicit *trap state* that serves as the target state for all transitions that are not shown.

In what follows, we denote the i-th symbol of the string c by c_i. We also use the terms *string* and *sequence* interchangeably. Also, we will say simply *automaton* to refer to an State-Labeled DFA.

Definition 2 (Generalized Transition Function)**.** *Given an automaton* $(\Sigma, \Gamma, Q, q_0, \delta, \ell)$*, its* generalized transition function $\hat{\delta}\colon \Sigma^* \to Q$ *is defined as below, where* ε *represents the empty string:*

$$\hat{\delta}(c) = \begin{cases} q_0 & \text{if } c = \varepsilon, \\ \delta(\hat{\delta}(c_1 \dots c_{n-1}), c_n) & \text{if } c = c_1 \dots c_n. \end{cases}$$

Definition 3 (Generalized Labeling Function)**.** *Given an automaton* $(\Sigma, \Gamma, Q, q_0, \delta, \ell)$*, its* generalized labeling function $\hat{\ell}\colon \Sigma^* \to \Gamma$ *is defined by* $\hat{\ell}(x) = \ell(\hat{\delta}(x))$.

Given two sets Σ and Γ, we use $\mathbb{U}_{\Sigma,\Gamma}$ to denote the universe of all automata with Σ and Γ as input and output symbol sets, respectively.

Definition 4 (Lookup-Based Protocol)**.** *A* lookup-based protocol *is a finite set* $P \subseteq \mathbb{U}_{\Sigma,\Gamma}$*, for some sets* Σ *and* Γ.

We model a protocol as a set of automata, where each automaton describes the protocol's behaviour in the fast phase. The input symbols of the automaton are the challenges and the output symbols are the corresponding responses. The structure and labeling of such an automaton follows from the calculations in the slow phase, in which, e.g., the nonces are chosen. Consequently, every possible outcome of the slow phase results in an automaton, so the number of automata in the set that describes the protocol is equal to the number of different outcomes of the slow phase. The execution of a protocol therefore consists of the (random) selection of one of the automata (the slow phase) and a run of this automaton consisting of an alternation of input and output symbols (the fast phase).

As a running example, let's consider Hancke and Kuhn's protocol [3]. We recall that this protocol (see Figure 1) precomputes a $2n$-bit sequence, and all the prover's responses are based on lookup operations on such sequence.

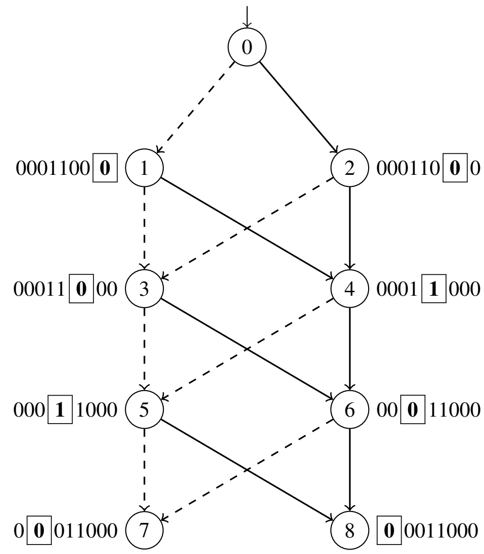

Figure 3: Automaton A_{24} for the HK protocol with 4 rounds. The binary sequence 00011000 besides the states stands for 24 in binary. Dashed and solid arrows denote transitions labeled with 0 and 1, respectively.

Example 1 (Hancke and Kuhn's protocol)**.** *The* HK *protocol is the set* $\{A_0, A_1, \dots, A_{2^{2n}-1}\}$ *such that* $A_i = (\Sigma, \Gamma, Q, q_0, \delta, \ell_i)$ *for every* $i \in \{0, \dots, 2^{2n} - 1\}$ *where:*

- $\Sigma = \Gamma = \{0,1\}$, $Q = \{0, 1, \dots, 2n\}$, $q_0 = 0$.
- $\delta(q,c) = \begin{cases} q + c + 1 & \text{if } q \text{ is even} \\ q + c + 2 & \text{otherwise.} \end{cases}$
- $\ell_i(q)$ *is the q-th least significant bit of i in the binary representation of i.*

The number 2^{2n} of automata used to represent the HK protocol in Example 1 is the number of all possible $2n$-bit sequences that can be pre-computed according to the protocol. In the example, we mapped each bit-sequence $b_1 \dots b_{2n} \in \{0,1\}^{2n}$ onto the automaton A_i where i equals to $b_1 \dots b_{2n}$ in decimal. Figure 3 shows a graphical representation of the automaton A_{24}, which represents the HK protocol with $n = 4$ and $b_1 \dots b_{2n} =$ 00011000. In this example, the states 4 and 5 are labeled with 1 because this is the value of both b_4 and b_5, while the remaining states are labeled with 0.

Definition 5 (Execution Model)**.** *Let n be positive integer number and P a lookup-based protocol. The triple* $(A, C, R) \in P \times \Sigma^n \times \Gamma^n$ *is a* correct execution of P with n rounds*, denoted* $(A, C, R) \sqsubset P$*, if* $r_i = \hat{\ell}(c_1 \dots c_i)$ *for all* $i \in \{1, \dots, n\}$.

The intuition behind the proposed execution model is the following. Before the start of the fast phase, the prover and the verifier agree on a *fresh* automaton A. Then in the fast phase, the verifier sends n challenges $c_1 \dots c_n$ and expects to receive as replies the sequence $\hat{\ell}(c_1) \dots \hat{\ell}(c_1 \dots c_n)$. As an example, let us consider again automaton A_{24} depicted in Figure 3. Given the input bit sequence 1100, this automaton transits over the states $2,4,5$, and 7, whose labels are 0, 1, 1, and 0, respectively. Hence, $(A_{24}, 1100, 0110)$ is a correct execution of the HK protocol with 4 rounds.

4 Properties of Lookup-Based Protocols

In this section we investigate relevant properties of lookup-based protocols. First, we make our adversarial model explicit and provide a definition of resistance to mafia fraud. We also prove that there does not exist a lookup-based protocol whose resistance to mafia fraud is less than $\frac{1}{2^n}(1+\frac{n}{2})$. This result implies that the Tree protocol [7] is optimal in terms of resistance to mafia fraud. Further, we prove a property on the probability distribution of labels in an optimal lookup-based protocol.

4.1 Mafia Fraud Resistance

As shown in [21], the best-known adversary strategy to perform a mafia fraud attack against distance-bounding protocols without final slow phase is the pre-ask attack. In this attack, the adversary relays all the communication between the prover and the verifier during the slow phase. Before the beginning of the fast phase, the adversary claims to be the legitimate verifier and queries the prover n times with a sequence of challenges. The responses from the prover to these challenges are used by the adversary to later execute the fast phase with the legitimate verifier. Below, we make this intuitive definition formal, where $\mathscr{F}$ denotes the universe of all sets F of functions $\{f_1, \ldots, f_n\}$ such that $f_i \colon \Sigma^i \times \Sigma^n \times \Gamma^n \to \Gamma$ for all $i \in \{1, \ldots, n\}$.

Definition 6 (Pre-ask Attack Success Probability)**.** *Let P be a lookup-based protocol. For every $F \in \mathscr{F}$ and every $x \in \Sigma^n$, let S_F^x be the event that $(A, c_1 \ldots c_n, z_1 \ldots z_n) \sqsubset P$ given that:*

- *A is a random automaton in P, and*
- *c is a random sequence in Σ^n, and*
- *$y \in \Gamma^n$ such that $(A,x,y) \sqsubset P$, and*
- *$z_i = f_i(c_1 \ldots c_i, x, y)$ for all $i \in \{1, \ldots, n\}$.*

Then the success probability of a pre-ask attack against P *is computed by:*

$$preask(P) = \max_{F \in \mathscr{F}, x \in \Sigma^n} \{\Pr(S_F^x)\}. \tag{1}$$

In Definition 6, the adversary knowledge is a correct execution (A,x,y) of P where x is chosen by the adversary and A is randomly chosen in P. That is, the adversary is able to query the prover with challenges x and receive the corresponding answers y. With this knowledge, the adversary defines a strategy to answer to the verifier's challenge. We represent such a strategy as a set of functions $F = \{f_1, \ldots, f_n\}$. Given a sequence of challenges $c_1 \ldots c_i$ for some $i \in \{1, \ldots, i\}$, the adversary's answer at the i-th round is uniquely determined by $f_i(c_1 \ldots c_i, x, y)$. This makes the assumption explicit that challenges are unpredictable and that the adversary replies immediately upon reception of a challenge.

4.2 Properties of Optimal Lookup-Based Protocols

As stated before, the adversary's success probability when performing mafia fraud against the Tree protocol is $\frac{1}{2^n}\left(1+\frac{n}{2}\right)$. We will prove that $\frac{1}{2^n}\left(1+\frac{n}{2}\right)$ is indeed optimal, i.e., there does not exist a lookup-based DB protocol whose resistance to mafia fraud is lower than $\frac{1}{2^n}\left(1+\frac{n}{2}\right)$. For the complexity analysis we assume the protocol to have a binary set of input and output symbols, as is the case in most existing distance-bounding protocols.

Theorem 1. *For every protocol $P \subseteq \mathbb{U}_{\{0,1\},\{0,1\}}$ with $n > 0$ the probability of success of a mafia fraud attack is tightly lower-bounded by $\frac{1}{2^n}\left(1+\frac{n}{2}\right)$.*

Proof. We define the following adversary strategy to execute a pre-ask attack as in Definition 6. We define $F = \{f_1, \ldots, f_n\} \in \mathscr{F}$ as follows:

- $f_i(c,x,y) = y_i$ if $c_1 \ldots c_i = x_1 \ldots x_i$,
- $f_i(c,x,y)$ is assigned with a random bit, otherwise.

According to the above strategy, at the i-th round the adversary replies randomly unless the adversary has guessed all the verifier's challenges till the i-th round.

We will first prove that $preask(P) \geq \frac{1}{2^n}\left(1+\frac{n}{2}\right)$. Indeed, let $x \in \{0,1\}^n$. We will proceed to compute the probability $\Pr(S_F^x)$, recall the event S_F^x from Definition 6. For a random $c \in \{0,1\}^n$, let M_i be the event that $c_1 \ldots c_i = x_1 \ldots x_{i-1}$ and that $c_i \neq x_i$ for every $i \in \{1, \ldots, n\}$. Then from the law of total probability we have:

$$\Pr(S_F^x) = \sum_{i=1}^{n} \Pr(F \mid M_i)\Pr(M_i) + \Pr(S_F^x \mid c = x)\Pr(c = x).$$

Hence, given that $\Pr(M_i) = \frac{1}{2^i}$ and $\Pr(c = x) = \frac{1}{2^n}$, we derive:

$$\Pr(S_F^x) = \sum_{i=1}^{n} \Pr(S_F^x \mid M_i) \cdot \frac{1}{2^i} + \Pr(S_F^x \mid c = x) \cdot \frac{1}{2^n}. \tag{2}$$

Now, because $\Pr(S_F^x \mid c = x) = 1$, Equation 2 gives:

$$\Pr(S_F^x) = \sum_{i=1}^{n} \Pr(S_F^x \mid M_i) \cdot \frac{1}{2^i} + \frac{1}{2^n}. \tag{3}$$

So far we have not used the particularity of F. Let's do so now, which gives us $\Pr(S_F^x \mid M_i) = 1/2^{n-i+1}$, because starting from the i-th round the adversary always replies randomly. Therefore, by applying this result in Equation 3 we obtain:

$$\Pr(S_F^x) = \sum_{i=1}^{n} \frac{1}{2^{n-i+1}} \cdot \frac{1}{2^i} + \frac{1}{2^n} = \frac{1}{2^n}\left(1+\frac{n}{2}\right).$$

We conclude this proof by remarking that this lower bound is tight, because it is realized by the Tree protocol [7]. □

Definition 7 (Optimal Lookup-Based Protocol)**.** *A lookup-based protocol for $n > 0$ rounds is* optimal *if its mafia fraud resistance is* $\frac{1}{2^n}\left(1+\frac{n}{2}\right)$.

Next we proceed to prove a necessary condition for optimal lookup-based protocols. This condition establishes that, given an input sequence x and a lookup-based protocol P, the labels assigned to the states reachable by x uniformly distribute in P.

Lemma 1. *Let $P \subseteq \mathbb{U}_{\{0,1\},\{0,1\}}$ be an optimal protocol with $n > 0$ rounds. For every $i \in \{1,\dots,n\}$ and every $x \in \{0,1\}^i$, define E_0^x (resp. E_1^x) the event that given a random automaton $(\Sigma,\Gamma,Q,q_0,\delta,\ell) \in P$ it holds that $\hat{\ell}(x) = 0$ (resp. $\hat{\ell}(x) = 1$). Then, $\Pr(E_0^x) = \Pr(E_1^x) = \frac{1}{2}$ for all $i \in \{1,\dots,n\}$ and all $x \in \{0,1\}^i$.*

Proof. Let's assume $j \in \{1,\dots,n\}$ and $\bar{x} \in \{0,1\}^j$ exist such that $\Pr(E_0^{\bar{x}}) \neq \frac{1}{2}$. We define the following adversary strategy $F = \{f_1,\dots,f_n\} \in \mathscr{F}$ to execute a pre-ask attack as in Definition 6:

- $f_i(c,x,y) = y_i$ if $c_1 \dots c_i = x_1 \dots x_i$,
- $f_i(c,x,y) = 0$ if $i = j$ and $c_1 \dots c_i = \bar{x}_1 \dots \bar{x}_i \neq x_1 \dots x_i$ and $\Pr(E_0^{\bar{x}}) > \frac{1}{2}$,
- $f_i(c,x,y) = 0$ if $i = j$ and $c_1 \dots c_i = \bar{x}_1 \dots \bar{x}_i \neq x_1 \dots x_i$ and $\Pr(E_0^{\bar{x}}) < \frac{1}{2}$,
- $f_i(c,x,y)$ is assigned with a random bit, otherwise.

Let $x \in \{0,1\}^n$. As in Theorem 1's proof, for a random $c \in \{0,1\}^n$ we consider M_i to be the event that $c_1 \dots c_{i-1} = x_1 \dots x_{i-1}$ and $c_i \neq x_i$ for every $i \in \{1,\dots,n\}$. We proceed as in Theorem 1's proof until Equation 3:

$$\Pr(S_F^x) = \sum_{i=1}^{n} \Pr(S_F^x \mid M_i) \cdot \frac{1}{2^i} + \frac{1}{2^n}. \tag{4}$$

From the adversary's strategy we obtain that , for every $i \in \{1,\dots,n\}$, $\Pr(S_F^x \mid M_i) = 1/2^{n-i+1}$ unless $c_1 \dots c_i = \bar{x}_1 \dots \bar{x}_i \neq x_1 \dots x_i$. In this case, the probability of success of the adversary at the j-th round is $\Pr(E_0^{\bar{x}})$ if $\Pr(E_0^{\bar{x}}) > \frac{1}{2}$ or otherwise $1 - \Pr(E_0^{\bar{x}})$. Therefore, we obtain that $\Pr(S_F^x \mid M_i) > 1/2^{n-i+1}$ if $c_1 \dots c_j = \bar{x}_1 \dots \bar{x}_j \neq x_1 \dots x_j$ or otherwise $\Pr(S_F^x \mid M_i) = 1/2^{n-i+1}$. This implies that $\Pr(S_F^x \mid M_i) > 1/2^{n-i+1}$ and by applying this result to Equation 4 we have:

$$\Pr(S_F^x) > \sum_{i=1}^{n} \frac{1}{2^{n-i+1}} \cdot \frac{1}{2^i} + \frac{1}{2^n} = \frac{1}{2^n}\left(1 + \frac{n}{2}\right). \tag{5}$$

Equation 5 contradicts the assumption that P is optimal, which proves that $\Pr(E_0^{\bar{x}}) = \frac{1}{2}$. Analogously, we can obtain that $\Pr(E_1^{\bar{x}}) = \frac{1}{2}$. □

4.3 Layered and Random-Labeled Protocols

Based on Lemma 1, we observe that the labeling function of the DFAs in a protocol plays an important role in the protocol's resistance to pre-ask attacks. We thus define the notion of *random-labeling*, whose intent is to capture protocols with maximum uncertainty on the labels of the states.

Definition 8 (Random-Labeled)**.** *A lookup-based protocol P is* random-labeled *if for every $(\Sigma,\Gamma,Q,q_0,\delta,\ell) \in P$ and for every labeling function $\ell' \colon Q \to \Gamma$, the automaton $(\Sigma,\Gamma,Q,q_0,\delta,\ell')$ is also in P.*

A random-labeled lookup-based protocol P accounts for all possible labeling functions that can be defined on a set of states. This property is indeed satisfied by most existing distance-bounding protocols. We show next that being random-labeled is a sufficient condition to satisfy the implication of Lemma 1.

Another property of lookup-based protocols is that of being *layered*. Intuitively, in a layered lookup-based protocol with n rounds we can partition the set of states of every automaton into n subsets (layers), in such a way that all states within a layer are only reachable by input sequences of the same size.

Definition 9 (Layered)**.** *A lookup-based protocol P is* layered *if $\hat{\delta}(x) = \hat{\delta}(y) \implies |x| = |y|$ for every $(\Sigma,\Gamma,Q,q_0,\delta,\ell) \in P$ and every pair $x, y \in \Sigma^*$.*

We observe that all existing lookup-based protocols, except Poulidor [6], are layered and random-labeled. Figure 3 clearly shows that the example automaton of the HK protocol is layered, because the states of the i-th layer, i.e. $2i-1$ and $2i$, with $0 < i \leq 4$, can only be reached by an input sequence of length i. The labeling function, i.e., the association from states to bits, is composed in a random way. In the remaining of this paper we will thus focus on the analysis of layered and random-labeled lookup-based protocols.

5 Pre-Ask Attacks in Layered and Random-Labeled Protocols

In this section we provide an optimal adversary strategy to execute a pre-ask attack in a layered and random-labeled lookup-based protocol. It turns out, as expressed in Theorem 2 below, that such a strategy is simply to reply to the verifier challenges exactly the same responses as those obtained from the prover in the pre-ask session.

Theorem 2. *Let $P \subseteq \mathbb{U}_{\{0,1\},\{0,1\}}$ be a layered and random-labeled lookup-based protocol with $n > 0$ rounds and let $x \in \{0,1\}^n$. Consider the pre-ask strategy $F = \{f_1,\dots,f_n\} \in \mathscr{F}$ defined by $f_i(c,x,y) = y_i$ for all $y \in \Gamma^i$ and $i \in \{1,\dots,n\}$ and $c \in \Sigma^i$. Let $G \in \mathscr{F}$ and consider the events S_F^x and S_G^x (as in Definition 6) that the strategies F and G succeed, respectively. Then $\Pr(S_F^x) \geq \Pr(S_G^x)$.*

Proof. For every $y \in \Gamma^n$ we denote $P^{x,y} \subseteq P$ as a subset of P defined by:

$$A \in P^{x,y} \iff (A,x,y) \sqsubset P.$$

That is, $P^{x,y}$ is the set of all possible automata in P that produce $y_1 \ldots y_n$ as the corresponding responses for the challenges $x_1 \ldots x_n$.

For every $i \in \{1, \ldots, n\}$, we denote $S_F^{x_1 \ldots x_i}$ the event that the adversary replies correctly to the first i verifier's challenges by using the pre-ask strategy F. Formally, $S_F^{x_1 \ldots x_i}$ is the event that, given a random automaton $A = (\Sigma, \Gamma, Q, q_0, \delta, \ell)$, a correct execution (A, x, y), and a random sequence $c_1 \ldots c_i$, it holds that $f_j(c_1 \ldots c_j, x, y) = \hat{\ell}(c_1 \ldots c_j)$ for all $j \in \{1, \ldots, i\}$. Analogously, we define the events $S_G^{x_1 \ldots x_1}$ for the pre-ask strategy G.

Because the verifier's challenges c and the automaton A are randomly chosen at each execution of the protocol we can compute $\Pr\left(S_F^{x_1 \ldots x_i}\right)$ as follows:

$$\Pr\left(S_F^{x_1 \ldots x_i}\right) = \frac{1}{2^i |P^{x,y}|} \times \sum_{c \in \{0,1\}^i, A \in P^{x,y}} h_i(A, c, F), \quad (6)$$

where $|P^{x,y}|$ stands for the cardinality of $P^{x,y}$, and h_i is defined by:

$$h_i(A, c, F) = \begin{cases} 1 & \text{if } \substack{\forall j \in \{1, \ldots, i\}. \\ \hat{\ell}(c_1 \ldots c_j) = f_j(c_1 \ldots c_j, x, y)} \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

For every $i \in \{1, \ldots, n\}$, we define $h_i(A, c, G)$ and compute $\Pr\left(S_G^{x_1 \ldots x_i}\right)$ analogously to Equations 7 and 6. We will proceed by induction to prove that $\Pr\left(S_F^{x_1 \ldots x_i}\right) \geq \Pr\left(S_G^{x_1 \ldots x_i}\right)$ for all $i \in \{1, \ldots, n\}$.

For every $c \in \Sigma^n$, let $P_c^{x,y}$ be the subset of $P^{x,y}$ defined by $(\Sigma, \Gamma, Q, q_0, \delta, \ell) \in P_c^{x,y} \iff \hat{\delta}(c) \neq \hat{\delta}(x)$. We also consider a relation $\mathscr{R}_c \subseteq P_c^{x,y} \times P_c^{x,y}$ defined by $(A, A') \in \mathscr{R}_c$ if and only if:

- $A = (\Sigma, \Gamma, Q, q_0, \delta, \ell)$ and $A' \in (\Sigma, \Gamma, Q, q_0, \delta, \ell')$ for some q_0, δ, ℓ and ℓ', and
- $\ell(\hat{\delta}(c)) \neq \ell'(\hat{\delta}(c))$, and
- $\forall q \in Q.\ q \neq \hat{\delta}(c) \implies \ell(q) = \ell'(q)$.

Note that $\mathscr{R}_c$ is symmetric and bijective in $P_c^{x,y}$, given that P is random-labeled. We continue the proof by induction.

- *Base case:* $\Pr\left(S_F^{x_1}\right) \geq \Pr\left(S_G^{x_1}\right)$.

  Consider $A = (\Sigma, \Gamma, Q, q_0, \delta, \ell) \in P$, and $c \in \{0,1\}$, and $y \in \{0,1\}$ such that $(A, x, y) \sqsubset P$. If $A \notin P_c^{x,y}$ then $\hat{\delta}(c) = \hat{\delta}(x)$ and consequently $f_1(c, x, y) = \hat{\ell}(c)$ and $h_1(A, c, F) = 1 \geq h_1(A, c, G)$. If $A \in P_c^{x,y}$ then $h_1(A, c, F) + h_1(A', c, F) = h_1(A, c, G) + h_1(A', c, G) = 1$ where $A' \in P$ and $(A, A') \in \mathscr{R}_c$. Note that any pre-ask strategy fails either in A or in A' because $\ell(\hat{\delta}(c)) \neq \ell'(\hat{\delta}(c))$. This gives us:

  $$\sum_{(A,A') \in \mathscr{R}_c} h_1(A, c, F) + h_1(A', c, F) = 2 \sum_{A \in P_c^{x,y}} h_1(A, c, F)$$
  $$\sum_{(A,A') \in \mathscr{R}_c} h_1(A, c, G) + h_1(A', c, G) = 2 \sum_{A \in P_c^{x,y}} h_1(A, c, G)$$

  for every $c \in \{0,1\}$. Therefore, we conclude that:

  $$\sum_{c \in \{0,1\}, A \in P^{x,y}} h_1(A, c, F) \geq \sum_{c \in \{0,1\}, A \in P^{x,y}} h_1(A, c, G)$$

  which means that $\Pr\left(S_F^{x_1}\right) \geq \Pr\left(S_G^{x_1}\right)$.

- *Hypothesis:* $\Pr\left(S_F^{x_1 \ldots x_{n-1}}\right) \geq \Pr\left(S_G^{x_1 \ldots x_{n-1}}\right)$.
- *Thesis:* To prove $\Pr\left(S_F^{x_1 \ldots x_n}\right) \geq \Pr\left(S_G^{x_1 \ldots x_n}\right)$ we first notice the following:

  $$h_n(A, c, F) = h_{n-1}(A, c_1 \ldots c_{n-1}, F) \times \begin{cases} 1 & \text{if } f_n(c, x, y) = \hat{\ell}(c_1 \ldots c_n) \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

  for every $A = (\Sigma, \Gamma, Q, q_0, \delta, \ell) \in P$ and every $c \in \{0,1\}^n$. We analogously write $h_n(A, c, G)$ in terms of $h_{n-1}(A, c_1 \ldots c_{n-1}, G)$.

  As in the base case, for each automaton $A = (\Sigma, \Gamma, Q, q_0, \delta, \ell) \in P$ and $c \in \{0,1\}^n$, we split our analysis depending on whether $A \in P^{x,y}$ or not as follows: If $A \notin P_c^{x,y}$ then $f_n(c, x, y) = \hat{\ell}(c)$ and $h_n(A, c, F) = h_{n-1}(A, c_1 \ldots c_{n-1}, F)$. Given that by definition $h_n(A, c, G) \leq h_{n-1}(A, c_1 \ldots c_{n-1}, G)$, we obtain:

  $$\forall A \in P^{x,y} \setminus P_c^{x,y}. \quad h_n(A, c, F) - h_n(A, c, G) \geq h_{n-1}(A, c_1 \ldots c_{n-1}, F) - h_{n-1}(A, c_1 \ldots c_{n-1}, G). \quad (9)$$

  If $A \in P_c^{x,y}$ then consider $A' = (\Sigma, \Gamma, Q, q_0, \delta, \ell')$ such that $(A, A') \in \mathscr{R}_c$. We note that state $\hat{\delta}(c)$ is unreachable by sequences of size smaller than n, because P is layered. On the other hand, $\forall q \in Q \setminus \{\hat{\delta}(c)\}.\ \ell(q) = \ell'(q)$. Consequently:

  $$h_{n-1}(A, c_1 \ldots c_{n-1}, F) = h_{n-1}(A', c_1 \ldots c_{n-1}, F),$$
  $$h_{n-1}(A, c_1 \ldots c_{n-1}, G) = h_{n-1}(A', c_1 \ldots c_{n-1}, G). \quad (10)$$

  As in the base case, we observe that any pre-ask strategy fails either in A or in A', given that $\ell(\hat{\delta}(c)) \neq \ell'(\hat{\delta}(c))$. This observation and Equation 10 lead to the following results:

  $$\sum_{A \in P_c^{x,y}} h_n(A, c, F) + h_n(A', c, F) = 2h_{n-1}(A, c_1 \ldots c_{n-1}, F)$$
  $$\sum_{A \in P_c^{x,y}} h_n(A, c, G) + h_n(A', c, G) = 2h_{n-1}(A, c_1 \ldots c_{n-1}, G).$$

  Hence, because $\mathscr{R}_c$ is bijective and symmetric we obtain:

  $$\sum_{A \in P_c^{x,y}} h_n(A, c, F) = h_{n-1}(A, c_1 \ldots c_{n-1}, F),$$
  $$\sum_{A \in P_c^{x,y}} h_n(A, c, G) = h_{n-1}(A, c_1 \ldots c_{n-1}, G). \quad (11)$$

  Hence, Equations 9 and 11 together give:

  $$\sum_{c \in \{0,1\}^n, A \in P^{x,y}} h_n(A, c, F) - h_n(A, c, G) \geq \sum_{c \in \{0,1\}^n, A \in P^{x,y}} \big(h_{n-1}(A, c_1 \ldots c_{n-1}, F) - h_{n-1}(A, c_1 \ldots c_{n-1}, G)\big). \quad (12)$$

Finally, from the induction hypothesis, we have that the right-hand side of Equation 12 is non-negative which gives us $\sum_{c\in\{0,1\}^n, A\in P^{x,y}} h_n(A,c,F) - h_n(A,c,G) \geq 0$ which consequently means that $\Pr\left(S_F^{x_1\ldots x_n}\right) \geq \Pr\left(S_G^{x_1\ldots x_n}\right)$.

□

6 Protocols Arbitrarily Close to Optimal

We have proven that $\frac{1}{2^n}\left(1+\frac{n}{2}\right)$ is a tight lower bound on the resistance to mafia fraud of lookup-based protocols with n rounds. Only the Tree protocol achieves such lower bound, at the cost of an exponential number of states, though. In this section we introduce a subclass of lookup-based protocols that contains protocols whose resistance to pre-ask attacks is arbitrarily close to the above-mentioned optimal bound.

6.1 Uniform Protocols

Protocols within the proposed subclass are layered and random-labeled. In addition, they satisfy an additional property that we call *uniformity*. The property is related to the possibility for an adversary to guess the correct states in an execution, i.e. to have certainty on the responses. We formally define the concept of uniformity and later on we explain our intuition behind it.

Definition 10 (Uniform Protocol)**.** *A lookup-based protocol P for $n > 0$ rounds is u-uniform, for some $u \in \{1,\ldots,n\}$, if it is layered and random-labeled and:*

$$\forall k \in \{1,\ldots,n\},\ x,y \in \{0,1\}^k.\quad \hat{\delta}(x) = \hat{\delta}(y) \iff \forall i \in \{\max(1,k-u+1),\ldots,k\}.\ x_i = y_i. \tag{13}$$

In other words, u-uniformity means that two input sequences of length k reach the same state if and only if the last u symbols (or k symbols if $k \leq u$) of the two sequences are equal.

Intuitively the notion of uniformity is related to the possibility for an adversary to predict the correct states in the pre-ask session. Suppose the adversary chooses a challenge sequence $x_1\ldots x_n$ to query the prover. Also, consider $A = (\Sigma,\Gamma,Q,q_0,\delta,\ell)$ to be the selected automaton for the protocol execution, which is unknown to the adversary. Suppose now $y_1\ldots y_n$ are the verifier's challenges. Let's call $q_1\ldots q_n$ and $q'_1\ldots q'_n$ the sequences of states reached by both challenge sequences, i.e. $q_i = \hat{\delta}(x_1\ldots x_i)$ and $q'_i = \hat{\delta}(y_1\ldots y_i)$ for all $i \in \{1,\ldots,n\}$. Intuitively, the more elements $q_1\ldots q_n$ and $q'_1\ldots q'_n$ have in common, the more vulnerable the protocol becomes, since the adversary has the responses for those states. In the case of a u-uniform protocol, for the adversary to reach the correct state, let's say at round i, he needs to guess all the u (or i if $i \leq u$) last verifier's challenges in advance. So, the higher the uniformity value u, the harder it gets for the adversary to make the correct guesses.

We say that a protocol is *uniform* if it is u-uniform for some u. We will refer to u as the *uniformity value* of a protocol. It is ensured that this uniformity value is unique because the set of input symbols has more than one element. Next, we show that HK [14] and Tree [7] protocols are 1-uniform and n-uniform, respectively.

Proposition 1. *The* HK *protocol is* 1*-uniform.*

Proof. In this proof we will use the definition of the HK protocol provided in Example 1. It is trivial that HK is layered and random-labeled. So let's proceed to prove Equation 13 with $u = 1$.

Let $A = (\Sigma,\Gamma,Q,q_0,\delta,\ell) \in \mathsf{HK}$, $k \in \{1,\ldots,n\}$ and $x,y \in \{0,1\}^n$. Let $a,b \in Q = \{0,\ldots,2n\}$ be two states such that $a \in \hat{\delta}(x_1\ldots x_{k-1})$ and $b \in \hat{\delta}(y_1\ldots y_{k-1})$. Also, let $a',b' \in \{0,1\}$ such that $a \equiv a' \pmod 2$ and $b \equiv b' \pmod 2$. Hence,

$$\begin{aligned}
&\hat{\delta}(x) = \hat{\delta}(y)\\
&\quad\iff \delta(a,x_k) = \delta(b,y_k)\\
&\quad\iff a + x_k + 1 + a' = b + y_k + 1 + b'\\
&\quad\iff a + x_k + 1 + a' \equiv b + y_k + 1 + b' \pmod 2\\
&\quad\iff 2a + x_k + 1 \equiv 2b + y_k + 1 \pmod 2\\
&\quad\iff x_k \equiv y_k \pmod 2 \iff x_k = y_k.
\end{aligned}$$

□

Proposition 2. *The* Tree *protocol with $n > 0$ rounds is n-uniform.*

Proof. Consider the following model for the $\mathsf{Tree} = \{A_0,A_1,\ldots,A_N\}$ where $N = 2^{2^{n+1}-2} - 1$ and, for all $i \in \{0,\ldots,N\}$, $A_i = (\Sigma,\Gamma,Q,q_0,\delta,\ell_i)$ such that:

- $\Sigma = \Gamma = \{0,1\}$, $Q = \{0,1,\ldots,2^{n+1}-2\}$, $q_0 = 0$.
- $\delta(q,c) = 2q + c + 1$.
- $\ell_i(q)$ is the q-th bit of the binary representation of i.

It is trivial that Tree is layered and random-labeled. So let's proceed to prove Equation 13 with $u = n$.

Let $A = (\Sigma,\Gamma,Q,q_0,\delta,\ell) \in \mathsf{Tree}$, $k \in \{1,\ldots,n\}$ and $x,y \in \{0,1\}^n$. Let $a,b \in Q = \{0,\ldots,2n\}$ be two states such that $a \in \hat{\delta}(x_1\ldots x_{k-1})$ and $b \in \hat{\delta}(y_1\ldots y_{k-1})$. Hence,

$$\begin{aligned}
\hat{\delta}(x) = \hat{\delta}(y) &\iff \delta(a,x_k) = \delta(b,y_k)\\
&\iff 2a + x_k + 1 = 2b + y_k + 1.
\end{aligned}$$

We will proceed to prove that $2a + x_k + 1 = 2b + y_k + 1 \iff (a = b \wedge x_k = y_k)$. The implication from right to left is trivial, so we proceed by proving the implication from left to right. Indeed,

$$\begin{aligned}
&2a + x_k + 1 = 2b + y_k + 1\\
&\quad\implies 2a + x_k + 1 \equiv 2b + y_k + 1 \pmod 2\\
&\quad\implies x_k \equiv y_k \pmod 2 \implies x_k = y_k.
\end{aligned}$$

Hence, if $2a + x_k + 1 = 2b + y_k + 1$ and $x_k = y_k$ then $a = b$. So, from a recursive reasoning we have:

$$
\begin{aligned}
&\hat{\delta}(x) = \hat{\delta}(y) \\
\iff\ & \hat{\delta}(x_1 \dots x_{k-1}) = \hat{\delta}(y_1 \dots y_{k-1}) \wedge x_k = y_k \\
\iff\ & \hat{\delta}(x_1 \dots x_{k-2}) = \hat{\delta}(y_1 \dots y_{k-2}) \\
& \wedge x_{k-1} = y_{k-1} \wedge x_k = y_k \\
& \dots \\
\iff\ & \hat{\delta}(x_1) = \hat{\delta}(y_1) \\
& \wedge x_2 = y_2 \wedge \dots \wedge x_{k-1} = y_{k-1} \wedge x_k = y_k \\
\iff\ & x_1 = y_1 \wedge x_2 = y_2 \wedge \dots \wedge x_{k-1} = y_{k-1} \wedge x_k = y_k.
\end{aligned}
$$

□

6.2 Pre-Ask Attacks in Uniform Protocols

To compute the highest success probability of a pre-ask attack in uniform protocols we will use the results from Theorem 2. That is to say, we will compute the adversary's success probability when executing the optimal pre-ask strategy defined in Theorem 2. We recall that such a strategy consists in replying to the verifier's challenges with the answers received from the prover.

Theorem 3. *Let P be a lookup-based protocol for $n > 0$ rounds that is u-uniform, for some $u \in \{1, \dots, n\}$. Then $preask(P) = R_n$, where $R_0 = 1$ and for all $i \in \{1, \dots, n\}$:*

$$R_i = \frac{1}{2^i} + \sum_{j=0}^{i-1} \frac{R_{i-j-1}}{2^{j+\min(u,j+1)+1}}.$$

Proof. Let $x \in \{0,1\}^n$ be an input sequence representing the adversary's challenges to query the prover in the pre-ask phase. Let $A = (\Sigma, \Gamma, Q, q_0, \delta, \ell) \in P$ and $x' \in \{0,1\}^n$ be a random automaton and a random bit sequence, respectively. Let $y, y' \in \{0,1\}^n$ be two binary sequences such that $(A, x, y) \sqsubset P$ and $(A, x', y') \sqsubset P$. The input sequence x' represents the one picked (randomly) by the verifier to execute the fast phase.

According to Theorem 2, we have that $preask(P) = \Pr(y = y')$. In order to compute $\Pr(y = y')$, consider the following events:

- S_i is the event that $y_1 \dots y_i = y'_1 \dots y'_i$ for all $i \in \{1, \dots, n\}$.
- $M_{i,j}$ is the event that $x_{i-j+1} \dots x_i = x'_{i-j+1} \dots x'_i \wedge x_{i-j} \neq x'_{i-j}$ for all $i \in \{1, \dots, n\}$ and $j \in \{0, \dots, i-1\}$. Note that $M_{i,0}$ becomes $x_i \neq x'_i$.
- E_i is the event that $x_1 \dots x_i = x'_1 \dots x'_i$ for all $i \in \{1, \dots, n\}$. Notice that E_i occurs if none of the events $M_{i,i-1}$ do, which means that $\Pr(E_i \vee M_{i,0} \vee \dots \vee M_{i,i-1}) = 1$.

Our goal is to compute the values of $\Pr(S_i)$ and in particular $\Pr(S_n)$. By the law of total probability we have:

$$
\begin{aligned}
\Pr(S_i) &= \Pr(S_i|E_i)\Pr(E_i) + \sum_{j=0}^{i-1} \Pr(S_i|M_{i,j})\Pr(M_{i,j}) \\
&= \frac{1}{2^i} + \sum_{j=0}^{i-1} \Pr(S_i|M_{i,j})\Pr(M_{i,j}), \qquad (14)
\end{aligned}
$$

because $\Pr(S_i|E_i) = 1$ and $\Pr(E_i) = \frac{1}{2^i}$. Moreover, for all $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, i-1\}$, since the sequence x' is chosen randomly and its bits are independent, we have that:

$$
\begin{aligned}
\Pr(M_{i,j}) = &\Pr(x_{i-j} \neq x'_{i-j}) \qquad (15) \\
&\times \prod_{k=i-j+1}^{i} \Pr(x_k = x'_k) = \frac{1}{2} \times \frac{1}{2^j} = \frac{1}{2^{j+1}}.
\end{aligned}
$$

Observe that $\Pr(M_{i,0}) = \Pr(x_i \neq x'_i) = \frac{1}{2}$. Now let's compute the values $\Pr(S_i \mid M_{i,j})$ for $i \in \{1, \dots, n\}$ and $j \in \{0, \dots, i-1\}$. Given i and j, assume $M_{i,j}$ occurs, i.e. the input sequences $x_1 \dots x_i$ and $x'_1 \dots x'_i$ have the same last j symbols. So, if $j \geq u$, because of the uniformity property, we have that:

$$
\begin{aligned}
&\hat{\delta}(x_1 \dots x_{i-j}) \neq \hat{\delta}(x'_1 \dots x'_{i-j}), \\
&\hat{\delta}(x_1 \dots x_{i-j+1}) \neq \hat{\delta}(x'_1 \dots x'_{i-j+1}), \\
&\dots \\
&\hat{\delta}(x_1 \dots x_{i-j+u-1}) \neq \hat{\delta}(x'_1 \dots x'_{i-j+u-1}), \qquad (16)
\end{aligned}
$$

and

$$
\begin{aligned}
&\hat{\delta}(x_1 \dots x_{i-j+u}) = \hat{\delta}(x'_1 \dots x'_{i-j+u}), \\
&\hat{\delta}(x_1 \dots x_{i-j+u+1}) = \hat{\delta}(x'_1 \dots x'_{i-j+u+1}), \\
&\dots \\
&\hat{\delta}(x_1 \dots x_i) = \hat{\delta}(x'_1 \dots x'_i). \qquad (17)
\end{aligned}
$$

From Equations 16 and 17 and given that the protocol is random-labeled we derive that, for every $k \in \{i-j, \dots, i\}$:

$$\Pr(y_k = y'_k | M_{i,j}) = \begin{cases} 1/2 & \text{if } k \leq i-j+u-1, \\ 1 & \text{otherwise.} \end{cases}$$

This leads to:

$$\Pr(y_{i-j} \dots y_i = y'_{i-j} \dots y'_i | M_{i,j}) = \frac{1}{2^u}. \qquad (18)$$

On the other hand, if $j < u$ then $\hat{\delta}(x_1 \dots x_k) \neq \hat{\delta}(x'_1 \dots x'_k)$ for every $k \in \{i-j, i\}$ and consequently:

$$\Pr(y_{i-j} \dots y_i = y'_{i-j} \dots y'_i | M_{i,j}) = \frac{1}{2^{j+1}}. \qquad (19)$$

Furthermore, the event S_{i-j-1} and the event that $y_{i-j} \dots y_i$ is equal to $y'_{i-j} \dots y'_i$ are independent, given the uniformity property and that $x_{i-j} \neq x'_{i-j}$. This gives:

$$
\begin{aligned}
\Pr(S_i|M_{i,j}) = &\Pr(S_{i-j-1}|M_{i,j}) \qquad (20) \\
&\times \Pr(y_{i-j} \dots y_i = y'_{i-j} \dots y'_i | M_{i,j}).
\end{aligned}
$$

Equations 18, 19 and 20 give:

$$\Pr(S_i|M_{i,j}) = \frac{\Pr(S_{i-j-1}|M_{i,j})}{2^{\min(u,j+1)}}. \quad (21)$$

Observe now that the events $M_{i,j}$ and S_{i-j-1} are independent, which means that $\Pr(S_{i-j-1}|M_{i,j}) = \Pr(S_{i-j-1})$. By applying this result to Equation 21 we obtain:

$$\Pr(S_i|M_{i,j}) = \frac{\Pr(S_{i-j-1})}{2^{\min(u,j+1)}}. \quad (22)$$

Finally, from Equations 14, 15 and 22 and by applying the substitution $R_i = \Pr(S_i)$ we obtain the expected recursive formula. We conclude this proof by remarking that $preask(P) = \Pr(S_n) = R_n$. □

A consequence of this theorem is that, in uniform protocols, the adversary has no advantage in selecting the challenges to query the prover. In the next corollaries we show, by using the previous theorem, a security computation in terms of pre-ask attacks for the mentioned HK and Tree protocols.

Corollary 1. *Consider the* HK *protocol for $n > 0$ rounds as defined in Example 1. Then*

$$preask(\mathsf{HK}) = \left(\frac{3}{4}\right)^n. \quad (23)$$

Proof. Since the HK protocol is 1-uniform (see Proposition 1), we have:

$$R_i = \frac{1}{2^i} + \sum_{j=0}^{i-1} \frac{R_{i-j-1}}{2^{j+2}}.$$

Hence, by multiplying the previous equation by 2^i we have:

$$2^i R_i = 1 + \sum_{j=0}^{i-1} 2^{i-j-2} R_{i-j-1} = 1 + \frac{1}{2}\sum_{j=0}^{i-1} 2^{i-j-1} R_{i-j-1}.$$

Now, by substituting $k = i-j-1$, the last equation can be written as $2^i R_i = 1 + \frac{1}{2}\sum_{k=0}^{i-1} 2^k R_k$, since $i-j-1$ goes from 0 to $i-1$. Now, by applying the substitution $B_i = 2^i R_i$ we obtain:

$$B_i = 1 + \frac{1}{2}\sum_{k=0}^{i-1} B_k. \quad (24)$$

Then, from Equation 24 we obtain $B_{i+1} - B_i = \frac{1}{2}B_i$ and thus $B_{i+1} = \frac{3}{2}B_i$. This implies that $B_{i+1} = \frac{3}{2}B_i$ and given that $B_0 = 1$, we obtain $B_i = \left(\frac{3}{2}\right)^i$. Therefore $R_i = \frac{1}{2^i}B_i = \left(\frac{3}{4}\right)^i$ and in particular, $preask(P) = R_n = \left(\frac{3}{4}\right)^n$. □

Corollary 2. *Consider the* Tree *protocol for $n > 0$ rounds as defined in Proposition 2 protocol. Then $preask(\mathsf{Tree}) = \frac{1}{2^n}\left(1 + \frac{n}{2}\right)$.*

Proof. Since the Tree protocol is n-uniform (see Proposition 2), we have:

$$R_i = \frac{1}{2^i} + \sum_{j=0}^{i-1} \frac{R_{i-j-1}}{2^{2j+2}}.$$

By multiplying the previous equation by 4^i we obtain:

$$4^i R_i = 2^i + \sum_{j=0}^{i-1} 4^{i-j-1} R_{i-j-1} = 2^i + \sum_{k=0}^{i-1} 4^k R_k.$$

Hence, let $B_i = 4^i R_i$, then:

$$B_i = 2^i + \sum_{k=0}^{i-1} B_k. \quad (25)$$

Therefore, from Equation 25 we derive that $B_{i+1} - B_i = 2^i + B_i$ and consequently $B_{i+1} = 2B_i + 2^i$ and $\frac{B_{i+1}}{2^{i+1}} = \frac{B_i}{2^i} + \frac{1}{2}$. Now, by letting $D_i = \frac{B_i}{2^i}$ we have that $D_{i+1} = D_i + \frac{1}{2}$ and therefore $D_i = \frac{i}{2} + D_0$. Since $R_0 = 1$, we have that $B_0 = D_0 = 1$. Therefore, $B_i = 2^i(1 + \frac{i}{2})$ which implies that $R_i = \frac{1}{2^i}(1 + \frac{i}{2})$. This gives us $preask(\mathsf{Tree}) = R_n = \frac{1}{2^n}\left(1 + \frac{n}{2}\right)$. □

Next we prove that the resistance to pre-ask attacks of uniform protocols monotonically depends on their uniformity value.

Theorem 4. *Let $n > 0$ an integer number and $u, v \in \{1, \ldots, n\}$ with $u \leq v$. Let P_u and P_v be an u-uniform and a v-uniform protocols, respectively; both with n rounds. Then $preask(P_u) \geq preask(P_v)$.*

Proof. We introduce the notation

$$R_i^u = \frac{1}{2^i} + \sum_{j=0}^{i-1} \frac{R_{i-j-1}^u}{2^{j+\min(u,j+1)+1}} \quad (26)$$

to refer to the recursive equation in Theorem 3 for P_u. Analogously, we use R_i^v for P_v.

We proceed by induction over i to prove that $R_i^u \geq R_i^v$ for every $i \in \{0, \ldots, n\}$ and in particular that $R_n^u \geq R_n^v$.

- *Base case:* It trivially holds, given that $R_0^u = R_0^v = 1$.
- *Hypothesis:* $\forall j < i.\ R_j^u \geq R_j^v$.
- *Thesis:* $R_i^u \geq R_i^v$. From $u \leq v$ it follows that $\min(u, j+1) \leq \min(v, j+1)$ for all $j \in \{1, \ldots, n\}$ and consequently $2^{\min(u,j+1)} \leq 2^{\min(v,j+1)}$. This implies that

$$\frac{1}{2^{j+\min(u,j+1)+1}} \geq \frac{1}{2^{j+\min(v,j+1)+1}}. \quad (27)$$

Besides, from the induction hypothesis, for every $j \in \{0, \ldots, i-1\}$ we obtain that

$$R_{i-j-1}^u \geq R_{i-j-1}^v. \quad (28)$$

From Equations 26, 27 and 28 we obtain that $R_i^u \geq R_i^v$.

□

For a given $n > 0$, consider $f\colon \{1,\ldots,n\} \to \left[\frac{1}{2^n}(1+\frac{n}{2}), \left(\frac{3}{4}\right)^n\right]$ to be a function such that $f(u) = preask(P_u)$ where P_u is an u-uniform protocol. Theorem 4 demonstrates that f is decreasing and approaches $\frac{1}{2^n}\left(1+\frac{n}{2}\right)$ when u approaches n. Based on these results, we can affirm that the closer the uniformity value gets to n (resp. 1) the lower (resp. higher) the success probability of a pre-ask attack. In particular, n-uniform protocols (such as Tree) are optimal within this class, whereas 1-uniform (such as HK) perform worst.

In the following, we model a family of uniform protocols and describe them in standard cryptographic notation. Also, we show that for every uniformity value, there exists at least one protocol within our proposed class. This means that, for every $u \leq n$, where n is the number of rounds, we provide a construction of a u-uniform protocol. This affirms that we can build a protocol with mafia fraud resistance arbitrarily close to optimal, by defining its uniformity value and using our model.

6.3 Realization of Uniform Protocols

Let $n > 0$ and $u \in \{1,\ldots,n\}$ be two integer numbers. Consider the following construction of protocol P for n rounds. $P = \{A_0, A_2, \ldots, A_N\}$ such that $A_i = (\Sigma, \Gamma, Q, q_0, \delta, \ell_i)$, where

- $\Sigma = \Gamma = \{0,1\}$,
- $Q = \{(i,d) \mid 0 \leq i \leq n \wedge 0 \leq d < \min(2^u, 2^i)\}$,
- $q_0 = (0,0)$,
- $\delta((i,d),c) = (i+1, (2d+c) \pmod{2^u})$ for every $c \in \{0,1\}$ and $(i,d) \in Q$ such that $i < n$,
- $N = 2^{|Q|-1} - 1$, and
- for every $j \in \{1,\ldots,N\}$, $\ell_j(q)$ is the k-th least significant bit in the binary representation of j and k is the position of q in $Q - \{q_0\}$.

In this construction, we define Q as a set of pairs of integers. Each pair (i,d) represents a state, in layer i, where d is the position in this layer. For two binary strings to share the last u bits, their decimal representations have to leave the same remainder when divided by 2^u. Based on this property, we build our state transition function. The expression $\delta((i,d),c) = (i+1, (2d+c) \pmod{2^u})$ stands for this idea. Notice that $d(c_1 \ldots c_i) = 2d(c_1 \ldots c_{i-1}) + c_i$, where $d(.)$ represents a function that converts binary strings into integers (left-to-right order). We use $d(.)$ here as a function since it relates the second value of the pairs representing the states. Moreover, for every $j \in \{1,\ldots,u\}$ and $c \in \{0,1\}^j$ the values $d(c)$ have at most 2^j different remainders when divided by 2^u, this explains our upper bound for d in the definition of Q. The last two rules in our construction represent the random-labeling property, although in practice this is achieved by generating a distribution of random bits over the set of states. A formal proof of these ideas is shown in the next lemma.

Lemma 2. *The proposed protocol is u-uniform.*

Proof. We first prove that P is layered and random-labeled. Because of the definition of δ, for every pair $x, y \in \{0,1\}^i \times \{0,1\}^j$ with $i \neq j$ we have that $\hat{\delta}(x) = (i, a_x)$ and $\hat{\delta}(y) = (j, a_y)$, where a_x and a_y are integer numbers (they are irrelevant here). Then $(i, a_x) \neq (j, a_y)$ and $\hat{\delta}(x) \neq \hat{\delta}(y)$. Therefore P is layered.

To show the random-labeling property, consider any labeling function $\ell\colon Q - \{q_0\} \to \{0,1\}$. Let $B = b_1 b_2 \ldots b_{|Q|-1}$ be a binary string such that $b_i = \ell(q_i)$ for every $i \in \{1,\ldots,|Q|-1\}$. We recall that q_0 does not require a label since it is never used for replies. Let $D = \sum_{i=1}^{|Q|-1} 2^{i-1} b_i$ be an integer number, i.e. the decimal representation of B. Then, given that $N = 2^{|Q|-1} - 1$ we have that $D \in \{0,\ldots,N\}$ and, because of our construction, $\ell_D = \ell$. This proves that for any ℓ, it holds that $(\{0,1\}, \{0,1\}, Q, q_0, \delta, \ell) \in P$. Every labeling function is unique since it is related to a unique integer number in D. This states that P is indeed a set.

Next, we demonstrate that P satisfies the uniformity property. Let $A = (\Sigma, \Gamma, Q, q_0, \delta, \ell) \in P$. Let $k \in \{1,\ldots,n\}$ and let $x, y \in \{0,1\}^k$. Because of our definition of δ, we derive that $\hat{\delta}(x) = (k, S_x \pmod{2^u})$ and $\hat{\delta}(y) = (k, S_y \pmod{2^u})$ where

$$S_x = \sum_{i=1}^{k} 2^{k-i} x_i \quad \text{and} \quad S_y = \sum_{i=1}^{k} 2^{k-i} y_i.$$

Note that S_x and S_y are the decimal representations of the bit strings x and y, respectively. Hence,

$$\hat{\delta}(x) = \hat{\delta}(y) \iff S_x \equiv S_y \pmod{2^u}.$$

We will analyze now two cases:

- $k \leq u$: We have $S_x < 2^u$ and $S_y < 2^u$ and therefore:

$$\hat{\delta}(x) = \hat{\delta}(y) \\ \iff S_x \equiv S_y \pmod{2^u} \iff S_x = S_y \iff x = y.$$

- $k > u$: We can write S_x (and analogously S_y) in the following way:

$$S_x = \sum_{i=1}^{k-u} 2^{k-i} x_i + \sum_{i=k-u+1}^{k} 2^{k-i} x_i \\ = 2^u \left(\sum_{i=1}^{k-u} 2^{k-i-u} x_i \right) + \sum_{i=k-u+1}^{k} 2^{k-i} x_i.$$

Since $k - i - u \geq 0$ for all $i \in \{1,\ldots,k-u\}$, the elements in the first sum are integers. This implies that $S_x \equiv S'_x \pmod{2^u}$ and $S_y \equiv S'_y \pmod{2^u}$, where:

$$S'_x = \sum_{i=k-u+1}^{k} 2^{k-i} x_i \quad \text{and} \quad S'_y = \sum_{i=k-u+1}^{k} 2^{k-i} y_i.$$

Therefore:

$$\hat{\delta}(x) = \hat{\delta}(y) \\ \iff S_x \equiv S_y \pmod{2^u} \iff S'_x \equiv S'_y \pmod{2^u}.$$

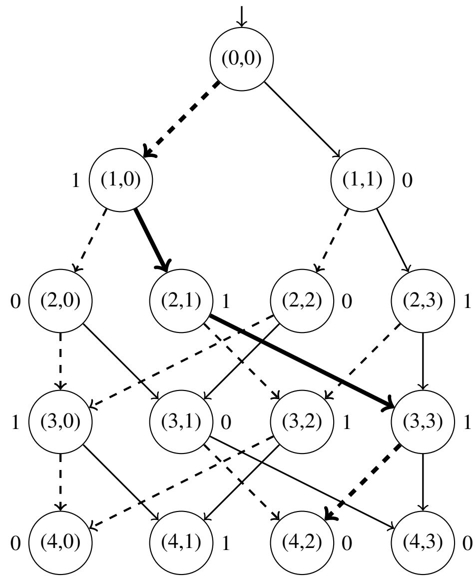

Figure 4: An automaton representing an instantiation of a 2-uniform protocol for $n = 4$ rounds. Dashed and solid arrows represent transitions when the input symbol is 0 and 1, respectively. We highlight in **bold** an execution with challenge sequence 0110 whose responses are 1110.

Given that $k - i < u$ for every $i \in \{k-u+1,\dots,k\}$, we deduce that both S'_x and S'_y are smaller than 2^u. This implies that:

$$\hat{\delta}(x) = \hat{\delta}(y)$$
$$\iff S'_x = S'_y \iff x_{k-u+1}\dots x_k = y_{k-u+1}\dots y_k.$$

Finally, from both cases we obtain the necessary and sufficient condition stated in Definition 10. □

The intuition behind the above construction is that the set of states Q can be partitioned into n sets $Q_0, Q_1, \dots, Q_n$ such that $i \in \{0,\dots,n\}$ where $Q_i = \{(i,d) \mid 0 \le d < \min(2^u, 2^i)\}$. The transition function only connects states in Q_i with states in Q_{i+1}. The corresponding state for a sequence of input symbols $c_1 \dots c_i \in \{0,1\}^i$ is the k-th state in the layer i, where k is the remainder of the division of $\sum_{j=1}^{i} 2^{j-1} c_j$ by 2^u. The composition of the labeling functions basically represents a random distribution of bits over the set of states Q. An example of an automaton for a 2-uniform protocol following the model above is depicted in Figure 4.

In Figure 5 we represent the proposed protocol. We describe next in standard cryptographic notation for distance-bounding protocols. Concretely, the proposed protocol consists of the following phases:

1. **Initialization phase:** The prover and the verifier agree on the following parameters:
   - A shared key x.
   - An integer number $m > 0$ which represents the length of the nonces.
   - An integer number $n > 0$ which represents the number of rounds.
   - An integer number $u \in \{1,\dots,n\}$ which represents the uniformity value.
   - A pseudo-random function g.
   - A threshold Δt_{max} for the RTTs.
2. **Slow phase:** Both parties generate nonces, $N_{\mathscr{P}}$ for the prover and $N_{\mathscr{V}}$ for the verifier. The value $N_{\mathscr{V}}$ is sent to the prover which constructs the labeling function from $g(x, N_{\mathscr{V}}, N_{\mathscr{P}})$. Then, the prover sends the nonce $N_{\mathscr{P}}$ to the verifier and the latter also computes the function $g(x, N_{\mathscr{V}}, N_{\mathscr{P}})$ to agree with the prover on the labeling function.

   The shared pseudo-random function g outputs n registers $B^1||B^2||\dots||B^n$ such that B^i is a $2^{\min(u,i)}$-bit string. Then $\ell((i,d)) = B^i_d$ for all $i \in \{1,\dots,n\}$ and $0 \le d < \min(2^u, 2^i)$.
3. **Fast phase:** This phase is composed of n rounds numbered from 1 to n. At the i-th round, the verifier sends a challenge bit c_i to the prover which moves from the previous state q_{i-1} to the next one $q_i = \delta(q_{i-1}, c_i)$ and replies with the corresponding bit $\ell(q_i)$.
4. **Verification phase:** The protocol succeeds if and only if (i) all the exchange times are less than or equal to the maximum value Δt_{max} and (ii) all the responses are correct.

The proposed protocol requires an amount of memory of $(n-u+2)2^u - 2$ bits. The length of the first u registers $B^1, B^2, \dots, B^u$ is $2^1, 2^2, \dots, 2^u$, respectively. The remaining ones $B^{u+1}, \dots, B^n$ have 2^u bits each one. So, in total we have $\sum_{i=1}^{u} 2^i + (n-u)2^u$ bits, which is $(2^{u+1} - 2) + (n-u)2^u = (n-u+2)2^u - 2$.

7 Conclusions

In this paper we have introduced an abstract model for the description of lookup-based distance-bounding protocols. The model represents the protocol as a set of State-Labeled Deterministic Finite Automata and the protocol's executions are represented by a random walk through a randomly selected automata. The model is sufficiently expressive to describe many published protocols, which include the well-known HK and the Tree protocols.

The virtue of this model is that it supports generic analysis. For instance, we can analyze the security limits of a protocol in relation to the number of rounds of the protocol's fast phase. Further, we introduced the notion of *uniformity*, which expresses that randomly chosen walks through the automaton have no bias towards a particular automaton state. Finally, we developed a family of uniform protocols in our model and proved that it has an excellent performance in relation to its memory requirements.

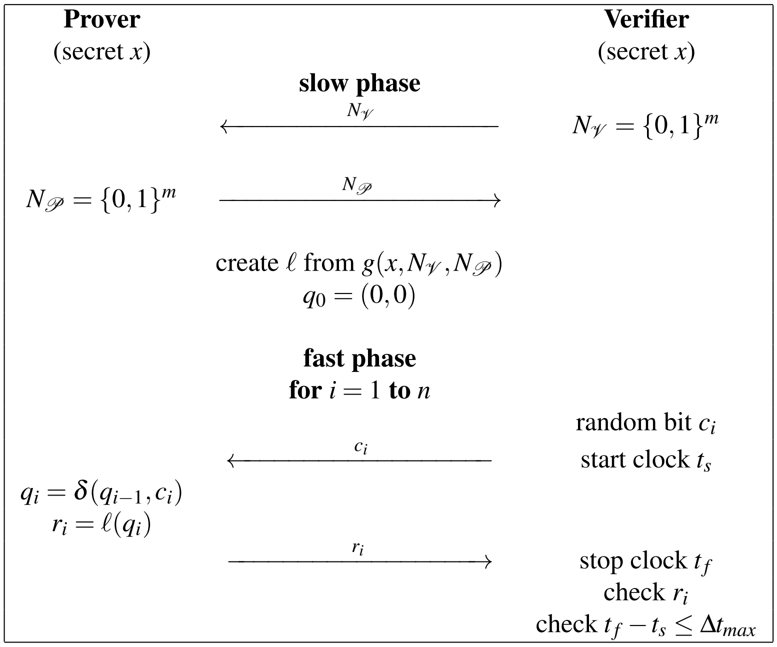

Figure 5: The proposed *u*-uniform protocol.

References

[1] G. P. Hancke, K. Mayes, and K. Markantonakis, "Confidence in smart token proximity: Relay attacks revisited," *Computers & Security*, vol. 28, no. 7, pp. 615–627, 2009.

[2] Y. Desmedt, C. Goutier, and S. Bengio, "Special uses and abuses of the fiat-shamir passport protocol," in *CRYPTO'87*, pp. 21–39, 1987.

[3] G. P. Hancke and M. G. Kuhn, "An RFID distance bounding protocol," in *SecureComm'05*, pp. 67–73, 2005.

[4] S. Brands and D. Chaum, "Distance-bounding protocols," in *EUROCRYPT'93*, vol. 765 of *LNCS*, pp. 344–359, Springer, 1994.

[5] T. Beth and Y. Desmedt, "Identification tokens - or: Solving the chess grandmaster problem," in *CRYPTO'90*, pp. 169–177, 1990.

[6] R. Trujillo-Rasua, B. Martin, and G. Avoine, "The Poulidor distance-bounding protocol," in *RFIDSec'10*, pp. 239–257, 2010.

[7] G. Avoine and A. Tchamkerten, "An efficient distance bounding RFID authentication protocol: Balancing false-acceptance rate and memory requirement," in *ISC'09*, pp. 250–261, 2009.

[8] R. Trujillo-Rasua, B. Martin, and G. Avoine, "Distance bounding facing both mafia and distance frauds," *IEEE Transactions on Wireless Communications*, vol. 13, no. 10, pp. 5690–5698, 2014.

[9] C. H. Kim and G. Avoine, "RFID distance bounding protocols with mixed challenges," *IEEE Transactions on Wireless Communications*, vol. 10, no. 5, pp. 1618–1626, 2011.

[10] S. Kardas, M. S. Kiraz, M. A. Bingöl, and H. Demirci, "A novel RFID distance bounding protocol based on physically unclonable functions," in *RFIDSec'11*, vol. 7055 of *LNCS*, pp. 78–93, Springer, 2012.

[11] G. Avoine, C. Lauradoux, and B. Martin, "How secret-sharing can defeat terrorist fraud," in *WISEC'11*, pp. 145–156, ACM, 2011.

[12] A. O. Gürel, A. Arslan, and M. Akgün, "Non-uniform stepping approach to RFID distance bounding problem," in *Proc. 5th International Workshop on Data Privacy Management (DPM'10), and 3rd International Conference on Autonomous Spontaneous Security (SETOP'10)*, vol. 6514 of *LNCS*, pp. 64–78, Springer, 2011.

[13] J. Clulow, G. P. Hancke, M. G. Kuhn, and T. Moore, "So near and yet so far: Distance-bounding attacks in wireless networks," in *ESAS'06*, vol. 4357 of *LNCS*, pp. 83–97, Springer, 2006.

[14] G. P. Hancke and M. G. Kuhn, "Attacks on time-of-flight distance bounding channels," in *WiSec'08*, pp. 194–202, ACM, 2008.

[15] L. Bussard and W. Bagga, "Distance-bounding proof of knowledge to avoid real-time attacks," in *Security and Privacy in the Age of Ubiquitous Computing, IFIP TC11 20th International Conference on Information Security (SEC'05)*, vol. 181 of *IFIP Advances in Information and Communication Technology*, pp. 223–238, Springer, 2005.

[16] S. Capkun, L. Buttyán, and J. Hubaux, "SECTOR: secure tracking of node encounters in multi-hop wireless networks," in *SASN'03*, pp. 21–32, ACM, 2003.

[17] C. H. Kim, G. Avoine, F. Koeune, F. Standaert, and O. Pereira, "The Swiss-Knife RFID distance bounding protocol," in *ICISC'08*, pp. 98–115, 2008.

[18] K. B. Rasmussen and S. Čapkun, "Location privacy of distance bounding protocols," in *CCS'08*, pp. 149–160, ACM, 2008.

[19] J. Munilla and A. Peinado, "Distance bounding protocols for RFID enhanced by using void-challenges and analysis in noisy channels," *Wireless Communications and Mobile Computing*, vol. 8, no. 9, pp. 1227–1232, 2008.

[20] G. Avoine, C. Floerkemeier, and B. Martin, "RFID distance bounding multistate enhancement," in *INDOCRYPT'09*, vol. 5922 of *LNCS*, pp. 290–307, Springer, 2009.

[21] G. Avoine, M. A. Bingöl, S. Kardas, C. Lauradoux, and B. Martin, "A framework for analyzing RFID distance bounding protocols," *Journal of Computer Security*, vol. 19, no. 2, pp. 289–317, 2011.

[22] G. Avoine, S. Mauw, and R. Trujillo-Rasua, "Comparing distance bounding protocols: A critical mission supported by decision theory," *Computer Communications*, vol. 67, pp. 92–102, 2015.

[23] R. Trujillo-Rasua, "Complexity of distance fraud attacks in graph-based distance bounding," in *Proc. 10th International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (MOBIQUITOUS'13)*, vol. 131 of *LNCS*, pp. 289–302, Springer, 2013.

[24] O. Kara, S. Kardaş, M. A. Bingöl, and G. Avoine, "Optimal security limits of RFID distance bounding protocols," in *RFIDSec'10*, vol. 6370 of *LNCS*, pp. 220–238, Springer, 2010.