Computing Science Group

Quantifying pervasive authentication: the case of the Hancke-Kuhn protocol

Dusko Pavlovic
Catherine Meadows

CS-RR-09-09

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford, OX1 3QD

Quantifying pervasive authentication: the case of the Hancke-Kuhn protocol

Dusko Pavlovic
Kestrel Insitute and Oxford University
dusko@comlab.ox.ac.uk

and

Catherine Meadows
Naval Research Laboratory
catherine.meadows@nrl.navy.mil

Abstract

As mobile devices pervade physical space, the familiar authentication patterns are becoming insufficient: besides entity authentication, many applications require, e.g. location authentication. While many interesting and subtle protocols have been proposed and implemented to provide such strengthened authentication, there are very few proofs that such protocols satisfy the required properties.

We consider the problem of adapting the Dolev-Yao-style reasoning methods for pervasive security. We show how the notion of guards*, previously used for symbolic reasoning about secrecy, can be extended into a tool for analyzing pervasive authentication. It supports a simple form of probabilistic reasoning, necessary for situations where the authentication cannot be achieved in absolute sense, and needs to be quantified. We show that extension of our protocol derivation logic, although quite modest, suffices to uncover some interesting properties of the Hancke-Kuhn distance bounding protocol, and to explain some of its deceiving simplicity.*

1 Introduction

Traditionally, there have been two paradigms used for proving protocol security. One, the symbolic paradigm, commonly known as "Dolev-Yao" models both protocol and attacker in terms of an algebraic theory [4]. While this has been criticized as crude, it is often highly effective and easily automated. The other, the computational paradigm, usually relies on some notion of indistinguishability from the point of view of a computationally limited attacker [6]. Recently, a lot of research, starting with [1], has been devoted to drawing the two paradigms closer together. This strategy has generally been to rely upon crypto-algorithms that themselves satisfy strong enough definitions of security, so that, if used in the proper way, they can be treated as Dolev-Yao "black boxes."

However, there is an emerging class of security protocols for which it seems very difficult to bring these two paradigms together. This class of protocols is used to provide security for what is known as *pervasive security*, that is security for a network that is pervasive in the human environment. Because of constraints on the various devices and communication channels involved, it is often necessary to use weak cryptography that does not satisfy standard notions of indistinguishability, but in such a way that it does not threaten the security of the protocol itself.

We consider the case where the standard authentication requirements need to be strengthened by the proofs of spatial proximity. In some cases, for example the protocols analyzed in [10, 14] it has been possible apply symbolic methods in a meaningful fashion. However, there other cases, such as the Hancke-Kuhn [8] distance bounding protocol that appear resistant to analysis in this fashion. This protocol implements the rapid, easily computable response of the prover by using weakly secure functions, that leak information. The amount of information leaked depends on a pseudo-random variable, generated during the course of the protocol. This makes it impossible to construct a useful symbolic model of this protocol. The algebraic approximations of the Hancke-Kuhn functions seem to either allow implementations which leak all information, or do not allow any implementations. So we must count bits somewhere. On the other hand, we would like to retain the advantages of algebraic modeling, such as its cleanness, and hopefully its amenability to automatic reasoning.

The goal of this paper is to provide a simple extension of the algebraic model used in [12], sufficient to capture these new primitives, while still supporting incremental reasoning and protocol derivations. Using this model, we provide a derivation and security proof of the Hancke-Kuhn protocol.

Paper outline. In Section 2 we review the basic ideas about the challenge-response authentication in general,

about proximity authentication in particular, and formulate the relevant security properties. We also describe the Hancke-Kuhn protocol itself, and finally introduce a probabilistic extension of the *guard* relation. Guards were originally used for symbolic reasoning about secrecy in [12]; here we use them for probabilistic reasoning about authenticity. In Section 3.2 we derive the necessary properties of the $\boxplus$ function used by the Hancke-Kuhn protocol, and prove that it is a canonical implementation of the properties required by Hancke-Kuhn. In Section 4 we give a quantitative proof of security of Hancke-Kuhn using the guards relation. In Section 5 we conclude the paper and discuss how our results could be extended.

2 Deriving the Hancke-Kuhn protocol

2.1 Challenge-response authentication in PDL

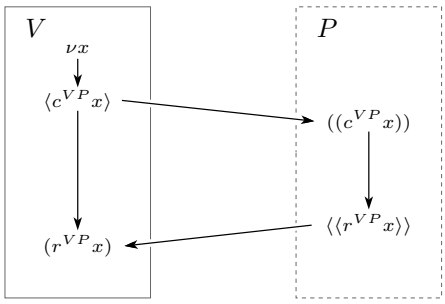

Figure 1. CR template

Our approach to modeling challenge-response protocols will be based on the approach used in the *Protocol Derivation Logic* (PDL) [9, 3]. PDL is a logic that is used to derive the conclusions a principal can form as a result of participating in a protocol. It is based on the Dolev-Yao black box model of security. The axioms of the logic describe the properties of communication channels and cryptographic algorithms, and the expected behavior of principals in cryptographic protocols. One uses the logic by expressing the desired properties of a protocol in terms of a template. One then uses the observations of a participant and the axioms of a logic to show that it is possible to derive the properties expressed in the template.

The earliest versions of PDL, e.g. in [9], only included axioms about authentication, assuming that some shared secrets were given in advance. A logical framework for reasoning about the protocols where such secrets are generated, was presented in [12]. Some methods to combine the authentication and the secrecy modules of PDL were discussed in [3]. The analysis of the Hancke-Kuhn authentication protocol in the present paper is based on a novel use of elements of secrecy logic from [12].

Like most other authentication protocols, the Hancke-Kuhn protocol is an instance of the Challenge-Response (CR) template, displayed in Figure 1. The verifier Victor generates a fresh value x, sends a challenge $c^{VP}x$ to the prover Peggy on the right, and receives a response $r^{VP}x$. The expressions νx, $\langle c^{VP}x\rangle$ and $(r^{VP}x)$ denote these three Victor's actions. The point of the CR template is that from his own actions, Victor can draw conclusions about Peggy's actions. More precisely, the functions c and r are assumed to be such that only Peggy could transform $c^{VP}x$ to $r^{VP}x$, so that Victor can be confident that, after he sent $c^{VP}x$ and before he received $r^{VP}x$, Peggy must have received a message containing $c^{VP}x$, and sent a message containing $r^{VP}x$. Formally, the CR template thus supports the following CR axiom, capturing Victor's reasoning:

$$\begin{aligned} & V:\ (\nu x)_V \triangleright \langle c^{VP}x\rangle_V \qquad\qquad \triangleright\ \ (r^{VP}x)_V \\ \Longrightarrow & \Big((\nu x)_V \triangleright \langle c^{VP}x\rangle_V \triangleright ((c^{VP}x))_P \triangleright \\ & \qquad\qquad \langle\langle r^{VP}x\rangle\rangle_{\overrightarrow{P}} \triangleright (r^{VP}x)_V \Big) \end{aligned} \qquad \textsf{(cr)}$$

where the relation $a \triangleright b$ says that action a occurs before action b. Intuitively, the reasoning in axiom (cr) can be justified by demonstrating that Victor can assume that

1. anyone who originated $r^{VP}x$ had to previously receive $c^{VP}x$, which could only happen after Victor sent it;
2. no one could produce $r^{VP}x$ without knowing the secret s^{VP}, so it must be Peggy.

This last conclusion is based on the assumption that only Peggy knows s^{VP}, or only Peggy and Victor. In both cases, Victor's reasoning is the same, because he knows that he did not send $r^{VP}x$.

The above informal justifications of (cr) can be refined into slightly more formal proof obligations, as follows. For any set of principals Π, it is required that

1. whenever there is a derivation $\Xi \vdash r^{VP}x$, then there must also be a derivation $\Xi \vdash c^{VP}x$, for any set of terms Ξ known to Π in a run of CR *before* $r^{VP}x$ is sent;
2. whenever there is a derivation $\Xi, c^{VP}x \vdash r^{VP}x$, then there must also be a derivation $\Xi, c^{VP}x \vdash s^{VP}$, for any set of terms Ξ known to Π in a run of CR before $r^{VP}x$ is sent.

Both of these proof obligations can be formalized in terms of the *guard* relation, used in [12]. We first recall the original idea of algebraic guards, and its formalization in [12], and then propose a probabilistic version, necessary for reasoning about the Hancke-Kuhn authentication.

2.1.1 Algebraic derivability and guards

The algebraic derivability relation, widely used in symbolic reasoning about security protocols, can be defined by

$$\Xi \vdash \Theta \quad \Longleftrightarrow \quad \forall t \in \Theta\, \exists \varphi \in \Sigma^*\, \exists \vec{\xi} \subseteq \Xi.\ t \stackrel{E}{=} \varphi(\vec{\xi}) \quad (1)$$

where Ξ and Θ are finite sets of terms from an algebra $\mathcal{T}$, and $\vec{\xi}$ is a tuple in Ξ. We assume that the algebra $\mathcal{T}$ is specified by an equational theory (Σ, E), where Σ are the operations and E the equations. We denote by Σ^* the set of derived operations, i.e. well-formed with respect to the arities; and by $t \stackrel{E}{=} s$ an equalition derivable from E.

Definition 2.1 *We say that a set of sets of terms $\mathcal{G}$* algebraically guards *a term t with respect to a set of terms Υ, and write*

$$\mathcal{G} \quad \text{guards} \quad t \quad \text{within} \quad \Upsilon$$

whenever

$$\forall \Xi \subseteq \Upsilon.\ \Xi \vdash t \quad \Rightarrow \quad \exists \Gamma \in \mathcal{G}.\ \Xi \vdash \Gamma \qquad (2)$$

Example. Let $\Upsilon = (DH)$ be the set of terms that may become known to the participants and eavesdroppers of a run of the Diffie-Hellman protocol. Then

$$\big\{\{x, g^y\}, \{y, g^x\}\big\} \quad \text{guards} \quad g^{xy} \quad \text{within} \quad \Upsilon_{DH}$$

Note that g^{xy} can be derived not only from $\{x, g^y\}$ and $\{y, g^x\}$ but also from $\{g, x, y\}$ and $\{g, xy\}$; however, neither of these sets can occur in a run of the Diffie-Hellman protocol between two honest principals, and are thus not contained in Υ_{DH}.

Perfect cryptography and pseudo-free algebras. The algebraic guard relation is based on the assumption that a term can only be derived algebraically, using the given operations and equations. A term t thus either lies in a subalgebra generated by a set of terms Ξ, or not, and we have

$$\Xi \vdash t \quad \vee \quad \Xi \nvdash t$$

This means that the attacks on the implementation of the term t are abstracted away. In particular, we assume that it is impossible to cryptanalyze the operations used to construct t, and to derive it by partial information about it. In other words, we assume *perfect cryptography*.

Moreover, we assume that the algebraic derivations $\Xi \vdash t$ only use the equations specified in the given algebraic theory (Σ, E). This means that the implementation of that algebraic theory is assumed to yield a free algebra, *or* that it is computationally unfeasible for the attacker to find the additional equations, not specified in the theory, and to use them in his derivations. This is roughly the *pseudo-free algebra* assumption [13].

2.1.2 Reducing authentication and freshness to guards

Both the authentication and freshness requirements in the CR template can be stated as instances of the guard relation. Authenticity of the response $r^{VP}x$ is established by proving that a secret s^{VP} is necessary to derive it, i.e. that it guards it relative to the set Υ_{CR} of the messages that become known in CR before the point when $r^{VP}x$ itself is first sent in some message. Likewise, the freshness of $r^{VP}x$ is established by proving that $c^{VP}x$ guards $r^{VP}x$ in Υ_{CR}.

In this section, we formalize this technique. We begin by formalizing the term context $\Upsilon_{\mathcal{Q}}$ of a protocol $\mathcal{Q}$.

In the following definition we use the notion of a *protocol run* defined in [12], that is, a run of a protocol P is a finite set of executions of P, whose actions are partially ordered by the $\triangleright$ relation, and such that each receive actions is preceded by a corresponding send action. That is, a protocol run is close to the idea of the *bundle* of Thayer, Herzog, and Guttman [5].

Definition 2.2 *Let $\mathcal{Q}$ be a protocol run, and A a set of actions in $\mathcal{Q}$. The* term context *is the set*

$$\mathcal{Q}(A) \quad = \quad \bigcup_{P \in \Pi} \Gamma_P^{\iota} \cup \Gamma_P^{\triangleright A}$$

where Π is the set of principals engaged in the run, Γ_P^{ι} is the set of terms known to a principal P initially, before $\mathcal{Q}$, and $\Gamma_P^{\triangleright A}$ is the set of the terms that P may receive in $\mathcal{Q}$ before *any of the actions $a \in A$ are executed.*

Proposition 2.3 *Let CR be challenge-response protocol, and let $\mathcal{Q}$ be a protocol run in CR. Suppose that the functions c^{VP} and r^{VP} used in CR are implemented to satisfy*

$$\Big\{\{c^{VP}x, s^{VP}\}\Big\} \quad \text{guards} \quad r^{VP}x \quad \text{within} \quad \mathcal{Q}(r^{VP}x) \quad \text{(crg)}$$

*where s^{VP} is a secret known only to Peggy (and possibly to Victor). Then (*cr*) is validated.*

The **proof** is obtained by expanding the definition of the algebraic guard relation and analyzing the term context of CR. For more about the use of this relation, the reader may wish to consult [12].

2.2 Proximity authentication

In pervasive computation, mobile devices often must prove not only their identity, but also their proximity. This is, for instance, the case whenever access to a secure location is controlled using smart cards. In principle, a smart card reader Victor could still simply use the standard cryptographic challenge-response authentication to assure the

prover Peggy has sent the response, and that she did that after she received his challenge. However, if Victor does not verify that Peggy is close, then an intruder Ivan could set up another smart card reader at a remote location where Peggy happens to be at that moment. He could then forward Victor's challenge to the other reader, and return Peggy's response by a radio link. In cyberspace, this would be a completely correct session of standard authentication, with Ivan as a mere relay. In physical space, this could be a breach of security, based not on spoofing Peggy's location, rather than her location. Hence the need for proximity authentication.

One method to ascertain that Peggy is close to Victor is by *distance bounding*. It is implemented by the template CRP, which is similar to the template CR, but Victor in it moreover records the time τ_0 when he sends the challenge $c^{VP}x$ and the time τ_1 when he receives the response. If CRP validates strengthening (cr) to

$$\begin{aligned} V:\ (\nu x)_V \triangleright \tau_0\langle c^{VP}x\rangle_V \quad &\triangleright\ \ \tau_1(r^{VP}x)_V \\ \Longrightarrow \Big((\nu x)_V \triangleright \tau_0\langle c^{VP}x\rangle_V \triangleright ((c^{VP}x))_P \triangleright \quad & \\ \langle\langle r^{VP}x\rangle\rangle_{\overrightarrow{P}} \triangleright\ \ \tau_1(r^{VP}x)_V \Big) \quad & \text{(crp)} \end{aligned}$$

then after a successful session, Victor conclude not only that the response must be from Peggy, but also Peggy must be at a distance

$$d(V,P) \quad \leq \quad \frac{c}{2}(\tau_1 - \tau_0)$$

where c is the maximal speed of the messages.

2.3 Hancke-Kuhn protocol

In order to guarantee that the timing information in CRP is accurate, the computation of the response from the challenge must take negligible time. Unfortunately, quickly computable functions tend to be cryptographically weak. Some protocols, such as the Brands-Chaum distance bounding protocol [2], achieve this quick response by separating the timed response from the cryptographic response. In contrast, the Hancke-Kuhn [8] protocol requires a single response message, where the cryptographic component is realized through a one-time secret, while the timed response is realized through a special function that leaks some bits, but the authentication apparently succeeds with an overwhelming probability. Proving the security of the Hancke-Kuhn protocol requires evaluating this probability.

The Hancke-Kuhn protocol can be viewed as an instance of the CR template, with the challenge $c^{VP}x = x$ and the response $r^{VP}x = x \boxplus H(s.a.w)$.

The one-time secret $H(s.a.w)$ is obtained by applying a public hash function H on the concatenation of Victor and

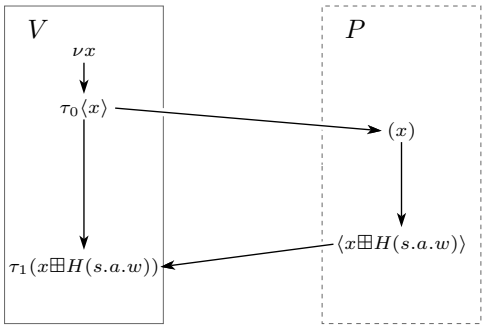

Figure 2. Hancke-Kuhn protocol

Peggy's shared secret s, Peggy's counter a, and Victor's nonce w. The secret s is agreed in to in private, and can be reused in many sessions, whereas the fresh values for a and w are announced publicly at the beginning of each protocol session. While a may be predictable, but must never be reused, w should be unpredictable for each session.

The function $\boxplus : \mathbb{Z}_2^\ell \times \mathbb{Z}_2^{2\ell} \longrightarrow \mathbb{Z}_2^\ell$ is defined bitwise by

$$(x \boxplus h)_i \quad = \quad h_i^{(x_i)} \tag{3}$$

for $i = 1, 2, \ldots, \ell$, and $h = h^{(0)} :: h^{(1)}$, where $h^{(0)}, h^{(1)} \in \mathbb{Z}_2^\ell$.

Clearly, some bits of $x \boxplus h$ can be extracted from $y \boxplus h$, namely, the bits $(x \boxplus h)_i$ where $x_i = y_i$. Moreover, it is possible to learn some bits of $x \boxplus h$ even without having seen x, since the bits $(x \boxplus h)_i$ where $h_i = h_{\ell+i}$ are the same no matter what the value of x is. How likely is it that the attacker, by challenging Peggy, learns enough about the secret $H(s.a.w)$ to impersonate her? How likely is it that Peggy can learn enough about $x \boxplus H(s.a.w)$ to respond before she has seen x? In the next section we introduce some tools to answer these questions.

3 Guessing

We denote by $\left[\Xi \vdash \Theta\right]$ the probability that a set of terms $\Theta \subseteq \mathcal{T}$ can be guessed from a set of terms $\Xi \subseteq \mathcal{T}$. To measure this probability, we assume that the algebra $\mathcal{T}$ of messages is given with a frequency distribution $\mathrm{Prob} : \mathcal{T} \longrightarrow [0,1]$, and that guessing consists of trying the terms one after the other, in order of their probability. To improve his chances, a guessing attacker could engage in Bayesian derivations. However, such derivations, and the distributions that they involve, must be feasibly computable, by the Turing machines from some suitable family $\mathcal{F}$. This means that the probability of guessing t from Ξ is not $\mathrm{Prob}(\Theta \mid \Xi)$,

but

$$\big[\Xi \vdash \Theta\big]_{\mathcal{F}} \quad = \quad \bigvee_{\mathbb{A}\in\mathcal{F}} \mathrm{Prob}(\Theta = \mathbb{A}(\Xi) \mid \Xi) \qquad (4)$$

where $\bigvee$ denotes the supremum.

In the present paper, unless specified otherwise, $\mathcal{F}$ will be the family of Probabilistic Polynomial-time Turing machines (PPT), and we shall usually omit the subscript and write $\big[\Xi \vdash \Theta\big]$ instead of $\big[\Xi \vdash \Theta\big]_{\mathrm{PPT}}$. We also abbreviate $\big[t\big] = \big[\emptyset \vdash t\big]$.

Proposition 3.1 *For all sets of terms Ξ, Γ, Θ, the following inequality holds*

$$\big[\Xi \vdash \Gamma\big] \cdot \big[\Xi, \Gamma \vdash \Theta\big] \quad \leq \quad \big[\Xi \vdash \Gamma, \Theta\big] \qquad (5)$$

and whenever $\big[\Gamma\big] > 0$

$$\big[\Gamma \vdash \Theta\big] \quad \leq \quad \frac{\big[\Gamma, \Theta\big]}{\big[\Gamma\big]} \qquad (6)$$

Remark. Since an algorithm constrained to first guess Γ from Ξ alone, and then Θ from Γ and Ξ may have a significantly worse guessing probability than an algorithm allowed to guess the elements of Γ and Θ in any order, inequalities (5) and (6) may be strict. For conditional probabilities, (5) and (6) are, of course, always equations, and the Bayes' Theorem follows from them.

Definition 3.2 *The* advantage *provided by a set of terms Ξ in computing the terms Θ is the value*

$$\mathsf{Adv}\big[\Xi \vdash \Theta\big] \quad = \quad \big[\Xi \vdash \Theta\big] - \big[\Theta\big]$$

When this advantage is zero, we say that Θ is computationally independent *of Ξ, and write*

$$\begin{aligned} \big[\Xi \perp \Theta\big] \quad &\Longleftrightarrow \quad \mathsf{Adv}\big[\Xi \vdash \Theta\big] = 0 \\ &\Longleftrightarrow \quad \big[\Xi \vdash \Theta\big] = \big[\Theta\big] \end{aligned}$$

Remark . The corresponding relation of *statistical independence*, requiring that $\mathrm{Prob}(\Theta|\Xi) = \mathrm{Prob}(\Theta)$, is well known to be symmetric. In contrast, computational independence is not a symmetric relation. For instance,

- $\big[g^x \vdash x\big] = \big[x\big]$, because it is hard to compute x from g^x, but
- $\big[x \vdash g^x\big] > \big[g^x\big]$, because it is easy to compute g^x from x.

3.1 Probabilistic guards

The idea of the guard relation is that a term t is guarded by one of the guards from $\mathcal{G}$ if whenever t is derived, then at least one of the guards $\Gamma \in \mathcal{G}$ is also derived. In the algebraic model, this was simple enough to state by Definition 2.1. When t can be guessed, then this crude statement needs to be refined: the event that t is guessed must be preceded by the event that some $\Gamma \in \mathcal{G}$ is guessed.

Definition 3.3 *We say that a set of sets of terms $\mathcal{G}$* guards (against guessing) *a term t with respect to a set of terms Υ, and write*

$$\mathcal{G} \quad \mathsf{guards} \quad t \quad \mathsf{within} \quad \Upsilon$$

if for every set of terms $\Xi \subseteq \Upsilon$ such that $\mathsf{Adv}\big[\Xi \vdash t\big] > 0$, we have that

$$\big[\Xi \vdash t\big] \quad \leq \quad \bigvee_{\Gamma\in\mathcal{G}} \big[\Xi \vdash \Gamma\big] \cdot \big[\Xi, \Gamma \vdash t\big] \qquad (7)$$

The following proposition, with its straightforward proof, tells that Definition 3.3 can be viewed as a refinement of Definition 2.1.

Proposition 3.4 *Suppose that the guessing machines $\mathcal{F}$ used in (4) are constrained to never read their random bits, so that guessing boils down to algebraic derivations. Then the guessing guard relation from (7) boils down to the algebraic guard relation from (2).*

Proof. If the Turing machines $\mathcal{F}$ used for guessing do not use randomness, they become Deterministic Polynomial-Time Turing machines (DPT): they construct some terms and apply some equations, in a prescribed order. For each such machine, it is determined with certainty whether it will output a term or not. It follows that the probabilities on the right-hand side of (4), restricted $\mathcal{F} = \mathrm{DPT}$, must be either 0 or 1, and thus

$$\big[\Xi \vdash t\big]_{\mathrm{DPT}} = 1 \quad \Longleftrightarrow \quad \Xi \vdash t \qquad (8)$$

The claim that (8) implies that ((7) $\Longleftrightarrow$ (2)) follows by case analysis.

□

3.2 Partitioned functions and $\boxplus$

In this section we analyze quickly computable functions, like the one used in the Hancke-Kuhn protocol. One requirement of a quickly computable function is the bit dependency of its outputs from its inputs must be partitioned: the i-th block of output bits should only depend on the i-th

block of input bits. Obviously, a function where every bit of output depends on every bit of input has to wait for the last bit of input before it can produce.

Definition 3.5 *We say that a boolean function $f : \mathbb{Z}_2^m \longrightarrow \mathbb{Z}_2^n$ is* partitioned *when*

$$\begin{aligned} m &= m_1 + m_2 + \cdots + m_\ell \\ n &= n_1 + n_2 + \cdots + n_\ell \\ f &= f_1 :: f_2 :: \cdots :: f_\ell \end{aligned}$$

where the inputs and the outputs of each component function $f_i : \mathbb{Z}_2^{m_i} \longrightarrow \mathbb{Z}_2^{n_i}$, for $i = 1, 2, \ldots \ell$ are independent on the inputs and the outputs of all other component functions, in the sense that $\left[x_{\overline{i}}, f_{\overline{i}}(x_{\overline{i}}) \perp f_i(x_i)\right]$, where $\overline{i} = \{j \leq \ell |\ j \neq i\}$.

Although partitioning a function decreases its complexity, it also decreases its security, since it limits the bit propagation. In particular, knowing $f(z)$ helps us guess $f(x)$. In the next section we make this more precise.

3.2.1 Guessing the values of partitioned functions

Proposition 3.6 (a) *Let f be a partitioned function as above, and let $x, z \in \mathbb{Z}_2^m$ be bitstrings with a common block $x_i = z_i$. Then $\left[x, z, f(z) \vdash f(x)\right] \geq 2^{n_i - n}$.*

(b) *Let $f : \mathbb{Z}_2^\ell \longrightarrow \mathbb{Z}_2^\ell$ be* bitwise *partitioned, i.e. $|m_i| = |n_i| = 1$ for all $i \leq \ell$. Then $\left[x, z, f(z) \vdash f(x)\right] \geq 2^{-\Delta(x,z)}$, where $\Delta(x, z) = \#\{i | x \neq z\}$ is the Hamming distance.*

Proof. For (a), $x_i = z_i$ yields $f_i(x_i) = f_i(z_i)$, so we only need to guess at most $n - n_i$ bits. For (b), x_i and z_i are bits, and $n - \Delta(x, z)$ of them are equal, so we only need to guess at most $\Delta(x, z)$ bits. □

A consequence of Prop. 3.6 is that a proximity authentication protocol, implemented using a partitioned function R to compute the response $r^{VP}x = R(s^{VP}, c^{VP}x)$, cannot be secure in an absolute sense, because the response may be guessed with a non-negligible probability from the other responses $r^{VP}z$. Moreover, it seems that the Attacker can always obtain some other responses $r^{VP}z$ by impersonating Victor and issuing challenges $c^{VP}z$ (as an attempt to authenticate the challenge in an authentication protocol leads into a vicious circle).

Lemma 3.7 *A boolean function $f : \mathbb{Z}_2^\ell \longrightarrow \mathbb{Z}_2^\ell$ is bitwise partitioned if and only if for every $x \in \mathbb{Z}_2^\ell$ holds*

$$f(x) = x \boxplus \left(f(0^\ell) :: f(1^\ell)\right) \tag{9}$$

where $\boxplus$ is the Hancke-Kuhn function (3), and $0^\ell, 1^\ell \in \mathbb{Z}_2^\ell$ are the strings of 0s and 1s, respectively.

Proof. Using the definition of bitwise partitioned functions at the first step, and (3) at the second, we get

$$\left(f(x)\right)_i = f_i(x_i) = \left(x \boxplus \left(f(0^\ell) :: f(1^\ell)\right)\right)_i$$

□

Proposition 3.8 *If $f : \mathbb{Z}_2^\ell \longrightarrow \mathbb{Z}_2^\ell$ is a bitwise partitioned function, then the probability that its values can be guessed from one another is at the minimum*

$$\left[x, z, f(z) \vdash f(x)\right] = 2^{-\Delta(z,x)} \tag{10}$$

if and only if for every $i \leq \ell$ holds

$$\left[f_i(0) \perp f_i(1)\right] \quad \textit{and} \quad \left[f_i(1) \perp f_i(0)\right] \tag{11}$$

Proof. If (11) are satisfied, then $x_i \neq z_i$ implies $\left[x_i, z_i, f_i(z_i) \vdash f_i(x_i)\right] = \left[x_i \vdash f_i(x_i)\right]$. On the other hand, by definition, the components of a partitioned function are mutually independent, i.e. $\left[x_{\overline{i}}, z_{\overline{i}}, f_{\overline{i}}(z_{\overline{i}}) \vdash f_i(x_i)\right] = \left[x_i \vdash f_i(x_i)\right]$. Hence

$$\begin{aligned} \left[x, z, f(z) \vdash f(x)\right] &= \prod_{i=1}^{\ell} \left[x, z, f(z) \vdash f_i(x_i)\right] \\ &= \prod_{\substack{i=1 \\ x_i \neq z_i}}^{\ell} \left[x_i \vdash f_i(x_i)\right] \\ &= \prod_{\Delta(z,x)} \frac{1}{2} = 2^{-\Delta(z,x)}. \end{aligned}$$

The other way around, using (10) at the second step, we get

$$\begin{aligned} \prod_{i=1}^{\ell} \left[x, z, f(z) \vdash f_i(x_i)\right] &= \left[x, z, f(z) \vdash f(x)\right] \\ &= 2^{-\Delta(z,x)} \\ &= \prod_{\substack{i=1 \\ x_i \neq z_i}}^{\ell} \left[x_i \vdash f_i(x_i)\right] \end{aligned}$$

which, together with the componentwise independence again, yields (11). □

Remark. In a sense, $x \boxplus (-) : \mathbb{Z}_2^{2\ell} \longrightarrow \mathbb{Z}_2^\ell$ is thus a "one-and-half-way function", since $x \boxplus h$ discloses only one half of the bits of h.

On the other hand, $(-) \boxplus h : \mathbb{Z}_2^\ell \longrightarrow \mathbb{Z}_2^\ell$ is not just an example of a bitwise partitioned function, satisfying the needs of the Hancke-Kuhn protocol, but it is a canonical way to represent such functions.

3.2.2 Guessing $x \boxplus h$

Looking at Prop. 3.8 from the other direction, note that the values of $x \boxplus \left(h^{(0)} :: h^{(1)}\right)$ are easier to guess from one another when $h^{(0)}$ and $h^{(1)}$ are more dependent. E.g., note

$$x \boxplus (y \ :: \ y) = y \qquad x \boxplus (y \ :: \ \neg y) = x \oplus y$$

where $x \oplus y$ is the "exclusive or" operation.

Definition 3.9 *For $x \in \mathbb{Z}_2^\ell$ and $I \subseteq \ell = \{0, 1, 2, \ldots \ell - 1\}$ we define $x^{\circledast I} \in \mathbb{Z}_2^\ell$ to be the bit string obtained by replacing for all $i \in I$ the bits x_i with a "wild card" $\circledast$*

$$x_k^{\circledast I} \quad = \quad \begin{cases} \circledast & \text{if } k \in I \\ x_k & \text{otherwise} \end{cases}$$

For $h = h^{(0)} :: h^{(1)}$, where $h^{(0)}, h^{(1)} \in \mathbb{Z}_2^\ell$ we define the kernel *κh to be the set of places where its first and its second half coincide*

$$\kappa h \quad = \quad \{i \in \ell \mid h_i^{(0)} = h_i^{(1)}\}$$

Proposition 3.10 *For x and h as above holds*

(a) $\left[h \vdash x \boxplus h\right] = 2^{|\kappa h| - \ell}$

(b) $\left[x \vdash x \boxplus h\right] = \left[x^{\circledast \kappa h} \vdash x \boxplus h\right]$

(c) $\left[x, h \vdash x \boxplus h\right] = \left[x^{\circledast \kappa h}, h \vdash x \boxplus h\right]$

Proof. Note that for each $i \in \kappa h$, the bit $(x \boxplus h)_i = h_i^{(0)} = h_i^{(1)}$ does not depend on x_i. This means that $x \boxplus h$ only depends on $x^{\circledast \kappa h}$. □

The following illustrates that guessing $x \boxplus h$ may get unexpectedly subtle.

Proposition 3.11 *For random $x \in \mathbb{Z}_2^\ell$ and $h \in \mathbb{Z}_2^{2\ell}$ and for every $z \in \mathbb{Z}_2^\ell$ holds.*

$$\left[z \boxplus h \vdash x \boxplus h\right] \quad = \quad \left(\frac{3}{4}\right)^\ell$$

Proof. Guessing $x \boxplus h$ from z and $z \boxplus h$ can be modeled as a version of the Monty Hall problem [15], where Monty randomly selects x and h, and the contestant chooses z. Monty then announces $z \boxplus h$ and the contestant must guess $x \boxplus h$.

Consider the case when $\ell = 1$. Monty thus flips three fair coins to pick the secret bits $x, h^{(0)}$ and $h^{(1)}$, while the contestant picks a bit z. Monty then announces $z \boxplus h = h^{(z)}$. Should the contestant now guess that $x \boxplus h = z \boxplus h$, or should he switch to $x \boxplus h = \neg(z \boxplus h)$?

Denote by q the probability that the contestant picks $x \boxplus h = z \boxplus h$. If $h^{(0)} = h^{(1)}$, the contestant wins with this choice, because the value $x \boxplus h$ is the same for every x. Since $h^{(0)}$ and $h^{(1)}$ were randomly chosen, $\mathrm{Prob}(h^{(0)} = h^{(1)}) = \frac{1}{2}$. Otherwise, if $h^{(0)} \neq h^{(1)}$, then $x \boxplus h = z \boxplus h$ holds if and only if $x = z$. Since x is random, $\mathrm{Prob}(x = z) = \frac{1}{2}$, and hence $\mathrm{Prob}(h^{(0)} \neq h^{(1)} \wedge x = z) = \frac{1}{4}$, because $h^{(0)}$, $h^{(1)}$ and x are independent.

The probability that the contestant will make a correct guess is thus

$$q \cdot \left(\mathrm{Prob}\left(h^{(0)} = h^{(1)}\right) + \mathrm{Prob}\left(h^{(0)} \neq h^{(1)} \wedge x = z\right)\right) \quad = \quad \frac{3q}{4}$$

To maximize this probability, the contestant needs $q = 1$, and should thus stick[1] with Monty's bit $z \boxplus h$.

The inductive step to $\ell + 1$ is left to the reader. □

Remark. Taken together, Propositions 3.11 and 3.8 imply that there are situations when

$$\left[z \boxplus h \vdash x \boxplus h\right] \quad > \quad \left[x, z, z \boxplus h \vdash x \boxplus h\right] \qquad (12)$$

Formally, this inequality is satisfied whenever $\Delta(z, x) > (2 - \log 3)\ell$, which happens with a probability of about .42. But how can it be easier to guess $x \boxplus h$ from $z \boxplus h$ alone, than together with x and z?

In 3.11, when x and z are not available, the optimal guessing strategy is to always try $(x \boxplus h)_i = (z \boxplus h)_i$. In 3.8, when x and z are available, the optimal guessing strategy can be to test for every i whether $x_i = z_i$ and to try $(x \boxplus h)_i = (z \boxplus h)_i$ if it is, or to randomize $(x \boxplus h)_i$ otherwise. But whenever $\left[h_i^{(0)} \perp h_i^{(1)}\right]$, as assumed in 3.8, the guessing algorithm can randomize $(x \boxplus h)_i$ by setting $(x \boxplus h)_i = (z \boxplus h)_i$. Whether x and z are known or not, the guessing algorithm can thus proceed in the same way, by always trying $(x \boxplus h)_i = (z \boxplus h)_i$.

This strategy is, of course, more successful in the cases when the Hamming distance between x and z happens to be smaller. When x and z are known, the probability of its success can be precisely evaluated, as it was done in 3.8. When x and z are not known, we can, in principle, only estimate the expected (average) value of this probability. Prop. 4.5 below shows that this is just what was done in 3.11, albeit implicitly. The situations when $\Delta(z, x) > (2 - \log 3)\ell$ and (12) hold are just those when the probability of a successful guess of $x \boxplus h$ from $z \boxplus h$, for the specific x and z, happens to be lower than its expected value, averaged over x and z.

[1] This solution is in contrast from the original Monty Hall problem [15], where it is advantageous to switch. The reasoning is, however, quite similar.

4 Security of the Hancke-Kuhn protocol

We quantify the security of the Hancke-Kuhn protocol by evaluating the probability $\mathrm{Prob}(\mathsf{crp})$, that the sequence of events in a complete protocol run validates (crp). In order to evaluate this probability, we analyze the probability that (crp) fails. How can it happen that Victor observes a satisfactory sequence

$$\mathcal{V} \quad = \quad (\nu x)_V \triangleright \tau_0\langle x\rangle_V \triangleright \tau_1 \left(x \boxplus H(s.a.w)\right)_V \quad (13)$$

but that the event

$$\mathcal{O} \;=\; \tau_0\langle x\rangle_V \triangleright (x)_P \triangleright \\ \langle x \boxplus H(s.a.w)\rangle_P \triangleright \tau_1 \left(x \boxplus H(s.a.w)\right)_V \quad (14)$$

did not take place? The idea that we pursue is that this can happen either when

$\mathcal{A}$**:** the responder is not Peggy, but an $\mathcal{A}$ttacker, who does not know the secret s, or when

$\mathcal{E}$**:** the responder is Peggy, but she responded $\mathcal{E}$arlier than she received the challenge.

We shall us $\mathcal{A}$ and $\mathcal{E}$ as generic notations for the runs where the above events occur. In other words, the event $\neg\mathcal{O}$ is decomposed as a disjoint union of $\mathcal{A}$ and $\mathcal{E}$. We proceed to evaluate $\mathrm{Prob}(\mathcal{A})$ and $\mathrm{Prob}(\mathcal{E})$. This will be done by determining probabilistic guards of the response in the term contexts of $\mathcal{A}$ and $\mathcal{E}$, Defn. 2.2.

Response token

Recall that Peggy's response token $H(s.a.w)$ is derived from the shared secret s, Peggy's counter a (predictable but never reused), and Victor's nonce w (unpredictable), using a secure public hash function H.

The requirements that s and w are random and that H is secure are needed to assure that the families

$$F \quad = \quad \{f_s = H(s.\_.\_)\}_{s\in\mathbb{Z}_2^m} \quad (15)$$
$$G \quad = \quad \{g_w = H(\_.\_.w)\}_{w\in\mathbb{Z}_2^n} \quad (16)$$

are pseudorandom [7]. For our purpose, these requirements boil down to the fact that a PPT cannot distinguish between the outputs of a member of F and the values of a random variable without knowing the index s; and ditto for G and w.

The pseudo-randomness of F means that the Attacker who is attempting in a run $\mathcal{A}$ to impersonate Peggy should not be able to guess any bits of $H(s.a.w)$ without knowing s, even if a, w and some other bits of $H(s.a.w)$ are known.

The reason for requiring that G be pseudo-random is more subtle. Suppose that Peggy is trying in a run $\mathcal{E}$ to respond early, without having seen x. According to Prop. 3.10(a), if $|\kappa H(s.a.w)| = k$ then Peggy only needs to guess $\ell - k$ bits of x in order to respond. If $k = \ell$, she does not need x at all! A dishonest Peggy will thus try to choose a in such a way to make $\kappa H(s.a.w)$ as large as possible. The pseudo-randomness of G means that she should not be able to control $H(s.a.w)$ without knowing w.

For brevity, we often write h^{VP} instead of $H(s.a.w)$.

4.1 Guards in undesired runs

In order to evaluate $\mathrm{Prob}(\mathsf{crp})$, we need to determine the probability that the correct response $x \boxplus h^{VP}$ is guessed in the undesired runs $\mathcal{A}$ and $\mathcal{E}$. More precisely, what can be guessed in the term contexts $\mathcal{A}(x \boxplus h^{VP})$ and $\mathcal{E}(x)$, as defined in 2.2? — The following lemmas simplify this question.

Lemma 4.1 *(a) Let $\mathcal{A}$ be an attack run with a long term secret s, Peggy's counter a, Victor's nonce w, and $\mathcal{A}$ttacker's challenge z, for which he obtains the response $z \boxplus h^{VP}$, where $h^{VP} = H(s.a.w)$. Then for any $\Xi \subseteq \mathcal{A}(x \boxplus h^{VP})$ holds*

$$\left[\Xi \vdash x \boxplus h^{VP}\right] = \left[\Xi \cap \left\{s, a, w, x, z, z \boxplus h^{VP}\right\} \vdash x \boxplus h^{VP}\right]$$

(b) Let $\mathcal{E}$ be a run with a long term secret s, Peggy's counter a, Victor's nonce w, and where Peggy responds early. Then for any $\Xi \subseteq \mathcal{E}(x)$

$$\left[\Xi \;\vdash\; x \;\boxplus\; h^{VP}\right] \quad = \quad \left[\Xi \cap \{s, a, w\} \;\vdash\; x \;\boxplus\; h^{VP}\right]$$

Proof (a). Recall that $\mathcal{A}(x \boxplus h^{VP})$ is the union of the contexts observed by the possible participants in the run $\mathcal{A}$, before $x \boxplus h^{VP}$ is known. Besides s, known by Victor and Peggy, and a, w and x, announced publicly but never reused, the context $\mathcal{A}(x \boxplus h^{VP})$ thus also contains a single additional challenge z, issued by the $\mathcal{A}$ttacker, and the corresponding response $z \boxplus h^{VP}$ (provided by Peggy before she receives Victor's challenge x).

Moreover, the $\mathcal{A}$ttacker may issue a family $Y \subseteq \mathbb{Z}_2^\ell$ of additional challenges to Peggy, and even construct and provide some suitable nonces $\{w_y\}_{y\in Y}$. To each new challenge, Peggy will respond with $y \boxplus h_y$, where the response token $h_y = H(s.a_y.w_y)$ is derived *using a new value of the counter* a_y. The whole family of responses $\{h_y\}_{y\in Y}$ will not uncover anything about h^{VP} because the family F

from (15) is pseudorandom, i.e. because $\lambda y.f_s(a_y, w_y) = H(s.a_y.w_y)$ seems random as soon as $a : Y \longrightarrow \mathbb{Z}_2^\ell$ varies.

In summary, the term context is thus

$$\mathcal{A}(x \boxplus h^{VP}) = \{s, a, w, x, z, z \boxplus h^{VP}\} \cup \{y, a_y, w_y, y \boxplus h_y \mid h_y = H(s.a_y.w_y) \wedge y \in Y\}$$

for some $Y \subseteq \mathbb{Z}_2^\ell$, where $a : Y \to \mathbb{Z}_2^\ell$ is injective, and $w : Y \to \mathbb{Z}_2^n$ arbitrary. The pseudo-randomness of $F = \{H(s.\_.\_)$ implies $[y, a_y, w_y, y \boxplus h_y \perp x \boxplus h^{VP}]$, which further implies that for any $\Xi \subseteq \mathcal{A}(x \boxplus h^{VP})$

$$\{s, a, w, z, z \boxplus h^{VP}\} \cap \Xi = \emptyset \implies [\Xi \perp x \boxplus h^{VP}]$$

and we are done.

The reasoning towards **(b)** is slightly simpler, elaborating the fact that obtaining one challenge tells nothing about another one. The details are left to the reader. □

Besides the preceding lemma, the main results of this section, stated below, depends on the following simple equations:

Lemma 4.2 *(a)* $[x \boxplus h^{VP}] = [x \vdash x \boxplus h^{VP}] = [z \vdash x \boxplus h^{VP}] = [x, z \vdash x \boxplus h^{VP}] = 2^{-\ell}$.

(b) $[\Sigma, \Omega \vdash x \boxplus h^{VP}] = [\Omega \vdash x \boxplus h^{VP}]$ *when* $\Sigma \subset \{s, a, w\}$ *and* $\Omega \subseteq \{x, z, z \boxplus h^{VP}\}$.

(c) $[a, w, s, x, \Omega \vdash x \boxplus h^{VP}] = [a, w, s, x^{\circledast \kappa h^{VP}} \vdash x \boxplus h^{VP}] = 1$ *where* $\Omega \subseteq \{z, z \boxplus h^{VP}\}$.

Proposition 4.3 $\{\{s\}, \{z \boxplus h^{VP}\}\}$ guards $x \boxplus h^{VP}$ within $\mathcal{A}(x \boxplus h^{VP})$

Proof. The claim follows from the fact that each $\Xi \subseteq \mathcal{A}(z \boxplus h^{VP})$ satisfies at least one of the following inequalities:

$$[\Xi \vdash x \boxplus h^{VP}] \leq [\Xi \vdash s] \cdot [\Xi, s \vdash x \boxplus h^{VP}] \quad (17)$$

$$[\Xi \vdash x \boxplus h^{VP}] \leq [\Xi \vdash z \boxplus h^{VP}] \cdot [\Xi, z \boxplus h^{VP} \vdash x \boxplus h^{VP}] \quad (18)$$

But this is easily shown by case analysis, using Lemma 4.2, since according to Lemma 4.1(a) it suffices to consider the subsets Ξ of $\{s, a, w, x, z, z \boxplus h^{VP}\}$. □

Proposition 4.4 $\{\{x^{\circledast \kappa h^{VP}}\}\}$ guards $x \boxplus h^{VP}$ within $\mathcal{E}(x)$

Proof. The claim is that each $\Xi \subseteq \mathcal{E}(x)$ satisfies

$$[\Xi \vdash x \boxplus h^{VP}] \leq [\Xi \vdash x^{\circledast \kappa h^{VP}}] \cdot [\Xi, x^{\circledast \kappa h^{VP}} \vdash x \boxplus h^{VP}] \quad (19)$$

Lemma 4.1(b) says that suffices to consider $\Xi \subseteq \{s, a, w\}$. But the pseudo-randomness assumption implies that $[\Xi \vdash x \boxplus h^{VP}] = 2^{-\ell}$ whenever Ξ is a proper subset. So (19) holds trivially. For $\Xi = \{s, a, w\}$, using Prop. 3.10 and Lemma 4.2, we have $[\Xi \vdash x \boxplus h^{VP}] = [\Xi \vdash x^{\circledast \kappa h^{VP}}] = 2^{|\kappa h^{VP}| - \ell}$ and on the other hand $[\Xi, x^{\circledast \kappa h^{VP}} \vdash x \boxplus h^{VP}] = 1$. Hence (19).

□

4.2 Bounds on undesired runs

We are now ready to compute the probability that the $\mathcal{A}$ttacker can authentication, or that the response can be $\mathcal{E}$arlier than the challenge. By Proposition 4.3, for a given x, the probability that an $\mathcal{A}$ttacker can violate authentication is bounded above by

$$[\Phi \vdash s] \cdot [\Phi, s \vdash x \boxplus h^{VP}] \text{ or by}$$
$$[\Phi \vdash z \boxplus h^{VP}] \cdot [\Phi, z \boxplus h^{VP} \vdash x \boxplus h^{VP}]$$

where $\Phi = \{a, w, x, z, z \boxplus h^{VP}\}$

By Proposition 4.4, the probability that Peggy can respond $\mathcal{E}$arly, again for a given x, is bounded above by

$$[s, a, w \vdash x^{\circledast \kappa h^{VP}}] \cdot [s, a, w, x^{\circledast \kappa h^{VP}} \vdash x \boxplus H(s, a, w)]$$

Note that in the attack run $\mathcal{A}$, the $\mathcal{A}$ttacker cannot learn x until after she has created z. The distribution of z is thus independent from that of x.

Proposition 4.5 *Suppose that Victor's challenge* $x \in \mathbb{Z}_2^\ell$ *is chosen randomly, according to the uniform distribution; and that the Attacker, before receiving* x*, can pick her own challenge* $z \in \mathbb{Z}_2^\ell$ *and obtain a single response* $z \boxplus h^{VP}$*. Then the expected probability* $\mathrm{Prob}(\mathcal{V} \mid \mathcal{A})$ *that the Attacker can guess the correct response* $x \boxplus h^{VP}$*, and deceive Victor is*

$$\int_{x \in \mathbb{Z}_2^\ell} [x, z, z \boxplus h^{VP} \vdash x \boxplus h^{VP}] = \left(\frac{3}{4}\right)^\ell$$

Proof. Since $\mathrm{Prob}(x \in \mathbb{Z}_2^\ell) = 2^{-\ell}$ by assumption, and $[x, z, z \boxplus h^{VP} \vdash x \boxplus h^{VP}]$ by (10), it follows that

$$\int_{x \in \mathbb{Z}_2^\ell} [x, z, z \boxplus h^{VP} \vdash x \boxplus h^{VP}] = \sum_{x \in \mathbb{Z}_2^\ell} 2^{-\ell} [x, z, z \boxplus h^{VP} \vdash x \boxplus h^{VP}] = 2^{-\ell} \cdot \sum_{i=0}^{\ell} \binom{\ell}{i} 2^{-i} = 2^{-\ell} \cdot \frac{3^\ell}{2^\ell} = \left(\frac{3}{4}\right)^\ell$$

□

Proposition 4.6 *Suppose that Victor's challenge $x \in \mathbb{Z}_2^\ell$ and his random nonce $w \in \mathbb{Z}_2^m$ (announced prior to the challenge-response exchange) are chosen independently and according to the uniform distribution. The expected probability that Peggy can guess and send her response $x \boxplus h^{VP}$, where $h^{VP} = H(s.a.w)$, before she receives the challenge x is*

$$\int_{x \in \mathbb{Z}_2^\ell} \left[s, a, w \vdash x \boxplus h^{VP} \right] \;=\; \left(\frac{3}{4}\right)^\ell$$

Proof. Since $\left[s, a, w \perp x \right]$ holds by assumption, $\left[s, a, w \vdash x \boxplus h^{VP} \right] = \left[h^{VP} \vdash x \boxplus h^{VP} \right]$ follows, since s, a, w can only be useful to derive $h^{VP} = H(s.a.w)$. But Prop. 3.10(a) then implies that $\left[s, a, w \vdash x \boxplus h^{VP} \right] = 2^{i-\ell}$, where $i = |\kappa h^{VP}|$. Since H is assumed to be a pseudorandom function, because of $w \in \mathbb{Z}_2^\ell$, Peggy cannot distinguish $h^{VP} = H(s.a.w)$ from a random variable $\eta \in \mathbb{Z}_2^{2\ell}$. The expected value that she will guess $x \boxplus h^{VP}$ must therefore be averaged over the possible values of η, and hence

$$\int_{\eta \in \mathbb{Z}_2^{2\ell}} \int_{x \in \mathbb{Z}_2^\ell} \left[\eta \vdash x \boxplus \eta \right] = \sum_{\eta \in \mathbb{Z}_2^\ell} \sum_{x \in \mathbb{Z}_2^\ell} 2^{-\ell} \left[\eta \vdash x \boxplus \eta \right] = 2^{-\ell} \cdot \sum_{i}^{\ell} \binom{\ell}{i} 2^{i-\ell} = 2^{-2\ell} \cdot 3^\ell \;=\; \left(\frac{3}{4}\right)^\ell$$

□

Theorem 4.7 *Suppose that it is assured that*

- $\mathrm{Prob}(\mathcal{A}), \mathrm{Prob}(\mathcal{E}) < C_1 < 1$, *i.e. not every response is from the Attacker, and not every response is too Early;*
- $\mathrm{Prob}(\mathcal{V}) > C_2 > 0$, *i.e. Victor does not always reject.*

Then the probability that the Hancke-Kuhn protocol implements (crp) *satisfies*

$$\mathrm{Prob}(\mathsf{crp}) \;\geq\; 1 - \frac{2C_1}{C_2} \left(\frac{3}{4}\right)^\ell$$

Proof. As explained in the beginning of this section, $\mathrm{Prob}(\mathsf{crp})$ is the conditional probability $\mathrm{Prob}(\mathcal{O} \mid \mathcal{V})$, where $\mathcal{O}$ is a desired run and $\mathcal{V}$ Victor's observation, as defined in (13-14). Since the undesired runs $\neg\mathcal{O}$ decompose into a disjoint union of $\mathcal{A}$ and $\mathcal{E}$, it follows that

$$\mathrm{Prob}(\mathsf{crp}) \;=\; 1 - \mathrm{Prob}(\mathcal{A} \mid \mathcal{V}) - \mathrm{Prob}(\mathcal{E} \mid \mathcal{V}) \qquad (20)$$

Since Bayes' Theorem gives

$$\mathrm{Prob}(\mathcal{A} \mid \mathcal{V}) \;=\; \frac{\mathrm{Prob}(\mathcal{V} \mid \mathcal{A}) \;\cdot\; \mathrm{Prob}(\mathcal{A})}{\mathrm{Prob}(\mathcal{V})}$$

we derive from Prop. 4.5 and the hypothesis

$$\frac{\mathrm{Prob}(\mathcal{V} \mid \mathcal{A}) \;\cdot\; \mathrm{Prob}(\mathcal{A})}{\mathrm{Prob}(\mathcal{V})} \leq \frac{C_1}{C_2} \left(\frac{3}{4}\right)^\ell \qquad (21)$$

Similarly, from Prop. 4.6 and the hypothesis we derive

$$\frac{\mathrm{Prob}(\mathcal{V} \mid \mathcal{E}) \;\cdot\; \mathrm{Prob}(\mathcal{E})}{\mathrm{Prob}(\mathcal{V})} \leq \frac{C_1}{C_2} \left(\frac{3}{4}\right)^\ell \qquad (22)$$

The result follows by substituting (21) and (22) into (20). □

Remark. The presented analysis suggests a strategy for reducing attackers' chances to break freshness. If Victor and Peggy agree to abort if the kernel size of h^{VP} is over a certain threshold, then guessing $x \boxplus h^{VP}$ without knowing x becomes harder.

5 Conclusion

In this paper we have given a structured probabilistic proof of the security of the Hancke-Kuhn distance bounding protocol. It not only provides a proof of security of Hancke-Kuhn in particular, but provides a template for reasoning about probabilistic challenge-response in general:

- Determine all messages that can be sent before the response.
- Compute the probability of deriving the response from each subset of the set of messages computed in the first step.
- Use the results of the second step to derive the guards relation for both challenge and and secret key.
- Use the results of the second and third step to derive the probability that a protocol satisfies the desired challenge-response template.

One advantage of our methodology is that it can teach us things about a protocol beyond the main security results we start out to prove. For example, we were able to show that, on the advantage, that an outside attacker gains no advantage in guessing the response from knowing Victor's challenge. Comparing Propositions 3.11 and 4.5 we see that both probabilities are on the average the same. All the attacker obtains from knowing the challenge is the knowledge

of how successful her guess is likely to be in a given instance. The ability to detect such subtle interplay between different types of guessing can be very useful when analyzing protocols like these.

But the main contribution of this paper is that it gives a methodology that can be applied to complex probabilistic functions that cannot easily be reasoned about in an algebraic model. We expect it to have applications beyond timed challenge and response; for example, another class of pervasive protocols that employs *human-verifiable channels* [16, 11] relies upon weak hash functions that satisfy probabilistic security guarantees, and are currently working on applying our framework there. Finally, our ultimate goal is the develop a *probabilistic* Protocol Derivation Logic that could be used to reason formally about protocols with these types of probabilistic guarantees. This paper represents the first step in that direction.

References

[1] Martin Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. of Cryptology*, 15(2):103–127, 2002.

[2] Stefan Brands and David Chaum. Distance-bounding protocols. In *EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, pages 344–359, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.

[3] Iliano Cervesato, Catherine Meadows, and Dusko Pavlovic. An encapsulated authentication logic for reasoning about key distribution protocols. In Joshua Guttman, editor, *Proceedings of CSFW 2005*, pages 48–61. IEEE, 2005.

[4] Danny Dolev and Andrew C. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198–208, 1983.

[5] Javier Thayer Fabrega, Jonathan Herzog, and Joshua Guttman. Strand spaces: What makes a security protocol correct? *Journal of Computer Security*, 7:191–230, 1999.

[6] Oded Goldreich. *Foundations of Cryptography. Volume I: Basic Tools*. Cambridge University Press, 2000.

[7] Oded Goldreich. Pseudorandom generators: A primer. `http://www.wisdom.weizmann.ac.il/oded/prg-primer.html/`, 2008.

[8] Gerhard P. Hancke and Markus G. Kuhn. An RFID distance bounding protocol. In *SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 67–73, Washington, DC, USA, 2005. IEEE Computer Society.

[9] Catherine Meadows and Dusko Pavlovic. Deriving, attacking and defending the GDOI protocol. In Peter Ryan, Pierangela Samarati, Dieter Gollmann, and Refik Molva, editors, *Proceedings of ESORICS 2004*, volume 3193 of *Lecture Notes in Computer Science*, pages 53–72. Springer Verlag, 2004.

[10] Catherine Meadows, Radha Poovendran, Dusko Pavlovic, LiWu Chang, and Paul Syverson. Distance bounding protocols: authentication logic analysis and collusion attacks. In R. Poovendran, C. Wang, and S. Roy, editors, *Secure Localization and Time Synchronization in Wireless Ad Hoc and Sensor Networks*. Springer Verlag, 2006.

[11] Long H. Nguyen and Andrew W. Roscoe. Authenticating ad hoc networks by comparison of short digests. *Journal of Information and Computation*, 206, Issues 2-4:250–271, Feb-Apr 2008.

[12] Dusko Pavlovic and Catherine Meadows. Deriving secrecy properties in key establishment protocols. In Dieter Gollmann and Andrei Sabelfeld, editors, *Proceedings of ESORICS 2006*, volume 4189 of *Lecture Notes in Computer Science*. Springer Verlag, 2006.

[13] Ronald L. Rivest. On the notion of pseudo-free groups. In Moni Naor, editor, *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 505–521. Springer, 2004.

[14] Patrick Schaller, Benedikt Schmidt, David Basin, and Srdjan Capkun. Modeling and verifying physical properties of security protocols for wireless networks. In *In Proceedings of the IEEE Computer Security Foundations Symposium*. IEEE Computer Society Press, 2009. to appear.

[15] Steve Selvin. On the Monty Hall problem. *American Statistician*, 29(3):134, August 1975. (letter to the editor).

[16] Ford Long Wong and Frank Stajano. Multichannel security protocols. *IEEE Pervasive Computing*, 6(4):31–39, 2007.