Symbolic Probabilistic Analysis of Off-line Guessing

Bruno Conchinha and David Basin
Information Security Institute
ETH Zürich
Zürich, Switzerland
{*bruno.conchinha, basin*}*@inf.ethz.ch*

Carlos Caleiro
Security and Quantum Information Group
Instituto Superior Técnico
Lisbon, Portugal
ccal@math.ist.utl.pt

***Abstract*—We introduce a probabilistic framework for analyzing security protocols. Our framework provides a general method for expressing properties of cryptographic primitives that is capable of formalizing weaknesses of the primitives such as leaking partial information about a message or the use of weak algorithms for random number generation. These properties can be used to find attacks and estimate their probability of success. Under assumptions about the accuracy of the properties considered, we show that our estimates differ negligibly from the real-world probability of success.**

We illustrate the usefulness of our approach by modeling non-trivial properties of RSA encryption and using them to estimate the success probability of off-line guessing attacks on the EKE [1] protocol. These attacks were previously outside the scope of symbolic methods.

***Keywords*-probability; off-line guessing; equational theories; computational soundness;**

I. Introduction

Cryptographic protocols play an important role in securing distributed computation and it is crucial that they work correctly. Symbolic verification approaches are usually based on the *Dolev-Yao model*: messages are represented by terms in a term algebra, cryptography is assumed to be perfect, and properties of cryptographic operators are formalized equationally [2]. This strong abstraction eases analysis, and numerous successful verification tools rely on it [3]. However, it may not accurately represent an attacker's capabilities. As a consequence, broad classes of attacks that rely on cryptanalysis or weaknesses of cryptographic primitives may fall outside the scope of such methods.

Proving security by reasoning more directly about probabilities, as in so-called *computational approaches* [4], [5], yields much stronger security guarantees. However, it requires long and error-prone hand-written proofs to establish the security of given protocols, using specific cryptographic primitives.

Given the limitations of each of these approaches, much effort has been devoted to get the best of both worlds: strong security guarantees that can automatically proved, or at least whose proofs can be machine-checked. One can distinguish between two distinct lines of research in this direction: (1) obtaining computational soundness results for symbolic methods, or (2) developing techniques that reason directly with the computational models. A good survey of such results is provided in [6].

Computational soundness results have been much sought after since Abadi and Rogaway's seminal paper [7]. They aim at proving that, under certain assumptions, symbolic security implies computational security, whereby protocols that are secure against a Dolev-Yao attacker are also secure against the much more powerful adversary of the computational world. Such results now exist for length revealing and same-key revealing encryption systems [8], active attackers [9], encryption with composed keys [10] and hash functions [11], to name but a few. The limitations of these results include the typically very strong assumptions on the cryptographic primitives, the often-made assumption that messages are tagged so that their structure is known to any observer, and the difficulty of extending the results to new cryptographic primitives, which usually involves redoing most of the work.

The second line of research aims to obtain, automatically where possible, a sequence of property-preserving transformations between game-theoretic problem formulations, as is often done in computational security proofs. The original ideas in this direction [12], [13] have been developed into tools like CryptoVerif [14] and CertiCrypt [15]. These tools can be used to prove protocols correct in the computational world. Moreover, when they succeed, they can provide upper bounds on the probability of a successful attack, based on the probabilities of the attacker winning each of the games used to express security properties of the cryptographic primitives.

In this paper we present a fundamentally new approach to strenghtening the guarantees provided by symbolic methods. Building upon the ideas first presented in [16], we incorporate probabilistic reasoning directly into a symbolic method, by associating each term (representing a message) in the term algebra to a random variable whose values are bitstrings. We express properties of the cryptographic primitives used by using an equational theory, as is standard in symbolic methods. Moreover, we also formalize typing information, which associates terms with bitstrings and operators with functions over bitstrings. Our bitstring types are used to represent random data generation, properties

of cryptographic operators, and attacker capabilities: For instance, in our example we will use *name types* to represent the random generation of an RSA public-key. While it is unfeasible for an attacker to check the validity of an RSA public-key, he can at least check that the exponent is odd and that the modulus has no small prime factors. These capabilities will be expressed by using *test types*. However, unlike most computational soundness results, we do not assume any explicit tagging in messages. [1]

This approach thus allows us to reason about an attacker that is more powerful than that considered in the standard Dolev-Yao model. In contrast to [16], we focus on efficiency rather than expressiveness: our framework is thus better suited to represent properties of cryptographic primitives that are useful in real-world attacks, while greatly reducing both the theoretical and computational difficulty of computing probabilities.

Our approach is in a sense dual to the two lines of formal-methods research described above: Rather than assuming very strong security properties of cryptographic primitives and using them to prove security, we explicitly describe an attacker's knowledge about the cryptographic primitives and the random number generation algorithms being used. This knowledge (or these properties of the cryptographic primitives) can then be used both to find attacks and to estimate their probability.

As one might expect, there is a tradeoff between efficiency and accuracy when computing a probability distribution for the random variables associated to terms in our framework. The probability distribution we proposed is computed from the equational theory and the type properties of cryptographic functions described in the model. It should therefore be understood as an approximation of the real world intended to represent an attacker's capabilities, while still being efficiently computable. We show that, under strong but reasonable assumptions on the cryptographic primitives used, our estimates of the probability of success of the attacks we describe are negligibly different from those of the real world. Perhaps unsurprisingly, our assumptions resemble computational soundness results for (symbolic) static equivalence, such as those discussed, e.g., in [17]–[19]. However, our notions are flexible enough to take into account the properties of cryptographic primitives that we express. In other models, such properties can typically violate the security properties required for computational soundness. [2]

We illustrate the usefulness of our approach[3] by showing how to use it to analyze the security of protocols against off-line guessing attacks. Given the pervasive use of human-picked (and often weak) passwords, off-line guessing attacks are a major concern in security protocol analysis and have been the subject of much research, using symbolic [20]–[22] and computational approaches [23], or relating the two via computational soundness results [19], [24]. We show that our framework can be used to straightforwardly represent non-trivial properties of cryptographic primitives like the redundancy of RSA keys We will use these properties to find off-line guessing attacks on the EKE protocol [1] and estimate their success probabilities. Although these problems are well-known, their analysis was previously outside the scope of symbolic methods. Further applications of this approach may well include modeling and reasoning about differential cryptanalysis or side-channel attacks [25] as well as short-string authentication and distance-bounding protocols.

We proceed as follows. In Section II we introduce terms and their probabilistic interpretation. In Section III we introduce the building blocks of our attacker model: equational theories, type interpretation functions, and property statements about the cryptographic functions. In Section IV we formalize and illustrate off-line guessing in our attacker model. In Section V we establish properties of our approximate probabilistic model, leading up to the proof of our soundness theorem in Section VI. We make comparisons and draw conclusions in Section VII.

II. Basic definitions

A. Syntax.

A *signature* $\Sigma = \biguplus_{n\in\mathbb{N}} \Sigma_n$ is a finite set of function symbols, where Σ_i contains the functions symbols of arity i. For each $f \in \Sigma$, the function $\mathsf{ar}\colon \Sigma \to \mathbb{N}$ returns the arity $\mathsf{ar}(f)$ of f. Given a set X, $T_\Sigma(X)$ is the set of Σ-*terms over* X, *i.e.*, the smallest set such that $X \subseteq T_\Sigma(X)$ and $f(t_1,\ldots,t_n) \in T_\Sigma(X)$ for all $t_1,\ldots,t_n \in T_\Sigma(X)$ and all $f \in \Sigma_n$.

Although we consider an untyped signature, we shall also model relevant type information. To this end, we assume given a fixed a set $\mathcal{NT}$ of *name types* and, for each $T \in \mathcal{NT}$, a countably infinite set $\mathcal{N}_T$ of *names of type* T. The set of *names* is given by $\mathcal{N} = \biguplus_{T\in\mathcal{NT}} \mathcal{N}_T$. Given a name $a \in \mathcal{N}$, we denote by $type(a)$ the unique $T \in \mathcal{NT}$ such that $a \in \mathcal{N}_T$. Intuitively a name stands for a random value that is sampled from its type $type(a)$ by a probabilistic algorithm.

Example II.1. The signature $\Sigma^{\mathcal{DY}}$ is used to represent the cryptographic primitives present in simple Dolev-Yao models containing a hash function h, a pairing function pair, projection functions π_1 and π_2, and symmetric encryption and decryption. It is given by $\Sigma^{\mathcal{DY}} = \Sigma_1^{\mathcal{DY}} \cup \Sigma_2^{\mathcal{DY}}$, where $\Sigma_1^{\mathcal{DY}} = \{\mathsf{h}, \pi_1, \pi_2\}$ and $\Sigma_2^{\mathcal{DY}} = \left\{ \{\!|\cdot|\!\}_\cdot, \{\!|\cdot|\!\}_\cdot^{-1}, \langle\cdot,\cdot\rangle \right\}$.

We consider the following name types: pw, representing weak passwords (e.g., a human-chosen password); $\mathsf{sym\_key}$,

[1] **(FLAG: Please check this!!)**

[2] **(FLAG: Please check this!!)**

[3] **(FLAG: approach versus method versus framework. Do you need all 3? I believe you introduced framework in the last day or two. Maybe approach with a method is enough?)**

representing symmetric keys; and text, representing plaintexts. The set of names is then given by $\mathcal{N} = \mathcal{N}_{\mathsf{pw}} \uplus \mathcal{N}_{\mathsf{sym\_key}} \uplus \mathcal{N}_{\mathsf{text}}$, and the set of terms we consider in this model is $T_{\Sigma^{\mathcal{DY}}}(\mathcal{N})$.

We define the set $sub(t)$ of *subterms* of a term $t \in T_\Sigma(X)$ as usual: $sub(t) = \{t\}$ if $t \in X$ and $sub(t) = \{t\} \cup (\bigcup_{i=1}^n sub(t_i))$ if $t = f(t_1, \ldots, t_n)$. The set $names(t) = sub(t) \cap \mathcal{N}$ is the set of names occurring in a term $t \in T_\Sigma(\mathcal{N})$. We extend these definitions naturally to sets of terms: for $S \subseteq T_\Sigma(\mathcal{N})$, we define $sub(S) = \bigcup_{t \in S} sub(t)$ and $names(S) = \bigcup_{t \in S} names(t)$. Given a term t and $p \in \mathbb{N}^*$, we denote the *subterm of* t *at position* p by $t|_p$, where $t|_\epsilon = t$ and, for $t = f(t_1, \ldots, t_n)$, $t|_{i.p} = t_i|_p$ for $i \in \{1, \ldots, n\}$. Here $i.p$ denotes the sequence of integers obtained by prepending i to the sequence p.

Let $\mathcal{V}$ be a countably infinite set of variables, disjoint from $\mathcal{N}$. We use the standard notion of *substitution* as a partial function $\sigma \colon X \nrightarrow T_\Sigma(\mathcal{N})$. We will use both substitutions on names and substitutions on variables. We abuse notation by using the same symbol σ for a substitution and its homomorphic extension to $T_\Sigma(\mathcal{N})$. As usual, we write $t\sigma$ instead of $\sigma(t)$.

We will represent the concrete knowledge of an attacker using the notion of frame [26]. A *frame* is a pair $(\tilde{n}, \sigma)$, written $\nu\tilde{n}.\sigma$, where $\tilde{n} \subseteq \mathcal{N}$ is a finite set of names and $\sigma \colon \mathcal{V} \nrightarrow T_\Sigma(\mathcal{N})$ is a substitution with finite domain. Given a frame $\phi = \nu\tilde{n}.\sigma$, we define $T_\phi = T_\Sigma((\mathcal{N} \setminus \tilde{n}) \cup dom(\sigma))$. We say that terms in T_ϕ are ϕ*-recipes*. Intuitively, the names in $\tilde{n}$ represent randomly generated nonces unknown to the attacker, and the terms in the range of σ represent the messages learned by the attacker, for instance by eavesdropping on the network. A term t can be *constructed* from ϕ if there is a ϕ-recipe ζ such that $\zeta\sigma = t$. Thus, the set of *terms constructible from* ϕ is $\sigma[T_\phi]$.

B. Term interpretation.

As usual in computational scenarios, we will consider models based on bitstrings. Let $\mathcal{B} = \{0,1\}^*$ and $\mathcal{B}^\emptyset = \emptyset$. For $n, m \in \mathbb{N}$, let $\mathcal{B}^n = \{0,1\}^n$, $\mathcal{B}^{\leq n} = \bigcup_{i \leq n} \mathcal{B}^i$, and $\mathcal{B}^{\geq n, \leq m)} = \bigcup_{n \leq i \leq m} \mathcal{B}^i$. Given $S \subseteq \mathcal{B}$, we write $S_\perp$ for the set $S \cup \{\perp\}$. The symbol $\perp$ represents an undefinedness/error value.

A *model* is a probability space $(\mathbf{\Omega}, \mathcal{F}, \mu)$, where:

- $\mathbf{\Omega}$ is the set of all functions $\omega \colon T_\Sigma(\mathcal{N}) \to \mathcal{B}_\perp$;
- $\mathcal{F} \subseteq \mathcal{P}(\mathbf{\Omega})$ is the σ-algebra of sets generated by

$$\{\{\omega \in \mathbf{\Omega} \mid \omega(t) = b\} \mid t \in T_\Sigma(\mathcal{N}), b \in \mathcal{B}_\perp\};$$

- $\mu \colon \mathcal{F} \to [0,1]$ is a probability measure.

As the sample space $\mathbf{\Omega}$ and the set of events $\mathcal{F}$ are determined by the signature under consideration, we will often identify a model with its probability measure μ.

If $t \in T_\Sigma(\mathcal{N})$, we represent by $\boldsymbol{t} \colon \mathbf{\Omega} \to \mathcal{B}_\perp$ the random variable on $\mathbf{\Omega}$ defined by $\boldsymbol{t}(\omega) = \omega(t)$.

We adopt standard (abuses of) notation in probability theory. if $C(b_1, \ldots, b_n)$ is some condition who satisfaction depends on the bitstring values $b_1, \ldots, b_n$, we will write

$$P_\mu[C(\boldsymbol{t_1}, \ldots, \boldsymbol{t_n})]$$

to mean

$$\mu(\{\omega \in \mathbf{\Omega} \mid C(\boldsymbol{t_1}(\omega), \ldots, \boldsymbol{t_n}(\omega))\}),$$

provided that

$$\{\omega \in \mathbf{\Omega} \mid C(\boldsymbol{t_1}(\omega), \ldots, \boldsymbol{t_n}(\omega))\} \in \mathcal{F}.$$

If $\Omega \in \mathcal{F}$, we will also write $P_\mu[\Omega_C]$ instead of $\mu(\Omega_C)$. We use standard notation for conditional probability: namely, if $P_\mu[B] > 0$, then we define $P_\mu[A \mid B] = P_\mu[A, B]/P_\mu[B]$. For instance, if $t, t' \in T_\Sigma(\mathcal{N})$ and $\Omega \in \mathcal{F}$, we have

$$P_\mu[\boldsymbol{t} = \boldsymbol{t'} \mid \Omega] = frac\mu(\{\omega \in \Omega \mid \omega(t) = \omega(t')\})\mu(\Omega).$$

...[4]

Note that this general notion of model does not require that terms are interpreted homomorphically; that is, if $t = f(t_1, \ldots, t_n)$, and $t' = f(t'_1, \ldots, t'_n)$ it is not necessarily the case that

$$P_\mu[\boldsymbol{t} \neq \boldsymbol{t'}, \boldsymbol{t_1} = \boldsymbol{t'_1}, \ldots, \boldsymbol{t_n} = \boldsymbol{t'_n}] = 0.$$

This ought to be the case, though, for models representing any actual implementation of the cryptographic primitives. Our analysis will assume that such a *real world* model is fixed. In particular, we assume that we are given a probabilistic algorithm $\widehat{[\![T]\!]}$ for each $T \in \mathcal{NT}$, and an *implementation* $\widehat{[\![f]\!]} \colon \mathcal{B}^n \to \mathcal{B}_\perp$ of each function symbol $f \in \Sigma_n$. We assume that the inputs to the algorithm $\widehat{[\![f]\!]}$ are outside the domain of the function computed by $\widehat{[\![f]\!]}$ precisely when $\widehat{[\![f]\!]}$ outputs $\perp$. Together, these two ingredients canonically determine an homomorphic model, as described below.

An *assignment* $\rho \colon \mathcal{N} \to \mathcal{B}$ is a map from names to bitstrings. If ρ is an assignment, we define $\widehat{[\![\cdot]\!]}^\rho \colon T_\Sigma(\mathcal{N}) \to \mathcal{B}_\perp$ inductively by:

$$\widehat{[\![t]\!]}^\rho = \begin{cases} \rho(t) & \text{if } t \in \mathcal{N} \\ \widehat{[\![f]\!]}(\widehat{[\![t_1]\!]}^\rho, \ldots, \widehat{[\![t_n]\!]}^\rho) & \text{if } t = f(t_1, \ldots, t_n) \text{ and} \\ & \widehat{[\![t_i]\!]}^\rho \neq \perp \text{ for } 1 \leq i \leq n \\ \perp & \text{otherwise} \end{cases}.$$

Writing $b \leftarrow A$ to denote that the bitstring b is randomly sampled from the probabilistic algorithm A, the *real world* model is now easily defined to correspond to the unique probability measure $\widehat{\mu}$ such that, whenever $b_i \in \mathcal{B}_\perp$ and $a_i \in \mathcal{N}$ are distinct names for all $i \in \{1, \ldots, n\}$, we have

$$P_{\widehat{\mu}}[\boldsymbol{a_1} = b_1, \ldots, \boldsymbol{a_n} = b_n] = \prod_{i=1}^n P[b_i \leftarrow \widehat{[\![type(a_i)]\!]}]$$

[4] **(FLAG: Please check this!!)**

and

$$P_{\widehat{\mu}}\left[\boldsymbol{f(t_1,\ldots,t_n)} \neq \widehat{[\![f]\!]}(\boldsymbol{t_1},\ldots,\boldsymbol{t_n})\right] = 0$$

for all terms $f \in \Sigma_n$ and $t_1, \ldots, t_n \in T_\Sigma(\mathcal{N})$.

We say that an assignment ρ is *samplable* (from $[\![\cdot]\!]$) if, for each name $a \in \mathcal{N}$, we have $P[\rho(a) \leftarrow \widehat{[\![type(a)]\!]}] > 0$. If we set

$$\widehat{\boldsymbol{\Omega}} = \left\{ \widehat{[\![\cdot]\!]}^{\rho} \mid \rho \text{ is samplable} \right\},$$

we have that $\widehat{\mu}(\widehat{\boldsymbol{\Omega}}) = 1$. Equivalently, the set $\boldsymbol{\Omega} \setminus \widehat{\boldsymbol{\Omega}}$ of all functions $\omega \colon T_\Sigma(\mathcal{N}) \to \mathcal{B}_\perp$ that are non-homomorphic or homomorphic over non-samplable assignments (under the type interpretation function $[\![\cdot]\!]$) has zero $\widehat{\mu}$-measure.

Example II.2. Example II.1 introduces the set of name types $\mathcal{NT} = \{\mathsf{pw}, \mathsf{sym\_key}, \mathsf{text}\}$, and the signature $\Sigma^{\mathcal{DY}} = \left\{ \mathsf{h}, \pi_1, \pi_2, \{\!|\cdot|\!\}_\cdot, \{\!|\cdot|\!\}_\cdot^{-1}, \langle\cdot,\cdot\rangle \right\}$.

[5] As the basis for a concrete real world model, we shall consider the set up of a common block cipher. Thus, we may take $\widehat{[\![\mathsf{pw}]\!]}$ to be a random generator of *weak passwords* easy for a human to memorize (e.g., short passwords, or passwords based on dictionary words), $\widehat{[\![\mathsf{sym\_key}]\!]}$ a random generator of symmetric keys (e.g., with 256 bits), and $\widehat{[\![\mathsf{text}]\!]}$ a random generator of text messages (namely, of block size 256). Furthermore, we may take $\widehat{[\![h]\!]}$ to be some concrete 256-bit hashing algorithm, $\widehat{[\![\langle\cdot,\cdot\rangle]\!]}, \widehat{[\![\pi_1]\!]}, \widehat{[\![\pi_2]\!]}$ to provide concrete block concatenation and projection algorithms, and take $\widehat{[\![\{\!|\cdot|\!\}_\cdot]\!]}$ to implement the envisaged block cipher algorithm, with deciphering given by $\widehat{[\![\{\!|\cdot|\!\}_\cdot^{-1}]\!]}$.

In this setting, suppose that $\widehat{[\![\{\!|\cdot|\!\}_\cdot^{-1}]\!]}$ is used to decrypt a bitstring b with key k. If b's length is not a multiple of k's length, this decryption is not a valid operation and the corresponding random variable should evaluate $\widehat{[\![\{\!|b|\!\}_k^{-1}]\!]}$ to $\perp$.

The probability distribution induced by the example above accurately represents the generation of messages using real-world random generation algorithms and functions. Unfortunately, as one might expect, computing probabilities in such a model is impractical — in this case, studying a security property amounts to computing the exact probability of a successful attack to the property given the limited resources of the attacker.

In general, there is a tradeoff between accuracy and feasibility when choosing a probability distribution for our model. However, simpler probability distributions may not take into account all the knowledge that an attacker can possibly obtain by observing messages and reasoning about the properties of the cryptographic primitives used, ultimately reducing our framework to modeling a standard Dolev-Yao attacker dealing with perfect encryption.

[5] check, please

Much of this paper will be devoted to describing how to express relevant properties of the functions used and using them to compute an approximate, yet meaningful and practical, probability distribution. Put another way, we seek to obtain a feasible way of estimating the real world probabilities by carefully framing the resources that a potential attacker may use. In Section V, we propose a simple probability distribution that is easy to define and compute. We show in Section VI that this probability distribution is rich enough to provide meaningful estimates of the probability of successful off-line guessing attacks.

III. Modeling Functions

A. Equational theories.

In our approach, the attacker can take advantage of equational properties of the real model, as is standard in symbolic models [2]. Recall that an *equational theory* $\approx$ is a congruence relation on $T_\Sigma(\mathcal{N})$; that is, $\approx$ is a reflexive, symmetric and transitive binary relation on $T_\Sigma(\mathcal{N})$, closed under the application of function symbols. As usual, we write $t \approx t'$ instead of $(t, t') \in \approx$.

Let $\omega \colon T_\Sigma(\mathcal{N}) \to \mathcal{B}_\perp$. We say that ω *(weakly) satisfies* $\approx$, and write $\omega \models \approx$, if whenever $t \approx t'$ then either $\omega(t) = \omega(t')$, or $\omega(t) = \perp$, or $\omega(t') = \perp$. We say that a model μ *satisfies* $\approx$, and write $\mu \models \approx$, if $\mu(\{\omega \mid \omega \not\models \approx\}) = 0$.

A rewrite rule is a pair (l, r), written as $l \to r$, where $l, r \in T_\Sigma(\mathcal{V})$. A rewriting system R over Σ is a finite set of rewrite rules. Given a rewriting system R, the relation $\to_R \subseteq T_\Sigma(\mathcal{N}) \times T_\Sigma(\mathcal{N})$ is, as usual, as the smallest relation such that, if $(l \to r) \in R$ and $\sigma \colon vars(l) \to T_\Sigma(\mathcal{N})$, then $l\sigma \to_R r\sigma$, and if $t_1, \ldots, t_n, t_i' \in T_\Sigma(\mathcal{N})$, $t_i \to_R t_i'$, and $f \in \Sigma_n$, then $f(t_1, \ldots, t_i, \ldots, t_n) \to_R f(t_1, \ldots, t_i', \ldots, t_n)$.

If the rewriting system $\to_R$ is convergent[6], then each term t has a unique normal form $t{\downarrow_R} \in T_\Sigma(\mathcal{N})$. In this case, we define $\approx_R \subseteq T_\Sigma(\mathcal{N}) \times T_\Sigma(\mathcal{N})$ by

$$\approx_R = \{(t, t') \mid t{\downarrow_R} = t'{\downarrow_R}\}.$$

We will assume given a convergent rewriting system R whose equational theory $\approx_R$ is satisfied by the real world model $\widehat{\mu}$, or equivalently, that if $\widehat{[\![t]\!]}^{\rho} \neq \perp$ and $\widehat{[\![t']\!]}^{\rho} \neq \perp$ then $\widehat{[\![t]\!]}^{\rho} = \widehat{[\![t']\!]}^{\rho}$, for every samplable assignment ρ.

Example III.1. The capabilities of a Dolev-Yao intruder (without asymmetric encryption) can be represented by a rewriting system $R_{\mathcal{DY}}$ over $\Sigma^{\mathcal{DY}}$, given by

$$R_{\mathcal{DY}} = \{\pi_1(\langle x, y\rangle) \to x, \pi_2(\langle x, y\rangle) \to y, \left\{\!\left|\{\!|x|\!\}_y\right|\!\right\}_y^{-1} \to x \} .$$

It is simple to check that this rewriting system is convergent. Furthermore, it is expectedly satisfied by any reasonable real world model.

[6] **(FLAG: cite?)**

B. Describing name generation and cryptographic functions.

To represent random generation of data, we will consider a *type interpretation* function $[\![.]\!]$ that associate each name type $T \in \mathcal{NT}$ to a set $[\![T]\!] \subseteq \mathcal{B}$ of bitstrings (distinct from $\bot$). Intuitively, each name a of type T represents the random generation of a bitstring in $[\![T]\!]$, independent from any other random generation of data.

Let $\omega \colon T_\Sigma(\mathcal{N}) \to \mathcal{B}_\bot$. We say that ω *satisfies* $[\![.]\!]$, and write $\omega \models [\![.]\!]$, if $\omega(a) \in [\![type(a)]\!]$ for every $a \in \mathcal{N}$. We say that a model μ *satisfies* $[\![.]\!]$, and write $\mu \models [\![\cdot]\!]$, if $\mu(\{\omega \mid \omega \not\models [\![\cdot]\!]\}) = 0$.

As before, we will assume that the type interpretation function considered is satisfied by the real world model $\widehat{\mu}$. This amounts to the reasonable assumption that the attacker knows the targets of the real world sampling algorithms for the corresponding types.

Example III.2. In our running example, we now make concrete assumptions about the real world model previously described and assign a set $[\![T]\!]$ to each type $T \in \mathcal{NT}$ as follows:

- $[\![\mathsf{pw}]\!]$ is a subset of $\mathcal{B}^{256}$ with cardinality $|[\![\mathsf{pw}]\!]| = 2^{24}$,
- $[\![\mathsf{sym\_key}]\!] = \mathcal{B}^{256}$;
- $[\![\mathsf{text}]\!] = \bigcup_{n \in \mathbb{N}} \mathcal{B}^{256n}$.

To express an attacker's knowledge of the real world function implementations, we must consider additional subsets of bitstrings. These allow us to model the attacker's knowledge about the domain of definition of each of the algorithms, as well as about the range of their outputs when given inputs in specific subsets.

To represent such properties, we use a set $\mathcal{PT}$ of *property types*, not necessarily disjoint from $\mathcal{NT}$ along with the set $\overline{\mathcal{PT}} = \{\overline{T} \mid T \in \mathcal{PT}\}$ of *complement property types*. We extend our interpretation function $[\![\cdot]\!]$ to $\mathcal{PT} \cup \overline{\mathcal{PT}}$ so that, for each $T \in \mathcal{PT}$, $[\![T]\!]$ is a finite set of bitstrings and, for each $\overline{T} \in \overline{\mathcal{PT}}$, $[\![\overline{T}]\!] = \overline{[\![T]\!]}$.

A *property statement* is a tuple $(f, T_1, \ldots, T_n, T_r)$, written $f[T_1, \ldots, T_n] \subseteq T_r$, where $f \in \Sigma_n$ is a function symbol and $T_1, \ldots, T_n \in \mathcal{PT} \cup \overline{\mathcal{PT}}$ and $T_r \in \mathcal{PT}$ are property types. Property statements express properties of the functions represented by the symbols in Σ.

If $ps = (f[T_1, \ldots, T_n] \subseteq T_r)$, we define:

- the *head symbol* of ps is $head(ps) = f$;
- the *range type* of ps is $ran(ps) = T_r$;
- if $[\![\cdot]\!]$ is a domain interpretation function, then the $[\![\cdot]\!]$-*domain* of ps is

$$[\![dom]\!](ps) = [\![T_1]\!] \times \ldots \times [\![T_n]\!].$$

If PS is a set of property statements, we will denote by PS_f the set of property statements $ps \in PS$ whose head symbol is f, that is, $head(ps) = f$.

Let $\omega \colon T_\Sigma(\mathcal{N}) \to \mathcal{B}_\bot$ and ps be a probabilistic statement. We say that ω *satisfies* ps, and write $\omega \models ps$, if whenever $(\omega(t_1), \ldots, \omega(t_n)) \in [\![dom]\!](ps)$ then $\omega(f(t_1, \ldots, t_n)) \in [\![ran(ps)]\!]$. If PS is a set of property statements, we say that ω *satisfies* PS, and write $\omega \models PS$, if μ satisfies every property statement in PS. Further, we say that a model μ *satisfies* PS, and write $\mu \models PS$, if $\mu(\{\omega \mid \omega \not\models PS\}) = 0$.

Two property statements ps and ps' are said to be *disjoint* if either $head(ps) \neq head(ps')$ or $dom(ps) \cap dom(ps') = \emptyset$. We assume fixed a set PS of pairwise disjoint property statements sufficient for determining the domain of each $f \in \Sigma_n$, that is, for each $f \in \Sigma_n$, we can partition the set PS_f into two sets $PS_f = PS_d \uplus PS_{\overline{d}}$ such that:

- if $ps \in PS_{\overline{d}}$, then $[\![ran(ps)]\!] = \{\bot\}$;
- if $ps \in PS_d$, then $\bot \notin [\![ran(ps)]\!]$ and $dom(ps) \subseteq \mathcal{B}^n$;
- $\mathcal{B}^n_\bot = \biguplus_{ps \in PS_f} dom(PS)$.

Intuitively, if $ps \in PS_d$, then $dom(PS)$ is a subset of the domain of the function represented by f. The condition that $dom(PS) \subseteq \mathcal{B}^n$ reflects our requirement that the application of functions to $\bot$ yields $\bot$. If $ps \in PS_{\overline{d}}$, then $dom(PS)$ does not intersect the domain of f. Together, PS_d and $PS_{\overline{d}}$ uniquely determine the domain of f (for any suitable interpretation function $[\![\cdot]\!]$).

As before, we will assume that the set PS of property statements to be considered is satisfied by the real world model $\widehat{\mu}$. Intuitively, for each property statement $f[T_1, \ldots, T_n] \subseteq T_r$, this means that applying the algorithm $\widetilde{[\![f]\!]}$ to inputs in $[\![T_1]\!] \times \ldots \times [\![T_n]\!]$ yields a bitstring in $[\![T_r]\!]$.

Support: Given a term $t \in T_\Sigma(\mathcal{N})$, we will be interested in considering the *support* $supp_\mu(\boldsymbol{t})$ of the random variable $\boldsymbol{t}$, defined by

$$supp_\mu(\boldsymbol{t}) = \{b \in \mathcal{B}_\bot \mid P_\mu[\boldsymbol{t} = b] > 0\}.$$

The following lemma states that this support is finite for any term t and any probability measure μ satisfying PS.

Lemma III.3. *Let $t \in T_\Sigma(\mathcal{N})$. There exists a finite set $[\![supp_{PS}]\!](\boldsymbol{t})$ such that, for any probability measure $\mu \vDash PS, [\![\cdot]\!]$ satisfying PS and $[\![\cdot]\!]$, we have*

$$supp_\mu(\boldsymbol{t}) \subseteq [\![supp_{PS}]\!](\boldsymbol{t}).$$

If $t = f(t_1, \ldots, t_n)$ is a term, we define the set

$$PS_t = \{(f[T_1 \ldots, T_n] \subseteq T_r) \in PS \mid T_i \cap [\![supp_{PS}]\!](\mathbf{t_i}) \neq \emptyset \text{ for all } i \in \{1, \ldots, n\}\}.$$

Lemma III.3 ensures that PS_t is always finite.

Compatibility condition: In order to ensure the compatibility between the property statements in PS and the equational theory $\approx_R$, we want to relate the property statements relevant for each term t with those that relate to its normal form $t{\downarrow}$. To formally define our compatibility condition, we will first need the notion of *selection function* for a set of terms.

Definition III.4. Let $K \subseteq T_\Sigma(\mathcal{N})$ be a set of terms. A *selection function for* K is a function $\iota\colon K \to PS$ such that, for each $t \in K \setminus \mathcal{N}$, $\iota(t) \in PS_t$. We denote by $I(K)$ the set of selection functions for K.

If ι is a selection function for K, we say that $\omega \in \mathbf{\Omega}$ is a model of ι, and write $\omega \models \iota$, if for all $t = f(t_1, \ldots, t_n)$, we have $(\omega(t_1), \ldots, \omega(t_n)) \in [\![dom]\!](\iota(t))$ and $\omega(t) \in [\![ran(\iota(t))]\!]$.

Note that, by Lemma III.3, if K is finite, then so is $I(K)$.

We say that $[\![\cdot]\!]$ and $\approx_R$ satisfy our *compatibility condition* if whenever $(l \to r) \in R$ is a rewrite rule, $\sigma\colon vars(l) \to T_\Sigma(\mathcal{N})$ is a substitution, $K = sub(l\sigma) \cup sub(r\sigma)$, and ι is a selection function for K, then either $[\![ran(\iota(r\sigma))]\!] \subseteq [\![ran(\iota(l\sigma))]\!]$, or $ran(\iota(l\sigma)) = \mathcal{B}^\emptyset_\perp$, or there is no $\omega \in \omega$ such that $\omega \models \iota$. Note that, if $ran(\iota(l\sigma)) \neq \mathcal{B}^\emptyset_\perp$, our assumptions on the property statements imply that $\perp \notin [\![ran(\iota(l\sigma))]\!]$, and thus $\perp \notin [\![ran(\iota(r\sigma))]\!]$, so that $\omega(l\sigma) \neq \perp \Rightarrow \omega(r\sigma) \neq \perp$. In order words, if $l\sigma$ is defined, then so is $r\sigma$.

We restrict ourselves to considering domain interpretation function $[\![\cdot]\!]$ and equational theories $\approx_R$ that satisfy our compatibility condition.

Example III.5. We continue our running example by presenting a set of property types $\mathcal{PT}$ and a set of property statements PS, which we use to describe properties of the primitives in $\Sigma^{\mathcal{DY}}$.

We let $\mathcal{PT} = \{B^\emptyset, B^\emptyset_\perp\} \cup \left(\bigcup_{i\in\mathbb{N}} \left\{B^i, B^{\leq i}, B^i_\perp, B^{\leq i}_\perp\right\}\right)$, and we extend the function $[\![\cdot]\!]$ to $\mathcal{PT}$ by letting $[\![B^v_u]\!] = \mathcal{B}^v_u$. Recall that $[\![\cdot]\!]$ is extended to $\overline{\mathcal{PT}}$ by $[\![\overline{T}]\!] = \overline{[\![T]\!]}$.

For readability, we will write: $\langle S_1, S_2\rangle$ for $\langle\cdot,\cdot\rangle[S_1, S_2]$; $\{\!|S_1|\!\}_{S_2}$ for $\{\!|\cdot|\!\}_\cdot[S_1, S_2]$; and $\{\!|S_1|\!\}^{-1}_{S_2}$ for $\{\!|\cdot|\!\}^{-1}_\cdot[S_1, S_2]$.

$$
\begin{aligned}
PS \quad = \{ & \\
& \textstyle\bigcup_{n\in\mathbb{N}} \left\{h[B^i] \subseteq B^{256} \mid 256n < i \leq 256(n+1)\right\} \\
& \cup \left\{\langle B^n, B^m\rangle \subseteq B^{n+m+\lceil\log(n+m)\rceil} \mid n, m > 0\right\} \\
& \cup \left\{\pi_1[B^n] \subseteq B^{\leq n-1} \mid n \geq 2\right\} \\
& \cup \left\{\pi_2[B^n] \subseteq B^{\leq n-1} \mid n \geq 2\right\} \\
& \cup \left\{\left\{\left|B^{\geq 256n+1, \leq 256(n+1)}\right|\right\}_{B^{256}} \subseteq B^{256(n+1)} \mid n \in \mathbb{N}\right\} \\
& \cup \left\{\left\{\left|\overline{B^\emptyset}\right|\right\}_{\overline{B^{256}}} \subseteq B^\emptyset_\perp\right\} \\
& \cup \left\{\left\{\left|B^{256(n+1)}\right|\right\}^{-1}_{B^{256}} \subseteq B^{\geq 256n+1, \leq 256(n+1)} \mid n \in \mathbb{N}\right\} \\
& \cup \left\{\left\{\left|B^{\geq 256n+1, \leq 256n+255}\right|\right\}^{-1}_{B^\emptyset} \subseteq B^\emptyset_\perp\right\} \\
& \cup \left\{\left\{\left|\overline{B^\emptyset}\right|\right\}^{-1}_{\overline{B^{256}}} \subseteq B^\emptyset_\perp\right\} \\
\}&
\end{aligned}
$$

This set of probabilistic statements models a hash function h that can hash any bitstring into a bitstring of length 256, a pairing function that accepts any pair of bitstrings and returns its labeled concatenation, and a symmetric encryption scheme that uses a block cipher together with some reversible padding technique.

Example III.6. We use our framework to formalize RSA encryption taking into account properties of the key generation algorithm. To generate an RSA public key one first generates two large primes p and q, typically of around 512 bits. The public key is a pair (n, e), where $n = p \cdot q$ and e (the exponent) is relatively coprime to $\varphi(n) = (p-1)(q-1)$. The private key d is the multiplicative inverse of e modulo $\varphi(n)$, i.e., the only $d \in \{1, \ldots, \varphi(n)\}$ such that $(d \cdot e) \bmod \varphi(n) = 1$.

To model RSA encryption we add the name type random to the set $\mathcal{NT}$ that we have been using in our running example. The name type random represents the random values used to generate an RSA public-private key pair. This involves generating two 512-bits prime numbers and the 1024-bits exponent. Thus $[\![\mathsf{random}]\!] \subseteq \mathcal{B}^{2048}$. By the prime number theorem, the probability of a 512-bit bitstring representing a prime number is approximately $1/\ln 2^{512}$. Thus, $|[\![\mathsf{random}]\!]| \approx (\frac{2^{512}}{\ln 2^{512}})^2 \cdot 2^{1024} \approx 2^{2032}$, where we approximate by 1 the probability that a random 1024-bits bitstrings is coprime with $p \cdot q$.

For our purposes it is sufficient to extend the signature $\Sigma^{\mathcal{DY}}$ by adding the following five primitives:

- $\mathsf{mod} \in \Sigma^{\mathcal{DY}}_1$, representing the extraction from a name of type random of the modulo $n = p \cdot q$ used in an RSA public key;
- $\mathsf{expn}, \mathsf{inv} \in \Sigma^{\mathcal{DY}}_1$, representing the extraction from a name of type random of, respectively, the public exponent and its inverse used for RSA encryption and decryption.
- $\{\cdot\}_\cdot, \{\cdot\}^{-1}_\cdot \in \Sigma^{\mathcal{DY}}_2$, representing, respectively, encryption and decryption using RSA.

To model properties of these functions we use equations and probabilistic statements. The only property of RSA asymmetric encryption that will interest us can be expressed in our framework by adding to the rewriting system $R_{\mathcal{DY}}$ the rewrite rule

$$\left\{\{M\}_{\langle \mathsf{mod}(k), \mathsf{expn}(k)\rangle}\right\}^{-1}_{\mathsf{inv}(k)} \to M,$$

where $M, k \in \mathcal{V}$ are variables.

For describing probabilistic statements that can model the relevant properties of these functions, we need to add to $\mathcal{PT}$ the following types:

- random, already in $\mathcal{NT}$, representing the random data used for generating an RSA key pair;
- $\mathsf{prodprime}$, representing the set of bitstrings that correspond to the product of two 512-bits prime numbers;
- odd, representing the set of bitstrings whose rightmost bit is 1.

We thus have $[\![\mathsf{prodprime}]\!] \subseteq \mathcal{B}^{1024}$, with $|[\![\mathsf{prodprime}]\!]| \approx 2^{1008}$, and $[\![\mathsf{odd}]\!] \subseteq \mathcal{B}^{1024}$, with $|[\![\mathsf{odd}]\!]| = 2^{1023}$. Our new probabilistic statements are then simply

$$\mathsf{mod}[\mathsf{random}] \subseteq \mathsf{prodprime}, \quad \mathsf{expn}[\mathsf{random}] \subseteq \mathsf{odd}.$$

For the sake of simplicity, we merely require in this example that the public-key exponent is odd, rather than requiring it to be coprime with the modulo. Also for simplicity and brevity, we omit here the additional statements specifying the domain and range of each of these functions. These will not play a role in our case-study examples.

C. Tests.

In our approach, we allow the attacker to test the messages they see in two ways: he can check whether two messages correspond to the same bitstring, and he can test whether a message is a bitstring in a given set. To represent these tests, we introduce yet another set $\mathcal{TT}$ of *test types*, and we extend our type interpretation function $[\![\cdot]\!]$ to $\mathcal{TT}$, so that, for each $T \in \mathcal{TT}$, $[\![T]\!]$ is a set of bitstrings. Again, $\mathcal{TT}$, $\mathcal{PT}$ and $\mathcal{NT}$ need not be disjoint.

We assume that, for each test type $TT \in \mathcal{TT}$, there exists an algorithm $\widehat{[\![TT]\!]}$ that an attacker can use to test whether a given bitstring b is in $[\![TT]\!]$ or not — that is, for any $b \in \mathcal{B}_\perp$, $\widehat{[\![TT]\!]}(b)$is 1 if $b \in [\![TT]\!]$ and 0 otherwise. An attacker's capability to perform such tests may be used to mount attacks. The computational feasibility of attacks involving such tests naturally depends on the computational feasibility of the algorithm $\widehat{[\![TT]\!]}$. Therefore, it is important to choose test types that accurately model the capabilities of a realistic attacker. [7]

Example III.7. Continuing the RSA example, observe that RSA public keys consist of a modulo $n = p \cdot q$, where p and q are two large primes, and an exponent e coprime with $(p-1)(q-1)$. While it may be unfeasible to check whether a modulo n is indeed the product of two large prime factors, an attacker can nevertheless check whether the modulo has small prime factors, and at least check whether the exponent e is odd.

To describe these capabilities, we consider the set of test types $\mathcal{TT} = \{\mathsf{odd}, \mathsf{nspf}\}$, where odd is also a member of $\mathcal{PT}$ and is as described above, and nspf represents a set of 1024-bits bitstrings without prime factors smaller than 10^6. We have $|[\![\mathsf{nspf}]\!]| \approx 2^{1024}/24$.

In Section IV, we show how this redundancy of RSA public keys can be used by an attacker to mount off-line guessing attacks.

IV. Off-line guessing

Our analysis depends on an equational theory $\approx_R$, a type interpretation function $[\![\cdot]\!]$, and a set of property statements PS. Properties of the encryption scheme are represented by the equational theory $\approx_R$ and the property statements PS, adequately interpreted by the type interpretation function $[\![\cdot]\!]$. Test types in $\mathcal{TT}$ are used to represent the attacker's capabilities of checking whether bitstrings have certain properties. In this section we describe how these ingredients can be used to find and estimate the probability of success of non-trivial off-line guessing attacks.

Throughout this section we assume that R is a convergent rewriting system, $\phi = \upsilon\tilde{n}.\sigma$ is a frame, μ is a probability measure on $\mathbf{\Omega}$, and $s \in \mathcal{N}_T \cap \tilde{n}$ is a *secret* name.

Suppose that the set $[\![T]\!]$ is small enough that an attacker can feasibly enumerate all bitstrings in this set. In an *off-line guessing attack* of s, the attacker tries each possible bistring $b \in [\![T]\!]$ and tries to use his knowledge to rule out the possibility that s is b. The attacker's goal is to ultimately learn s by excluding all but one bitstring of the set $[\![T]\!]$. In the rest of this section we assume that the attacker is trying to off-line guess s.

We represent an attacker's guess by a fresh name $w \in \mathcal{N}_T \setminus names(ran(\sigma))$. We equip the attacker with two ways of verifying whether a guess w of s is correct. First, he can use his guess to construct a pair of terms (t, t') that are equal under $\approx_R$ if $w = s$, but different if $w \neq s$. This is the usual definition in symbolic methods and has been studied by using the standard notion of static equivalence [19], [20], [26]. Second, he can use his guess to construct a term t and choose some test type $T \in \mathcal{TT}$ such that t is in $[\![T]\!]$ if $w = s$, and not necessarily otherwise.

We first define the sets $eqv(\phi, s)$ and $tv(\phi, s)$ of *equational verifiers* and *type verifiers* that an attacker may use to verify his guess.

Definition IV.1. Let $w \in \mathcal{N}_T$ be a fixed fresh name (i.e., a name that does not occur in ϕ), and $x \notin dom(\sigma)$ be a fresh variable. Define $\phi_s = \upsilon\tilde{n}_x.\sigma_s$ and $\phi_w = \upsilon\tilde{n}_x.\sigma_w$, where $\tilde{n}_x = \tilde{n} \cup \{x\}$, $\sigma_s = \sigma \cup \{x \mapsto s\}$, and $\sigma_w = \sigma \cup \{x \mapsto w\}$.

The set $eqv(\phi, s)$ of *equational verifiers* of s (under ϕ) is defined by

$$\begin{aligned} eqv(\phi, s) = \{(t, t') \in T_\Sigma(\mathcal{N}) \times T_\Sigma(\mathcal{N}) \mid \\ t\sigma_s \approx_R t'\sigma_s, t\sigma_w \not\approx_R t'\sigma_w, \\ t|_p\sigma_s \approx_R t'|_p\sigma_s, p \neq \epsilon \Rightarrow t|_p = t'|_p\}. \end{aligned}$$

The set $tv(\phi, s)$ of *type verifiers* of s (under ϕ) is defined by

$$\begin{aligned} tv(\phi, s) = \{(t, TT) \in T_\Sigma(\mathcal{N}) \times \mathcal{TT} \mid \\ [\![supp_{PS}]\!](\boldsymbol{t\sigma_s}) \subseteq [\![TT]\!], [\![supp_{PS}]\!](\boldsymbol{t\sigma_w}) \not\subseteq [\![TT]\!], \\ p \neq \epsilon \Rightarrow [\![supp_{PS}]\!]((\boldsymbol{t\sigma_w})|_p) \subseteq [\![supp_{PS}]\!]((\boldsymbol{t\sigma_s})|_p)\}, \end{aligned}$$

where $[\![supp_{PS}]\!]$ is as given by Lemma III.3.[8] Given a set of equational verifiers $eqv(\phi, s)$ and a set of type verifiers $tv(\phi, s)$, we let

$$\Omega_{eqv(\phi,s)} = \{\omega \in \mathbf{\Omega} \mid (t, t') \in eqv(\phi, s) \Rightarrow \omega(t\sigma_w) = \omega(t'\sigma_w)\}$$

and

$$\Omega_{tv(\phi,s)} = \{\omega \in \mathbf{\Omega} \mid (t, TT) \in tv(\phi, s) \Rightarrow \omega(t\sigma_w) \in [\![TT]\!]\}$$

[7] **(FLAG: Please check. I wanted to avoid here words like "efficiency" or "feasibility", but that is the intuition behind our test types...)**

[8] **(FLAG: are you ok with this?!)**

be the subsets of $\mathbf{\Omega}$ in which all the equational verifiers in $eqv(\phi, s)$ (respectively, all the type verifiers in $tv(\phi, s)$) are satisfied.

Note that the sets of equational and type verifiers may be infinite.

Using μ, the expected number of guesses w that satisfy these tests is given by

$$|[\![T]\!]| \cdot P_\mu[\Omega_{eqv(\phi,s)} \cap \Omega_{tv(\phi,s)}].$$

Note that this value should be at least 1. Suppose that an attacker tests all possible guesses w and randomly chooses one that satisfies all equational verifiers and type verifiers. Our estimate of the probability success of such an off-line guessing attack is then

$$\frac{1}{|[\![T]\!]| \cdot P_\mu[\Omega_{eqv(\phi,s)} \cap \Omega_{tv(\phi,s)}]}.$$

Relating symbolic and real-world attacks: The symbolic attacks found in our model can be naturally translated into attacks in the real world. Suppose that, in our model, the attacker's knowledge is described by a frame $\phi = \nu\tilde{n}.\sigma$, and we consider off-line guessing attack of a name $s \in \mathcal{N}_T$ of type T, obtaining $eqv(\phi, s)$ and $tv(\phi, s)$ as sets of equational and type verifiers. The corresponding scenario in the real world is an attacker that learns the messages represented by the terms in $ran(\sigma)$, either by eavesdropping on a network or taking active part in the communication. The attacker then enumerates all bitstrings $w \in [\![T]\!]$. Each such bitstring constitutes a *guess*, and the attackes uses it to constructs the messages represented by the terms $eqv(\phi, s)$ and $tv(\phi, s)$. If the attacer's guess is correct, then, for each $(t, t') \in eqv(\phi, s)$, the attacker expects the messages represented by the terms t and t' to represent the same bitstring and, for each $(t, TT) \in tv(\phi, s)$, the attacker expects the message represented by the term t to correspond to a bitstring in the set $[\![TT]\!]$ — which the attacker can test by using the algorithm $\widehat{[\![TT]\!]}$. The attacker then takes the set of guesses that satisfied all properties that he expects to be satisfied by the right guess and randomly picks one. He is successful if his final pick indeed corresponds to the bitstring represented by the name s. Theorem VI.3 relates the probability of success estimated by our model with the real probability of success of the corresponding attack in the real world. [9]

We now present several examples of off-line guessing attacks. These examples illustrate that such attacks can result from implementation details that are outside the scope of traditional symbolic methods and how such details can be modeled in our framework and used to discover attacks.

[9] **(FLAG: this is new, please check...)**

A. Attack on a stored password hash.

Example IV.2. For privacy and security reasons, authentication servers commonly store password hashes instead of the user's passwords themselves. Suppose that an attacker compromises such a server and obtains the hash of a weak password. The attacker may attempt to use this hash to off-line guess the password.

Using the same name types and property statements of our previous examples, suppose that this password is represented by a name $s \in \mathcal{N}_{\mathsf{pw}}$, and the hash obtained by the attacker is represented by $h(s)$. The knowledge of such an attacker is represented by the frame

$$\phi = \nu\tilde{n}.\sigma = \nu\{s\}.\{x_1 \mapsto h(s)\}.$$

To study off-line guessing, one considers the frames obtained by extending ϕ with either the password s or an arbitrary guess w; that is,

$$\phi_s = \upsilon\{s, w\}.\{x_1 \mapsto h(s), x \mapsto s\}$$
$$\phi_w = \upsilon\{s, w\}.\{x_1 \mapsto h(s), x \mapsto w\}.$$

Here, the set of type verifiers is empty ($tv(\phi, s) = \emptyset$), and the set of equational verifiers is given by

$$eqv(\phi, s) = \{(x_1, h(x)\}.$$

Let us consider the expected number of guesses w for which this equational verifier is satisfied, that is, for which $h(w) = h(s)$. Clearly, the correct guess always satisfies this equation. Recall that $[\![\mathsf{pw}]\!] \subseteq \mathcal{B}^{256}$ and

$$(h[\mathcal{B}^{256}] \subseteq \mathcal{B}^{256}) \in PS.$$

We therefore expect that each wrong guess satisfies the equation $h(s) = h(w)$ with probability 2^{-256}. Since we assumed that $|[\![\mathsf{pw}]\!]| \approx 2^{24}$, we have $2^{24}-1$ wrong guesses to consider. We conclude that the expected number of guesses w satisfying $h(w) = h(s)$ is

$$1 + \frac{2^{24} - 1}{2^{256}},$$

and we obtain an estimated probability of success of

$$\frac{1}{1 + \frac{2^{24}-1}{2^{256}}} \approx \frac{1}{1 + \frac{1}{2^{232}}}.$$

B. Attacks on the EKE protocol using RSA.

The EKE (Encrypted Key Exchange) protocol is designed to allow two parties to exchange authenticated information using a weak symmetric key without allowing off-line guessing attacks [1]. In Alice&Bob notation, the protocol is described by

1. $\mathsf{A} \to \mathsf{B}$: $\{\!|\mathsf{pub}(c_1)|\!\}_s$
2. $\mathsf{B} \to \mathsf{A}$: $\left\{\!\left|\{k_1\}_{\mathsf{pub}(c_1)}\right|\!\right\}_s$
3. $\mathsf{A} \to \mathsf{B}$: $\{\!|m_1|\!\}_{k_1}$
4. $\mathsf{B} \to \mathsf{A}$: $\{\!|\langle m_1, m_2\rangle|\!\}_{k_1}$
5. $\mathsf{A} \to \mathsf{B}$: $\{\!|m_2|\!\}_{k_1}$

where $\{\!|M|\!\}_K$ (respectively $\{M\}_K$) represents the symmetric (respectively asymmetric encryption) of M with key (respectively public key) K and $\mathsf{pub}(M)$ represents the generation of a public key from some random data M.

EKE is known to be vulnerable to off-line guessing attacks when implemented using the RSA asymmetric encryption scheme [1]. This is due to the redundancy of RSA public keys: an RSA public key is a pair $(p \cdot q, n)$, where p, q are two large primes and n is coprime with $(p-1)(q-1)$. Thus, an attacker that sees the first message $\{\!|\mathsf{pub}(c_1)|\!\}_s$ can try to decrypt it using his multiple guesses and check whether a valid RSA public key results from each decryption.

In the next examples we show how our methods can be used to estimate the probability of a successful off-line guessing attack.

Example IV.3. Let $s \in \mathcal{N}_{\mathsf{pw}}$ be a name representing the weak password shared between A and B that the attacker tries to guess. Suppose that A generates an RSA key pair, represented as a name $r \in \mathcal{N}_{\mathsf{random}}$ of type random. A then computes from r his public key and sends it to B, encrypted with the weak password s, as prescribed by the protocol.

After observing this message in the network, the attacker's knowledge is described by the the frame

$$\phi = \upsilon\{r\}.\{x_1 \mapsto \{\!|\langle \mathsf{mod}(r), \mathsf{expn}(r)\rangle|\!\}_s\}.$$

To study off-line guessing attacks, one uses the frames ϕ_s and ϕ_w given by:

$$\phi_s = \upsilon\{k, w\}.\{x_1 \mapsto \{\!|\langle \mathsf{mod}(r), \mathsf{expn}(r)\rangle|\!\}_s, x_2 \mapsto s\},$$
$$\phi_w = \upsilon\{k, w\}.\{x_1 \mapsto \{\!|\langle \mathsf{mod}(r), \mathsf{expn}(r)\rangle|\!\}_s, x_2 \mapsto w\}.$$

For ϕ and s there are no equational verifiers: $eqv(\phi, s) = \emptyset$. However, the attacker can try to decrypt the message sent by A with his guess w for the secret and check whether it yields a valid RSA public key. One thus obtains the set of type verifiers

$$tv(\phi, s) = \left\{(\pi_1(\{\!|x_1|\!\}^{-1}_{x_2}), \mathsf{nspf}), (\pi_2(\{\!|x_1|\!\}^{-1}_{x_2}), \mathsf{odd})\right\}.$$

The attacker tries all 2^{24} possible bitstrings for the weak password s and checks, for each of them, whether all conditions in $tv(\phi, s)$ are satisfied. He then chooses a random bitstring among those that satisfy all those conditions. The right bitstring always satisfies $tv(\phi, s)$; each wrong one satisfies these conditions with probability

$$P_\mu\left[\boldsymbol{\pi_1(\{\!|x_1|\!\}^{-1}_{x_2})} \in [\![\mathsf{nspf}]\!], \boldsymbol{\pi_2(\{\!|x_1|\!\}^{-1}_{x_2})} \in [\![\mathsf{odd}]\!]\right] \approx \tfrac{1}{24} \cdot \tfrac{1}{2}.$$

We thus obtain

$$1 + \frac{2^{24} - 1}{48}$$

as an estimate for the total number of passwords satisfying the conditions in $tv(\phi, s)$, and an estimated probability of success of

$$\frac{1}{1 + \frac{2^{24}-1}{48}} \approx \frac{1}{2^{18.5}}$$

for this off-line guessing attack.

In Section V we present a simple and general way to compute a probability distribution μ from the equational theory $\approx_R$, the set of property statements PS, and the type intepretation function $[\![\cdot]\!]$ that yields precisely this estimate for the probability of success of this off-line guessing attack.

Example IV.4. Suppose that, to prevent the previous attack, only the exponent of the RSA public key is encrypted in the first message. The authors of EKE note [1] that this variant is still vulnerable to off-line guessing attacks if more sessions of the protocol are executed: Since the exponent of an RSA key is always odd, one can try to decrypt each encryption of a public key with each guess. For the right guess, decrypting each of the encryptions will always yield an odd exponent. The probability that a wrong guess achieves this decreases exponentially with the number of encryptions, and the attacker gets one such encryption per session.

To formalize this in our setting, we let $\phi = \upsilon\tilde{n}.\sigma$ be the frame representing the attacker's knowledge, where

$$\tilde{n} = \{r_1, \ldots, r_n, s\},$$
$$\sigma = \{x_i \mapsto \langle \mathsf{mod}(r_i), \{\!|\mathsf{expn}(r_i)|\!\}_s\rangle \mid i \in \{1, \ldots, n\}\}.$$

The frames ϕ_s and ϕ_w used are as expected: $\phi_s = \upsilon\tilde{n} \cup \{w\}.\sigma_s$ and $\phi_w = \upsilon\tilde{n} \cup \{w\}.\sigma_w$, and $\sigma_s = \sigma \cup \{x_{n+1} \mapsto s\}$, $\sigma_w = \sigma \cup \{x_{n+1} \mapsto w\}$.

As before, there are no equational verifiers: $eqv(\phi, s) = \emptyset$. The set of type verifiers is given by

$$tv(\phi, s) = \{(\{\!|\pi_2(x_1)|\!\}^{-1}_{x_{n+1}}, \mathsf{odd}), \ldots, (\{\!|\pi_2(x_n)|\!\}^{-1}_{x_{n+1}}, \mathsf{odd})\}.$$

By a reasoning similar to that employed in IV.3, we obtain

$$\frac{1}{P_\mu[tv(\phi, s)]} = \frac{1}{1 + \frac{2^{24}-1}{2^n}} = \frac{2^n}{2^n + 2^{24} - 1}$$

as an estimate for the probability of success of this off-line guessing attack.

Again, the probability distribution that we will present in Section V yields precisely the same estimate for the probability of success of this off-line guessing attack.

V. Probability distributions

Recall that the ingredients to our analysis are an equational theory $\approx_R$, a type interpretation function $[\![\cdot]\!]$, and a set PS of property statements. In this section we use these elements to define the probability distribution that determines our model. This probability distribution to obtain our estimates of the probability of success of attacks, as already illustrated by the examples in the previous section.

A. The probability measure μ_w.

Let

$$\Omega_{(PS,[\![\cdot]\!])} = \{\omega \in \mathbf{\Omega} \mid \omega \vDash PS, \omega \vDash [\![\cdot]\!]\}$$

be the subset of functions $\omega \in \mathbf{\Omega}$ that satisfy PS, $[\![\cdot]\!]$, and

$$\Omega_{(PS,[\![\cdot]\!],\approx_R)} = \left\{\omega \in \Omega_{(PS,[\![\cdot]\!])} \mid \omega \vDash \approx_R\right\}$$

be the subset of functions $\omega \in \mathbf{\Omega}$ that also satisfy $\approx_R$.

Let $R \subseteq \mathcal{P}(\mathbf{\Omega})$ be the set[10] of sets of the form

$$\{\omega \in \mathbf{\Omega} \mid \omega(t_i) \in B_i, i \in \{1, \ldots, n\}\}$$

for some terms $t_1, \ldots, t_n \in T_\Sigma(\mathcal{N})$ and some sets $B_1, \ldots, B_n \subseteq \mathcal{B}_\perp$. Below, we will use $r^n_{i=1}(t_i, B_i)$ to denote the set

$$r^n_{i=1}(t_i, B_i) = \{\omega \in \mathbf{\Omega} \mid \omega(t_i) \in B_i, i \in \{1, \ldots, n\}\},$$

and define $terms(r^n_{i=1}(t_i, B_i)) = \{t_i \mid 1 \leq i \leq n\}$ and $names(r^n_{i=1}(t_i, B_i)) = sub(\{t_i \mid 1 \leq i \leq n\}) \cap \mathcal{N}$. Clearly, $r^n_{i=1}(t_i, B_i) \in R$.

Definition V.1. If $r = r^n_{i=1}(t_i, B_i)$ is a set in r, $K = sub(terms(r)) \cup \{t\downarrow \mid t \in terms(r)\}$ and $\iota \in I(K)$ is a selection function for K, we define $\iota(r)$ as the set of all $\omega \in \mathbf{\Omega}$ satisfying:

- $\omega \in r$ and $\omega \models \iota(r)$;
- if $t \in K \cap \mathcal{N}_T$, then $\omega(t) \in [\![T]\!]$;
- if $i \in \{1, \ldots, n\}$ and $ran(\iota(t_i)) \neq \mathcal{B}^\emptyset_\perp$, then $\omega(t_i\downarrow) \in B_i$.

Definition V.2. Let $r \in R$, $K = sub(terms(r)) \cup \{t\downarrow \mid t \in terms(r)\}$, and $\iota \in I(K)$.

For $t \in K$, we define the *r-support* $supp_{\iota(r)}(\boldsymbol{t})$ of $\boldsymbol{t}$ by

$$supp_{\iota(r)}(\boldsymbol{t}) = \{b \in \mathcal{B}_\perp \mid \omega(t) = b \text{ for some } \omega \in \iota(r)\}.$$

If $t \in K \cap \in \mathcal{N}_T$, we define the $\iota(r)$, *PS-support* of the random variable $\boldsymbol{a}$ by

$$[\![supp_{\iota(r),PS}]\!](\boldsymbol{a}) = [\![T]\!].$$

If $t = f(t_1, \ldots, t_n) \in K$, we define

$$[\![supp_{\iota(r),PS}]\!](\boldsymbol{f(t_1, \ldots, t_n)}) = [\![ran(\iota(t))]\!].$$

Lemma V.3. *Let $r = r^n_{i=1}(t_i, B_i)$ and ι be a selection function for $K = sub(terms(r)) \cup \{t\downarrow \mid t \in terms(r)\}$.*

We have that

$$\iota(r) \in R,$$

and

$$\left(r \cap \Omega_{(PS,[\![\cdot]\!],\approx_R)}\right) = \left(\left(\biguplus_{\iota \in I(K)} \iota(r)\right) \cap \Omega_{(PS,[\![\cdot]\!],\approx_R)}\right).$$

[10]**(FLAG: it is very important to change letter!!)**

We finally define $\mu_w \colon R \to [0,1]$ for sets $r = r^n_{i=1}(t_i, B_i) \in R$ by

$$\mu_w(r) = \sum_{\iota \in I(K(r))} \prod_{t \in K(r)} \frac{\left|[\![supp_{\iota(r)}]\!](\boldsymbol{t})\right|}{\left|[\![supp_{\iota(r),PS}]\!](\boldsymbol{t})\right|},$$

where $K(r) = sub(terms(r)) \cup \{t\downarrow \mid t \in sub(terms(r))\}$.

Lemma V.4. *There exists a unique extension of μ_w to $\mathcal{F}$ that is a probability measure.*

We will abuse notation by using the same symbol μ_w for the function in R defined above and its unique extension to $\mathcal{F}$ that is a probability measure. Note that the definition μ_w depends only on the equational theory $\approx_R$, the set of property statement PS and the type interpretation function $[\![\cdot]\!]$. As such, we write $\mu_w(\approx_R, PS, [\![\cdot]\!])$ and for the probability measure μ_w thus defined.

Lemma V.5. *$\mu_w(\approx_R, PS, [\![\cdot]\!])$ satisfies the following properties.*

- *If $a_1, \ldots, a_n \in \mathcal{N}$ are distinct names, $T_1, \ldots, T_n$ are types, $B_1, \ldots, B_n$ are sets of bitstrings, and $a_i \in T_i$ for $i \in \{1, \ldots, n\}$, then*

$$P_{\mu_w}[\bigwedge_{i=1}^n (\boldsymbol{a_i} \in B_i)] = \prod_{i=1}^n \frac{|B_i \cap [\![T_i]\!]|}{|[\![T_i]\!]|}.$$

- $\mu_w(\Omega_{[\![\cdot]\!],PS,\approx_R}) = 1$.

Note that μ_w makes perfect sense as a probability measure for our model, but only as long as we assume that all the names in our signature are strong. In order to model weak names, we need to consider the possibility that distinct weak names may possibly take the same value.

Example V.6. [11]

B. The probability measure μ.

Let $r \in R$, and let $N_w(r) = names(r) \cap \mathcal{N}_w$ be the (finite) set of weak names occurring in r. We will consider the set $P(N_w(r))$ of partitions of $N_w(r)$. Let $p = \{p_1, \ldots, p_m\}$ be one such partition, so that $N_w(r) = \biguplus_{j=1}^m p_j$, and, for each $a \in N_w(r)$, denote by $j(a)$ the only index such that $a \in p_{j(a)}$. We define the set $\Omega_p \subseteq \mathbf{\Omega}$ by

$$\begin{aligned}\Omega_p = \{\omega \in \mathbf{\Omega} \mid\ & a \in N_w(r) \Rightarrow \omega(a) \in type(a),\\ & j(a) = j(a') \Rightarrow \omega(a) = \omega(a'),\\ & j(a) \neq j(a') \Rightarrow \omega(a) \neq \omega(a')\}.\end{aligned}$$

For each set $p_j \in p$, we choose a representative $a_j \in p_j$, and we define a substitution $\theta_p \colon N_w(r) \to N_w(r)$ given by $a\theta_p = a_{j(a)}$. Note that different choices of representantives a_j lead to sets $r\theta_p$ that are equal up to renaming of weak names. We say that such a partition θ_p is a *p-substitution (for $N_w(r)$)*.

[11]**(FLAG:)**

Finally, assuming that $r = r^n_{i=1}(t_i, B_i) \in R$, we define

$$r\theta_p = \{\omega \in \mathbf{\Omega} \mid \omega(t_i\theta_p) \in B_i \text{ for all } i \in \{1, \ldots, n\}\},$$

and

$$\begin{aligned}\Omega^*_p = \{\omega \in \mathbf{\Omega} \mid \\ j \in \{1, \ldots, m\} \Rightarrow \omega(a^j) \in \textstyle\bigcap_{a \in p_j} [\![type(a)]\!], \\ i \neq j \Rightarrow \omega(a^i) \neq \omega(a^j)\}.\end{aligned}$$

It is simple to check that $r\theta_p \in R$, and $\Omega_p, \Omega^*_p \in \mathcal{F}$.

We define $\mu(r)$ by

$$\mu(r) = \sum_{p \in P(N_w(r))} P_{\mu_w}[r\theta_p \mid \Omega^*_p] \cdot \mu_w(\Omega_p).$$

Lemma V.7. *There exists a unique extension of μ to $\mathcal{F}$ that is a probability measure.*

As above, we abuse notation by using the symbol μ also for the unique extension of μ to $\mathcal{F}$ that is a probability measure. This probability measure is our intended model. As for μ_w, μ depends only on $\approx_R$, PS and $[\![\cdot]\!]$. As such, we write $\mu(\approx_R, PS, [\![\cdot]\!])$ for the probability measures μ defined as above, and use the term *model* also for the tuple $(\approx_R, PS, [\![\cdot]\!])$.

Lemma V.8. *$\mu(\approx_R, PS, [\![\cdot]\!])$ satisfies the following properties.*

- $\mu(\Omega_{[\![\cdot]\!], PS, \approx_R}) = 1$.
- *If $a_1, \ldots, a_n \in \mathcal{N}$ are distinct names, $T_1, \ldots, T_n$ are types, $B_1, \ldots, B_n$ are sets of bitstrings, and $a_i \in T_i$ for $i \in \{1, \ldots, n\}$, then*

$$P_\mu[\bigwedge_{i=1}^{n} \boldsymbol{a_i} = b_i] = \prod_{i=1}^{n} \frac{|B_i \cap [\![T_i]\!]|}{|[\![T_i]\!]|}.$$

- *If $a_1, a_2 \in \mathcal{N}_w$ are weak names, $\theta = \{a_1 \mapsto a_2\}$, $t_1, t_2 \in T_\Sigma(\mathcal{N})$ are such that $t_1\theta \approx_R t_2\theta$, and $P_\mu[\boldsymbol{a_1} = \boldsymbol{a_2}] > 0$, then*

$$P_\mu[\boldsymbol{t_1} \neq \boldsymbol{t_2}, \boldsymbol{t_1} \neq \bot, \boldsymbol{t_2} \neq \bot \mid \boldsymbol{a_1} = \boldsymbol{a_2}] = 0.$$

Example V.9. [12]

C. Computing probabilities.

[13]

VI. Soundness

One is often interested in studying asymptotic properties of the results yielded by symbolic methods; for instance, computational soundness results guarantee that, when the cryptographic primitives being used satisfy certain security properties and the frames representing the attacker's knowledge satisfy certain "well-formedness" conditions, a proof of security in a symbolic model implies that the probability of a successful attack in the real world is negligible as a function of a security parameter.

In this section we provide a different result: namely, we prove that, under certain assumptions on the cryptographic primitives being used and the adequacy of our model (as expressed by $(PS, [\![\cdot]\!], \approx_R)$), then the attacks described by our framework correspond to attacks in the real world whose probability of success differs negligibly from our estimated probability.

In order to obtain such results, we need two countably infinite sequences of models, each indexed by a security parameter η. One is the sequence $(\widehat{\Omega}_\eta)_{\eta \in \mathbb{N}}$, representing the real world model with its name generation algorithms and cryptographic primitives. We assume that, for security parameter η, $\widehat{\Omega}_\eta = (\mathbf{\Omega}, \mathcal{F}, \widehat{\mu}_\eta)$ is the real world model obtained (as described in Section II-B) if the random generation algorithm associated to each name type $T \in \mathcal{NT}$ is $\widehat{[\![T]\!]}_\eta$ and the function associated to each function symbol $f \in \Sigma_n$ is $\widehat{[\![f]\!]}_\eta$. The second is the sequence $(\Omega_\eta)_{\eta \in \mathbb{N}}$, where $\Omega_\eta == (\mathbf{\Omega}, \mathcal{F}, \mu_\eta)$ for each $\eta \in \mathbb{N}$. As before, we identify our models with their associated probability distribution μ, and our framework uses an equational theory $\approx_R$, a domain interpretation function $[\![\cdot]\!]$, and a set of property statements PS to estimate this probability distribution. We assume that the equational theory $\approx_R$ and the set of property statements PS is the same regardless of the security parameter, but that we have a countably infinite family $([\![\cdot]\!]_\eta)_\eta$ of domain interpretation functions. Thus, we have

$$\Omega_\eta = (\mathbf{\Omega}, \mathcal{F}, \mu_\eta),$$

where

$$\mu_\eta = \mu(\approx_R, [\![\cdot]\!]_\eta, PS).$$

Recall that a function $k \colon \mathbb{N} \to \mathbb{R}$ is *negligible* if, for all $c \in \mathbb{N}$, $k \in o(n^{-c})$.

Assumptions on name generation: We assume that, for each $T \in \mathcal{N}_T$ and each $\eta \in \mathbb{N}$, $\widehat{[\![T]\!]}_\eta$ outputs a random bitstring in $[\![T]\!]_\eta$ with uniform probability distribution, that is, we assume that

$$P[b \leftarrow \widehat{[\![T]\!]}_\eta] = \begin{cases} 1/\,|[\![T]\!]_\eta| & \text{if } b \in [\![T]\!]_\eta \\ 0 & \text{otherwise} \end{cases},$$

and that different samples are independent.

We assume further that, for all strong name types $T \in \mathcal{NT}_s$, $1/\,|[\![T]\!]_\eta|$ is negligible as a function of η, and, for all weak name types $T \in \mathcal{NT}_w$, we have $|[\![T]\!]_\eta| \in \mathcal{O}(\eta^k)$ for some $k \in \mathbb{N}$.

Finally, we assume that if $W \subseteq \mathcal{NT}_w$ is a set of weak name types, then there is some $T_W \in \mathcal{NT}_w$ such that

$$\bigcap_{T \in W} [\![T]\!] = [\![T_W]\!].$$

[12] **(FLAG:)**
[13] **(FLAG:)**

Assumptions on functions: Let $t, t' \in T_\Sigma(\mathcal{N})$ be terms such that $t \not\approx_R t'$, and

$$p = \{a \mid a \in (names(t) \cup names(t')) \cap \mathcal{N}_w\}$$

be the trivial discrete partition of the weak names of t and t'. We assume that the probabilities

$$P_{\widehat{\mu}_\eta}[\mathbf{t} = \mathbf{t}', \mathbf{t} \neq \bot, \mathbf{t}' \neq \bot \mid \Omega_p]$$

and

$$P_{\mu_\eta}[\mathbf{t} = \mathbf{t}', \mathbf{t} \neq \bot, \mathbf{t}' \neq \bot \mid \Omega_p]$$

are negligible as functions of η. Intuitively, this means that if two terms are not equationally equal and distinct weak names occurring in those terms are mapped to distinct bitstrings, then the probability (under either $\widehat{\mu}$ or μ) that the two terms are equal is negligible unless one of them is mapped to $\bot$).

A. Admissible models.

Definition VI.1. Let $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ and $\phi = \nu\tilde{n}.\sigma$ be a frame, and let $a_1, \ldots, a_n \in \mathcal{N}$ be fresh names (i.e., $\{a_1, \ldots, a_n\} \cap names(ran(\sigma)) = \emptyset$.

Consider the frames

$$\phi' = \nu(\tilde{n} \cup \{a_1, \ldots, a_n\}).\sigma$$

and

$$\phi_a = \nu(\tilde{n} \cup \{a_1, \ldots, a_n\}).\sigma_a,$$

where $\sigma_a = \{x_i \mapsto a_i \mid i \in \{1, \ldots, n\}\}$.

We say that ϕ *has no non-trivial equations* (with respect to $\approx_R$) if, for all $\zeta, \zeta' \in T_{\phi'}$, we have

$$\zeta\sigma = \zeta'\sigma \qquad \text{iff} \qquad \zeta\sigma_a = \zeta'\sigma_a.$$

If the above condition is not satisfied, we say that ϕ *has non-trivial equations* (with respect to $\approx_R$).

The above condition is a static equivalence property: a frame has no non-trivial conditions if it is statically equivalent to a frame that maps all variables to fresh names.

Let $T = \{t_1, \ldots, t_n\} \in T_\Sigma(\mathcal{N})$ be a finite set of terms, and let $\{w_1, \ldots, w_m\}$ be the set of weak names occurring in $\{t_1, \ldots, t_n\}$, so that

$$names(\{t_1, \ldots, t_n\}) \cap \mathcal{N}_w = \{w_1, \ldots, w_m\}.$$

We let $\phi^*(T)$ denote the frame $\phi^*(T) = \nu\tilde{n}_T.\sigma_T$, where $\tilde{n}_T = \bigcup_{t \in T} names(t)$ and

$$\sigma_T = \{x_i \mapsto t_i \mid 1 \leq i \leq n\} \cup \{x_{n+i} \mapsto w_i \mid 1 \leq i \leq m\}.$$

Definition VI.2. Consider the following experiment for each security parameter $\eta \in \mathbb{N}$:

1. An adversary A chooses a finite set $T \subset T_\Sigma(\mathcal{N})$.
2. If $\phi^*(T) = \nu\tilde{n}_T.\sigma_T$ has non-trivial equations with respect to $\approx_R$, then $\bot$ is output to the adversary; otherwise, a bit b is selected at random.
3. If $b = 0$, then $(\mathbf{x_1}\sigma, \ldots, \mathbf{x_n}\sigma)$ is sampled from μ_η and output to the adversary.
4. If $b = 1$, then $(\mathbf{x_1}\sigma, \ldots, \mathbf{x_n}\sigma)$ is repeatedly sampled from $\widehat{\mu}_\eta$, and the first sample satisfying

$$w_i, w_j \in \mathcal{N}_w \cap ran(\sigma_T), w_i \neq w_j \Rightarrow \mathbf{w_i} \neq \mathbf{w_j}$$

is output to the adversary.
5. The adversary outputs a bit b'.

The *advantage of* A is defined by

$$Adv^A_{\approx_R, \Omega_\eta, \widehat{\Omega}_\eta} = P[b = b'] - 1/2.$$

We say that $(\approx_R, PS, (\llbracket\cdot\rrbracket_\eta)_{\eta \in \mathbb{N}})$ is an *admissible model* of $(\widehat{\Omega}_\eta)_{\eta \in \mathbb{N}}$ if the function $Adv^A_{\approx_R, \Omega_{(\cdot)}, \widehat{\Omega}_{(\cdot)}}$ is negligible (in η) for any probabilistic polynomial-time adversary A.

B. Soundness theorem.

Theorem VI.3. *Suppose that:*

(1) $\approx$ *is generated by a subterm convergent rewriting system* R *such that, for each rewrite rule* $(l \to r) \in R$, *if* $l = f(l_1, \ldots, l_n)$, *then* f *does not occur in any other rewrite rule or any strict subterm of* l *— that is,*

$$(l' \to r') \in R \setminus \{l \to r\} \Rightarrow l' \in T_{\Sigma \setminus \{f\}}(\mathcal{N})$$

and

$$l_1, \ldots, l_n \in T_{\Sigma \setminus \{f\}}(\mathcal{N}).$$

(2) $(\approx, PS, (\llbracket\cdot\rrbracket_\eta)_{\eta \in \mathbb{N}})$ *is an admissible model of* $(\widehat{\llbracket\cdot\rrbracket}_\eta)_{\eta \in \mathbb{N}}$.
(3) *Our assumptions on names and functions are satisfied.*
(4) $\phi = \nu\tilde{n}.\sigma$ *is such that* $names(ran(\sigma)) \cap \mathcal{N}_w = \{s\}$ *and, for all equational verifiers* $(t, t') \in eqv(\phi, s)$,

$$P_{\mu_\eta}[\boldsymbol{t} = \bot], P_{\mu_\eta}[\boldsymbol{t'} = \bot], P_{\widehat{\mu}_\eta}[\boldsymbol{t} = \bot], P_{\widehat{\mu}_\eta}[\boldsymbol{t'} = \bot]$$

are all negligible as functions of η.

Let

$$\Omega_{\phi,\eta} = \{\omega \mid \omega(t) = \omega(t') \text{ for all } (t, t') \in eqv(\phi, s), \\ \omega(t) \in \llbracket TT \rrbracket_\eta \text{ for all } (t, TT) \in tv(\phi, s)\}$$

Then,

$$\left|P_{\mu_\eta}[\Omega_{\phi,\eta} \mid \Omega_{\phi,\bot}] - P_{\widehat{\mu}_\eta}[\Omega_{\phi,\eta} \mid \Omega_{\phi,\bot}]\right|$$

is negligible (as a function of η*).*

Furthermore, the probability of success of the attack described in our model differs at most negligibly from the probability of success of the corresponding attack in the real-world as described in Section IV.

Example VI.4. Applicability to examples...

C. Discussion.

About the soundness result

- its meaning and limitations
- on real world names
  - from weak names to weak terms, and the login example
  - further on names: the uniformity assumption is essential for computing the probabilities (seems reasonable, ultimately all probabilistic algorithms rely on some good enough uniform random generator), drawback is that we cannot accurately reason about attacks that explore weaknesses of real world pseudo-random generation
- on real world functions (and names, still)
  - uniformity of real world cryptographic functions versus the weak/strong name splitting
  - why is this relevant, and where?
  - what it excludes
  - problems with pairing
- on our model, and its soundness (admissibility?)
  - soundness/admissibility of our model is closely related to the typical indistinguishability-like assumptions on cryptographic functions that underly computational reasoning and computational soundness results; no obvious way to prove this directly; but should be possible to show that, in concrete situations, our property is implied by the indistinguishability assumptions on which the actual cryptographic functions and random generators rely on – it would be awesome to include such a case explicitly, with a sketch of proof :)
- on the ingredients to our model, attacker resources
  - subterm convergency makes things simpler but should be easily generalizable, explain; it can be trivially checked; still it is not more limitative than the average result in the literature
  - the head-of-rules condition is also easily checkable, but quite restrictive, though it covers many interesting examples; why we need it; generalization?

About our approach

- how completely do we cover the capabilities of actual attackers (much better than in the rest of the automatable literature, anyway)

Whatever else I may have forgotten ...

VII. Conclusion and future work

We presented a symbolic probabilistic framework for security protocol analysis. Our framework allows one to express type properties of cryptographic primitives, in addition to standard equational properties; one can thereby model a much stronger attacker than the standard Dolev-Yao model. We illustrated the usefulness of this approach by modeling non-trivial properties of RSA encryption and using them to describe off-line guessing attacks on the EKE protocol, currently outside the scope of other symbolic methods.

Our work is also the only existing symbolic model capable of estimating the probability of success of the attacks found. Under adequate[14] assumptions about the accuracy of the properties expressed, our estimate can be shown to differ negligibly from the probability of success in the real world.

There are many interesting and practically relevant continuations of this research. In the context of off-line guessing, it should be possible to both specialize and generalize the conditions in our soundness theorem. The recent work on the computational soundness of static equivalence [17], [18], for example, introduces new notions of security that could perhaps be adapted to obtain sharper sufficient conditions for our soundness theorem. Another relevant research question that has received surprisingly little attention is the modeling of weak hash functions and the analysis of off-line guessing of non-atomic secrets. This could be useful, for instance, in scenarios like the one described in Section IV-A.

More generally, we believe that our approach can be used to analyze a broad range of attacks and weaknesses of cryptographic primitives that could not previously be analyzed by symbolic models. These include some forms of cryptanalysis (such as differential cryptanalysis to AES, DES or hash functions, as illustrated in [16]) and side-channel attacks [25]. Short-string authentication protocols, used for example in device pairing [27], are also interesting applications: short strings are a natural source of cryptographic weakness, and the analysis of such protocols is intrinsically probabilistic. Finally, distance-bounding protocols that rely on rapid-bit exchange, such as [28], are notoriously ill-suited for analysis with existing symbolic methods; these may well be amenable to analysis using our framework.

An important goal for our future research is the integration of this approach with a symbolic protocol model-checker capable of generating protocol execution traces. Since our probabilistic analysis can be performed automatically, as discussed in V-C, this would provide a means for fully automating our analysis.

References

[1] S. M. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-based protocols secure against dictionary attacks," in *IEEE SYMPOSIUM ON RESEARCH IN SECURITY AND PRIVACY*, 1992, pp. 72–84.

[2] V. Cortier, S. Delaune, and P. Lafourcade, "A survey of algebraic properties used in cryptographic protocols," *J. Comput. Secur.*, vol. 14, pp. 1–43, January 2006. [Online]. Available: http://portal.acm.org/citation.cfm?id=1239376.1239377

[14] **(FLAG: delete this word? Is it clear?)**

[3] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, P.-C. Heám, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron, "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications," in *Proceedings of the 17th International Conference on Computer Aided Verification (CAV'05)*, ser. LNCS, K. Etessami and S. K. Rajamani, Eds. Springer, 2005, vol. 3576. [Online]. Available: http://www.avispa-project.org/publications.html

[4] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, 1984.

[5] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *CRYPTO*, ser. Lecture Notes in Computer Science, D. R. Stinson, Ed., vol. 773. Springer, 1993, pp. 232–249.

[6] V. Cortier, S. Kremer, and B. Warinschi, "A survey of symbolic methods in computational analysis of cryptographic systems," *J. Autom. Reasoning*, vol. 46, no. 3-4, pp. 225–259, 2011.

[7] M. Abadi and P. Rogaway, "Reconciling two views of cryptography (the computational soundness of formal encryption)," *J. Cryptology*, vol. 20, no. 3, p. 395, 2007.

[8] P. Adão, G. Bana, and A. Scedrov, "Computational and information-theoretic soundness and completeness of formal encryption," in *CSFW*. IEEE Computer Society, 2005, pp. 170–184.

[9] M. Backes, B. Pfitzmann, and M. Waidner, "A composable cryptographic library with nested operations," in *ACM Conference on Computer and Communications Security*, S. Jajodia, V. Atluri, and T. Jaeger, Eds. ACM, 2003, pp. 220–230.

[10] P. Laud and R. Corin, "Sound computational interpretation of formal encryption with composed keys," in *ICISC*, ser. Lecture Notes in Computer Science, J. I. Lim and D. H. Lee, Eds., vol. 2971. Springer, 2003, pp. 55–66.

[11] V. Cortier, S. Kremer, R. Küsters, and B. Warinschi, "Computationally sound symbolic secrecy in the presence of hash functions," *IACR Cryptology ePrint Archive*, vol. 2006, p. 218, 2006.

[12] M. Backes and P. Laud, "Computationally sound secrecy proofs by mechanized flow analysis," in *ACM Conference on Computer and Communications Security*, A. Juels, R. N. Wright, and S. D. C. di Vimercati, Eds. ACM, 2006, pp. 370–379.

[13] B. Blanchet and D. Pointcheval, "Automated security proofs with sequences of games," in *CRYPTO*, ser. Lecture Notes in Computer Science, C. Dwork, Ed., vol. 4117. Springer, 2006, pp. 537–554.

[14] B. Blanchet, "A computationally sound mechanized prover for security protocols," *IEEE Trans. Dependable Sec. Comput.*, vol. 5, no. 4, pp. 193–207, 2008.

[15] G. Barthe, B. Grégoire, and S. Z. Béguelin, "Formal certification of code-based cryptographic proofs," in *POPL*, Z. Shao and B. C. Pierce, Eds. ACM, 2009, pp. 90–101.

[16] B. Montalto and C. Caleiro, "Modeling and reasoning about an attacker with cryptanalytical capabilities," *Electr. Notes Theor. Comput. Sci.*, vol. 253, no. 3, pp. 143–165, 2009.

[17] S. Kremer and L. Mazaré, "Adaptive soundness of static equivalence," in *ESORICS*, ser. Lecture Notes in Computer Science, J. Biskup and J. Lopez, Eds., vol. 4734. Springer, 2007, pp. 610–625.

[18] M. Baudet, V. Cortier, and S. Kremer, "Computationally sound implementations of equational theories against passive adversaries," *Inf. Comput.*, vol. 207, pp. 496–520, April 2009. [Online]. Available: http://portal.acm.org/citation.cfm?id=1512988.1513047

[19] M. Abadi, M. Baudet, and B. Warinschi, "Guessing attacks and the computational soundness of static equivalence," *Journal of Computer Security*, pp. 909–968, December 2010.

[20] M. Baudet, "Deciding security of protocols against off-line guessing attacks," in *Proceedings of the 12th ACM conference on Computer and communications security*, ser. CCS '05. New York, NY, USA: ACM, 2005, pp. 16–25. [Online]. Available: http://doi.acm.org/10.1145/1102120.1102125

[21] R. Corin, J. Doumen, and S. Etalle, "Analysing password protocol security against off-line dictionary attacks," *Electron. Notes Theor. Comput. Sci.*, vol. 121, pp. 47–63, February 2005. [Online]. Available: http://dx.doi.org/10.1016/j.entcs.2004.10.007

[22] Z. Li and W. Wang, "Rethinking about guessing attacks," in *ASIACCS*, B. S. N. Cheung, L. C. K. Hui, R. S. Sandhu, and D. S. Wong, Eds. ACM, 2011, pp. 316–325.

[23] S. Halevi and H. Krawczyk, "Public-key cryptography and password protocols," *ACM Trans. Inf. Syst. Secur.*, vol. 2, no. 3, pp. 230–268, 1999.

[24] M. Abadi and B. Warinschi, "Password-based encryption analyzed," in *ICALP*, ser. Lecture Notes in Computer Science, L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, Eds., vol. 3580. Springer, 2005, pp. 664–676.

[25] B. Köpf and D. A. Basin, "An information-theoretic model for adaptive side-channel attacks," in *ACM Conference on Computer and Communications Security*, P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds. ACM, 2007, pp. 286–296.

[26] M. Abadi and C. Fournet, "Mobile values, new names, and secure communication," in *Proc. of the 28th ACM Symp. on Principles of Programming Languages*, ser. POPL '01. New York, NY, USA: ACM, 2001, pp. 104–115. [Online]. Available: http://doi.acm.org/10.1145/360204.360213

[27] S. Laur and K. Nyberg, "Efficient mutual data authentication using manually authenticated strings," in *CANS*, ser. Lecture Notes in Computer Science, D. Pointcheval, Y. Mu, and K. Chen, Eds., vol. 4301. Springer, 2006, pp. 90–107.

[28] 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.

APPENDIX

Proof (of Lemma III.3): [15] The proof is by induction on t. If $t \in \mathcal{N}_T$ for some name type $T \in \mathcal{NT}$ and $\mu \vDash PS, [\![\cdot]\!]$, then we choose

$$[\![supp_{PS}]\!](\boldsymbol{t}) = [\![T]\!],$$

and the result is proved.

Our assumptions on our set of probabilistic statements PS imply that, for each $(b_1, \ldots, b_n) \in \mathcal{B}_\perp^n$, there is exactly one property statement $f[T_1, \ldots, T_n] \subseteq T_r$ such that

$$(b_1, \ldots, b_n) \in [\![T_1]\!] \times \ldots \times [\![T_n]\!].$$

Let $T_r(b_1, \ldots, b_n) \in \mathcal{PT}$ denote the range type in this probabilistic statement.

Since μ satisfies PS, we have

$$P_\mu[\boldsymbol{f(t_1, \ldots, t_n)} \in [\![T_r(b_1, \ldots, b_n)]\!] \\ \mid \boldsymbol{t_1} \in \{b_1\}, \ldots, \boldsymbol{t_n} \in \{b_n\}] = 1.$$

Observe also that

$$supp_\mu(\boldsymbol{f(t_1, \ldots, t_n)}) \subseteq \bigcup_{(b_1,\ldots,b_n)\in B} T_r(b_1, \ldots, b_n),$$

where $B = supp_\mu(\boldsymbol{t_1}) \times \ldots \times supp_\mu(\boldsymbol{t_n})$. Now, the set B is finite by the induction hypothesis, and so are the sets $T_r(b_1, \ldots, b_n)$ for each n-uple $(b_1, \ldots, b_n)$. Thus, the result follows by taking

$$[\![supp_{PS}]\!](\boldsymbol{f(t_1, \ldots, t_n)}) = \bigcup_{(b_1,\ldots,b_n)\in B} T_r(b_1, \ldots, b_n).$$

■

Lemma VII.1. *Let $r = r_{i=1}^n(t_i, B_i)$ and ι be a selection function for $K = \{sub(t\!\downarrow) \mid t \in terms(r)\}$.*

We have that

$$\iota(r) \in R,$$

and

$$\left(r \cap \Omega_{(PS,[\![\cdot]\!],\approx_R)}\right) = \left(\left(\biguplus_{\iota\in I(K)} \iota(r)\right) \cap \Omega_{(PS,[\![\cdot]\!],\approx_R)}\right).$$

Proof: To see that $\iota(r) \in R$, we note that the conditions on ω that define the set $\iota(r)$ only depend on the bitstrings $\omega(t)$ for $t \in K$. Thus, we have

$$\iota(r) = \bigcap_{t\in K} \{\omega \in \boldsymbol{\Omega} \mid \omega(t) \in B_t\}$$

for some sets $B_t \subseteq \mathcal{B}_\perp$, and the result follows from the finiteness of K.

If $\omega \in \iota(r) \cap \Omega_{PS,[\![\cdot]\!],\approx_R}$, then $\omega \in r$ by definition of $\iota(r)$, and thus $\omega \in r \cap \Omega_{PS,[\![\cdot]\!],\approx_R}$.

Now, suppose that $\omega \in r \cap \Omega_{PS,[\![\cdot]\!],\approx_R}$, and define $i_\omega \in I(K)$ such that, for each $t = f(t_1, \ldots, t_n) \in$

[15] **(FLAG: Revise this proof to be better suited for using in the soundness theorem, using ι...)**

K, $i_\omega(t)$ is the only probabilistic statement ps such that $(\omega(t_1), \ldots, \omega(t_n)) \in [\![dom]\!](\iota(t))$. We show that $\omega \in \iota_\omega(r) \cap \Omega_{PS,[\![\cdot]\!],\approx_R}$.

- We have $\omega \in r$ by hypothesis. If $t = f(t_1, \ldots, t_n)$, then $(\omega(t_1), \ldots, \omega(t_n)) \in [\![dom]\!](\iota(t))$ by definition of ι_ω, and $\omega(t) \in [\![ran(\iota(t))]\!]$ because $\omega \models PS$. Thus, $\omega \models \iota$.
- If $t \in K \cap \mathcal{N}$, then $\omega(t) \in [\![T]\!]$, since $\omega \models [\![\cdot]\!]$.
- If $i \in \{1, \ldots, n\}$ and $ran(\iota(t_i)) \neq \mathcal{B}_\perp^\emptyset$, then $\perp \notin [\![ran(\iota(t_i))]\!]$ (from our assumptions on property statements). We have two cases.
  - If $t_i \in \mathcal{N}$, then $t_i = t_i{\downarrow}$ and, since $\omega \in r$, we have $\omega(t_i) = \omega(t_i{\downarrow}) \in B_i$;
  - If $t_i = f(t_{i,1}, \ldots, t_{i,n})$, then
    $$(\omega(t_{i,1}), \ldots, \omega(t_{i,n})) \in [\![dom]\!](\iota(t_i)),$$
    and we get $\omega(t_i) \neq \perp$ because $\omega \models PS$. Since $\omega \models \approx_R$, we conclude that $\omega(t_i{\downarrow}) \in B_i$.

This concludes the proof. ∎

Lemma VII.2. *Let $r = r_{i=1}^n(t_i, B_i)$ be a set in R, and define*

$$dec(r) = \{r_{i=1}^n(t_i, \{b_i\}) \mid b_i \in B_i \cap [\![supp_{PS}]\!](\boldsymbol{t_i}) \text{ for each } i \in \{1, \ldots, n\}\}.$$

Then, we have

$$\biguplus_{r' \in dec(r)} r' \subseteq r$$

and

$$\mu_w(r) = \sum_{r' \in dec(r)} \mu_w(r').$$

We say that $dec(r)$ is the canonical decomposition *of r.*

Proof: [16] ∎

Corollary VII.3. *Let $r_1, \ldots, r_n \in R$ be disjoint sets in R such that $r = \bigcup_{i=1}^n r_i$ is a set in R. Then,*

$$\mu_w(r) = \sum_{i=1}^n \mu_w(r_i).$$

Proof: [17] ∎

Lemma VII.4. *There exists a unique extension of μ_w to $\mathcal{F}$ that is a probability measure.*

Proof: We first prove that R is a semi-ring of sets. To prove that $\emptyset \in R$, we note that $r_{i=1}^1(t_1, \emptyset) = \emptyset$ is in R for any term t_1. Now let $r = r_{i=1}^n(t_i, B_i)$ and $r' = r_{i=1}^{n'}(t'_i, B'_i)$ be two arbitrary sets in R. Let

$$terms(r) \cup terms(r') = \{t''_1, \ldots, t''_{n''}\}.$$

There are sets $B_1, \ldots, B_{n''}, B'_1, \ldots, B'_{n''}$ such that $r = r_{i=1}^{n''}(t''_i, B_i)$ and $r' = r_{i=1}^{n''}(t''_i, B'_i)$. These sets can be obtained by simply choosing B_i to be set associated to t''_i in the definition of r if $t''_i \in terms(r)$, and $B_i = \mathcal{B}_\perp$ otherwise, and doing similar choices for the sets B'_i. We then have that

$$r \cap r' = r_{i=1}^{n''}(t''_i, B_i \cap B'_i).$$

Thus, R is closed for intersections.

Now, consider the set of functions

$$\Psi = \{\psi \colon \{1, \ldots, n''\} \to \{0, 1\}\},$$

and let $\psi_0 \in \Psi$ be the function given by $\psi_0(i) = 0$ for all $i \in \{1, \ldots, n''\}$. For each function $\psi \in \Psi$, consider the set

$$r_\psi = r_{i=1}^{n''}(t_i, B_i^{\psi(i)},$$

where $B_i^b = B_i \cap B'_i$ if $b = 0$ and $B_i^b = B_i \setminus B'_i$ if $b = 1$. The following properties are easy to verify.

- $r_\psi \in R$ for all $\psi \in \Psi$.
- If $\psi_1, \psi_2 \in \Psi$ are distinct functions, then $r_{\psi_1} \cap r_{\psi_2}$.
- $$r = \biguplus_{\psi \in \Psi} r_\psi.$$
- $$r \cap r' = r_{\psi_0}.$$

We conclude that

$$r \setminus r' = \biguplus_{\psi \in \Psi \setminus \{\psi_0\}} r_\psi$$

is a finite, disjoint union of elements in R. Thus, R is a semi-ring.

It is simple to check that $\mathcal{F}$ is the σ-algebra generated by R: On one hand, all the generators of $\mathcal{F}$ are in R, so that the σ-algebra generated by R contains $\mathcal{F}$. On the other hand, any set in R is generated by a finite intersection of sets of the form

$$\{\omega \in \boldsymbol{\Omega} \mid \omega(t) \in B\}$$

for some term t and some set of bitstrings $B \subseteq \mathcal{B}_\perp$. These sets are countable unions of sets of the form

$$\{\omega \in \boldsymbol{\Omega} \mid \omega(t) = b\},$$

which by definition are the generators of $\mathcal{F}$. Thus, $\mathcal{F}$ also contains the σ-algebra generated by R.

By Caratheodory's extension theorem, to complete our proof that μ_w is a probability measure we only need to show that, whenever

$$\left\{r^j = r_{i=1}^{n^j}(t_i^j, B_i^j) \mid j \in \mathbb{N}\right\}$$

is a set of pairwise disjoint sets in R such that their (disjoint) union

$$r = \biguplus_{j \in \mathbb{N}} r_j = r_{i=1}^n(t_i, B_i)$$

[16] **(FLAG:)**
[17] **(FLAG:)**

is also in R, then

$$\mu_w(r) = \sum_{j \in \mathbb{N}} \mu_w(r_j).$$

We first note that, since

$$\mu(r^j) = \mu(r^{n^j}_{i=1}(t^j_i, B^j_i \cap [\![supp_{PS}]\!](\boldsymbol{t^j_i}))$$

and

$$\mu(r) = \mu(r^n_{i=1}(t_i, B_i \cap [\![supp_{PS}]\!](\boldsymbol{t_i})),$$

we may assume without loss of generality that all the sets B^j_i are such that

$$B^j_i \subseteq [\![supp_{PS}]\!](\boldsymbol{t^j_i}).$$

For each $t \in T_\Sigma(\mathcal{N})$, consider the topological space $[\![supp_{PS}]\!](\boldsymbol{t})$, where all the subsets are open. This space is finite and, therefore, it is trivially compact. Now, consider the topological space

$$F = \{omega \colon T_\Sigma(\mathcal{N}) \to \mathcal{B}_\perp \mid f(t) \in [\![supp_{PS}]\!](\boldsymbol{t}) \text{ for all } t\}.$$

F is the cartesian product of the topological spaces associated to each term t. The open sets in this topological space with the product topology are precisely the sets of the form

$$r^n_{i=1}(t_i, B_i)$$

where $B_i \subseteq [\![supp_{PS}]\!](\boldsymbol{t_i})$ for all $i \in \{1, \ldots, n\}$. By Tychonoff's theorem, F with the product topology is also a compact space.

Because the open sets of F form a semi-ring (by an argument entirely analogous to the one we used to prove that R is a semi-ring), we know that $F \backslash r$ is a finite union of open sets, and thus is also open. We conclude that r is closed. Because F is compact, it follows that r is also compact. Since $\{r_j \mid j \in \mathbb{N}\}$ is a open cover of r, there must be a finite sub-cover — that is, there must be indexes $j_1, \ldots, j_m$ such that

$$r = \biguplus_{k=1}^{m} r_{j_k}.$$

And because the r_j are disjoint, it follows that $r_j = \emptyset$ for all $j \notin \{j_1, \ldots, j_m\}$.

Thus, we only need to prove that

$$\mu_w(r) = \sum_{k=1}^{m} \mu_w(r_{j_k}).$$

This follows from Corollary VII.3.

■

Lemma VII.5. *$\mu_w(\approx_R, PS, [\![\cdot]\!])$ satisfies the following properties.*

(1) *If $a_1, \ldots, a_n \in \mathcal{N}$ are distinct names, $T_1, \ldots, T_n$ are types, $B_1, \ldots, B_n$ are sets of bitstrings, and $a_i \in T_i$ for $i \in \{1, \ldots, n\}$, then*

$$P_{\mu_w}[\bigwedge_{i=1}^{n} (\boldsymbol{a_i} \in B_i)] = \prod_{i=1}^{n} \frac{|B_i \cap [\![T_i]\!]|}{|[\![T_i]\!]|}.$$

(2) $\mu_w(\Omega_{[\![\cdot]\!], PS, \approx_R}) = 1$.

Proof:

(1): This is a direct consequence of the definition of μ_w. We have

$$P_{\mu_w}[\bigwedge_{i=1}^{n} (\boldsymbol{a_i} \in B_i)] = \mu_w(r^n_{i=1}(a_i, B_i)).$$

Now, if $r = r^n_{i=1}(a_i, B_i)$, we have $K(r^n_{i=1}(a_i, B_i)) = \emptyset$ and, for all $\iota \in I(K(r))$:

- $\iota = \emptyset$;
- $[\![supp_{\iota(r)}]\!](\boldsymbol{a_i}) = B_i \cap [\![T_i]\!]$ for all $i \in \{1, \ldots, n\}$;
- $[\![supp_{\iota(r), PS}]\!](\boldsymbol{a_i}) = [\![T_i]\!]$ for all $i \in \{1, \ldots, n\}$.

Thus,

$$\mu_w(r) = \sum_{\iota \in I(K(r))} \left(\prod_{t \in K(r)} \frac{\left|[\![supp_{\iota(r)}]\!](\boldsymbol{t})\right|}{\left|[\![supp_{\iota(r), PS}]\!](\boldsymbol{t})\right|} \right)$$

$$= \prod_{i=1}^{n} \frac{|B_i \cap [\![T_i]\!]|}{|[\![T_i]\!]|}.$$

(2): Let $a \in \mathcal{N}$ and $r = r^1_{i=1}(a, \mathcal{B}_\perp \setminus [\![type(a)]\!])$. By (1), we know that

$$\mu_w(r) = P_{\mu_w}[\boldsymbol{a} \in \mathcal{B}_\perp \setminus [\![T]\!]] = 0.$$

Thus,

$$\begin{array}{l} \mu_w(\{\omega \mid \omega \not\models [\![\cdot]\!]\}) \\ = \mu_w(\bigcup_{a \in \mathcal{N}} r^1_{i=1}(a, \mathcal{B}_\perp \setminus [\![type(a)]\!])) \\ = \sum_{a \in \mathcal{N}} \mu_w(r^1_{i=1}(a, \mathcal{B}_\perp \setminus [\![type(a)]\!])) \\ = 0. \end{array} \tag{1}$$

Let $ps = (f[T_1, \ldots, T_n] \subseteq T_r)$, and $t_{n+1} = f(t_1, \ldots, t_n)$. Let $r = r^{n+1}_{i=1}(t_i, B_i)$, where $B_i = [\![T_i]\!]$ for $i \in \{1, \ldots, n\}$, and $B_{n+1} = \mathcal{B}_\perp \setminus [\![T_r]\!]$.

We have $t_1, \ldots, t_{n+1} \in K(r)$. Let $\iota \in I(K(r))$ and $\iota(t_{n+1}) = (f[T'_1, \ldots, T'_n] \subseteq T'_r)$. For all $\omega \in \iota(r)$ and all $i \in \{1, \ldots, n\}$, we have $\omega(t_i) \in [\![T'_i]\!]$; since $\omega(t_i) \in [\![T_i]\!]$ for all $\omega \in r$, we also have $\omega(t_i) \in [\![T_i]\!]$ for all $\omega \in \iota(r)$. If $\iota(t_{n+1}) \neq PS$, then, by our disjointness assumption on property statements, we have

$$([\![T_1]\!] \times \ldots \times [\![T_n]\!]) \cap ([\![T'_1]\!] \times \ldots \times [\![T'_n]\!]) = \emptyset,$$

from which we conclude that $[\![T_i]\!] \cap [\![T'_i]\!] = \emptyset$ for some $i \in \{1, \ldots, n\}$. We conclude that $supp_{\iota(r)}(\boldsymbol{t}) = \emptyset$ for all ι such that $\iota(t_{n+1}) \neq ps$. Thus,

$$\prod_{t \in K(r)} \frac{\left|[\![supp_{\iota(r)}]\!](\boldsymbol{t})\right|}{\left|[\![supp_{\iota(r), PS}]\!](\boldsymbol{t})\right|} = 0,$$

since the product contains a factor equal to 0.

On the other hand, if $\iota(t_{n+1}) = ps$, then, by a similar reasoning, we have, for all $\omega \in \iota(r)$, that $\omega(t_{n+1}) \in [\![T_r]\!]$ and $\omega(t_{n+1}) \notin [\![T_r]\!]$, and we conclude that

$$\prod_{t \in K(r)} \frac{\left|[\![supp_{\iota(r)}]\!](\boldsymbol{t})\right|}{\left|[\![supp_{\iota(r),PS}]\!](\boldsymbol{t})\right|} = 0$$

also for all ι such that $\iota(t_{n+1}) = ps$.

We conclude that

$$\mu_w(r) = \sum_{\iota \in I(K(r))} \left(\prod_{t \in K(r)} \frac{\left|[\![supp_{\iota(r)}]\!](\boldsymbol{t})\right|}{\left|[\![supp_{\iota(r),PS}]\!](\boldsymbol{t})\right|} \right) = 0.$$

Now, if $\omega \not\models PS$, we must have

$$\omega(t_1) \in [\![T_1]\!], \ldots, \omega(t_n) \in [\![T_n]\!], \omega(f(t_1, \ldots, t_n)) \notin [\![T_r]\!]$$

for some property statement $(f[T_1, \ldots, T_n] \subseteq T_r) \in PS$. For each $f(t_1, \ldots, t_n) \in T_\Sigma(\mathcal{N})$ and each $ps = (f[T_1, \ldots, T_n] \subseteq T_r) \in PS_f$, define

$$r(f(t_1, \ldots, t_n), ps) = r^{n+1}_{i=1}(t_i, B_i),$$

where again $t_{n+1} = f(t_1, \ldots, t_n)$, $B_i = [\![T_i]\!]$ for $i \in \{1, \ldots, n\}$, and $B_{n+1} = \mathcal{B}_\perp \setminus [\![T_r]\!]$. We have

$$\{\omega \mid \omega \not\models PS\} = \bigcup_{\substack{f(t_1,\ldots,t_n) \in T_\Sigma(\mathcal{N}) \\ ps \in PS_f}} r(f(t_1, \ldots, t_n), ps).$$

By the reasoning above, we know that $\mu(r(f(t_1, \ldots, t_n), ps)) = 0$ for all $f(t_1, \ldots, t_n) \in T_\Sigma(\mathcal{N})$ and all $ps \in PS_f$, and we conclude that

$$\begin{aligned} &\mu_w(\{\omega \mid \omega \not\models PS\}) \\ &= \mu_w\left(\textstyle\bigcup_{\substack{f(t_1,\ldots,t_n) \in T_\Sigma(\mathcal{N}) \\ ps \in PS_f}} r(f(t_1, \ldots, t_n), ps)\right) \\ &= \textstyle\sum_{\substack{f(t_1,\ldots,t_n) \in T_\Sigma(\mathcal{N}), \\ ps \in PS_f}} \mu_w(r(f(t_1, \ldots, t_n), ps)) \\ &= 0. \end{aligned} \tag{2}$$

Let $t_1, t_2 \in T_\Sigma(\mathcal{N})$ be terms such that $t_1 \approx_R t_2$, $b_2, b_2 \in \mathcal{B}$ be distinct non-$\perp$ bitstrings, and $r = r^2_{i=1}(t_i, \{b_i\})$. Let $\iota \in I(K(r))$ and $\omega \in \iota(r)$. Since $\omega \in r$, we always have $\omega(t_i) \subseteq \{b_i\}$ for $i \in \{1, 2\}$. Now, if $ran(\iota(t_i)) = \mathcal{B}^\emptyset_\perp$ for some $i \in \{1, 2\}$, we also have $\omega \in \{\perp\}$ for all $\omega \in \iota(r)$. Thus, we have

$$\prod_{t \in K(r)} \frac{\left|[\![supp_{\iota(r)}]\!](\boldsymbol{t})\right|}{\left|[\![supp_{\iota(r),PS}]\!](\boldsymbol{t})\right|} = 0$$

for all $\iota \in I(K(r))$ such that $\iota(t_i) = \mathcal{B}^\emptyset_\perp$ for some $i \in \{1, 2\}$. Now suppose that $ran(\iota(t_i)) \neq \mathcal{B}^\emptyset_\perp$ for $i \in \{1, 2\}$. Then, we have $\omega(t_i\!\downarrow) \in \{b_i\}$ for $i \in \{1, 2\}$, and since $t_1\!\downarrow = t_2\!\downarrow$, we conclude that $\omega(t_1\!\downarrow) \in \{b_1\} \cap \{b_2\} = \emptyset$ for all $\omega \in \iota(r)$. We conclude that $[\![supp_{\iota(r)}]\!](\boldsymbol{t_1}\!\downarrow) = \emptyset$, and thus

$$\prod_{t \in K(r)} \frac{\left|[\![supp_{\iota(r)}]\!](\boldsymbol{t})\right|}{\left|[\![supp_{\iota(r),PS}]\!](\boldsymbol{t})\right|} = 0$$

for all $\iota \in I(K(r))$ such that $\iota(t_i) \neq \mathcal{B}^\emptyset_\perp$ for $i \in \{1, 2\}$.

Thus, we have

$$\mu_w(r) = \sum_{\iota \in I(K(r))} \left(\prod_{t \in K(r)} \frac{\left|[\![supp_{\iota(r)}]\!](\boldsymbol{t})\right|}{\left|[\![supp_{\iota(r),PS}]\!](\boldsymbol{t})\right|} \right) = 0.$$

Now, if $\omega \not\models \approx_R$, we must have $\omega(t_1) = b_1$ and $\omega(t_2) = b_2$ for some terms t_1, t_2 such that $t_1 \approx_R t_2$ and some distinct bitstrings $b_1, b_2 \in \mathcal{B}$. Thus, we have

$$\{\omega \mid \omega \not\models \approx_R\} = \bigcup_{\substack{t_1 \approx_R t_2 \\ b_1, b_2 \in \mathcal{B}, b_1 \neq b_2}} r^2_{i=1}(t_i, \{b_i\}).$$

By the reasoning above, we know that $\mu(r^2_{i=1}(t_i, \{b_i\})) = 0$ for all such t_1, t_2, b_1, b_2, and we conclude that

$$\begin{aligned} &\mu_w(\{\omega \mid \omega \not\models \approx_R\}) \\ &= \mu_w\left(\textstyle\bigcup_{\substack{t_1 \approx_R t_2 \\ b_1, b_2 \in \mathcal{B}, b_1 \neq b_2}} r^2_{i=1}(t_i, \{b_i\})\right) \\ &= \textstyle\sum_{\substack{t_1 \approx_R t_2 \\ b_1, b_2 \in \mathcal{B}, b_1 \neq b_2}} \mu_w(r^2_{i=1}(t_i, \{b_i\})) \\ &= 0. \end{aligned} \tag{3}$$

From (1), (2) and (3), we conclude our result. ■

Lemma VII.6. *There exists a unique extension of μ to $\mathcal{F}$ that is a probability measure.*

Lemma VII.7. *$\mu(\approx_R, PS, [\![\cdot]\!])$ satisfies the following properties.*

(1) *If $a_1, \ldots, a_n \in \mathcal{N}$ are distinct names, $T_1, \ldots, T_n$ are types, $B_1, \ldots, B_n$ are sets of bitstrings, and $a_i \in T_i$ for $i \in \{1, \ldots, n\}$, then*

$$P_\mu[\bigwedge_{i=1}^n (\boldsymbol{a_i} \in B_i)] = \prod_{i=1}^n \frac{|B_i \cap [\![T_i]\!]|}{|[\![T_i]\!]|}.$$

(2) $\mu(\Omega_{[\![\cdot]\!], PS, \approx_R}) = 1$.

(3) *If $a_1, a_2 \in \mathcal{N}_w$ are weak names, $\theta = \{a_1 \mapsto a_2\}$, $t_1, t_2 \in T_\Sigma(\mathcal{N})$ are such that $t_1\theta \approx_R t_2\theta$, and $P_\mu[\boldsymbol{a_1} = \boldsymbol{a_2}] > 0$, then*

$$P_\mu[\boldsymbol{t_1} \neq \boldsymbol{t_2}, \boldsymbol{t_1} \neq \perp, \boldsymbol{t_2} \neq \perp \mid \boldsymbol{a_1} = \boldsymbol{a_2}] = 0.$$

Proof:

(1): Suppose that $r = r^n_{i=1}(a_i, B_i)$. Let

$$N_w(r) = \{w_1, \ldots, w_k\}$$

be the set of weak names in $terms(r) = \{a_i \mid i \in \{i, \ldots, n\}\}$, and

$$\{s_1, \ldots, s_{n-k}\} = terms(r) \setminus N_w(r)$$

be the set of strong names. Let $B_{w,1}, \ldots, B_{w,k}, B_{s,1}, \ldots, B_{s,n-k}$ be the sets of bitstrings such that

$$r = (r^k_{i=1}(w_i, B_{w,i})) \cap (r^{n-k}_{i=1}(s_i, B_{s,i}).$$

Fix a partition $p = \{p_1, \ldots, p_m\}$ of $N_w(r)$. For each $w \in N_w(r)$, let $j(w)$ be the only index such that $w \in p_{j(a)}$. For each $j \in \{1, \ldots, m\}$, we let $w^j \in p_j$ be the representative of the class p_j, so that $w\theta = w^j$ for all $w \in p_j$.

Define

$$\begin{aligned}\Lambda(r) = \{\lambda\colon\ \{1, \ldots, m\} \to \mathcal{B} \mid \\ \lambda(j) \in \left(\textstyle\bigcap_{w \in p_j} [\![type(w)]\!]\right), \\ j \neq j' \Rightarrow \lambda(j) \neq \lambda(j')\}\end{aligned}$$

and, for each $\lambda \in \Lambda(p)$, let

$$\Omega_\lambda = r^k_{i=1}(w_i, \{\lambda(j(w_i))\}).$$

It is simple to check that

$$\Omega_p = \biguplus_{\lambda \in \Lambda(p)} \Omega_\lambda$$

and, by property (1) of Lemma V.5, we have

$$\mu_w(\Omega_\lambda) = \prod_{i=1}^{k} \frac{1}{|[\![type(w_i)]\!]|}.$$

It follows that, for all $\lambda \in \Lambda(p)$, and any $\lambda_0 \in \Lambda(p)$,

$$\mu_w(\Omega_p) = \sum_{\lambda \in \Lambda(p)} \mu_w(\Omega_\lambda) = |\Lambda(p)|\, \mu_w(\Omega_{\lambda_0}).$$

We now observe that

$$P_{\mu_w}[r \mid \Omega_p] = \frac{P_{\mu_w}[r, \Omega_p]}{P[\Omega_p]} = \frac{\sum_{\lambda \in \Lambda(p)} P_{\mu_w}[r, \Omega_\lambda]}{|\Lambda(p)|\, \mu_w(\Omega_{\lambda_0})} \quad (4)$$

for any fixed $\lambda_0 \in \Lambda(p)$.

Similarly, for each $\lambda \in \Lambda(p)$, define

$$\Omega^*_\lambda = r^m_{j=1}(w^j, \lambda(j)).$$

Again, it is simple to check that

$$r\theta_p \cap \Omega^*_p = \biguplus_{\lambda \in \Lambda(p)} (r\theta_p \cap \Omega_\lambda)$$

and, by property (1) of Lemma V.5, we have

$$\mu_w(\Omega_\lambda) = \prod_{j=1}^{m} \frac{1}{|[\![type(w^j)]\!]|}.$$

As before, it follows that, for all $\lambda \in \Lambda(p)$, and any $\lambda_0 \in \Lambda(p)$,

$$\mu_w(\Omega^*_p) = \sum_{\lambda \in \Lambda(p)} \mu_w(\Omega_\lambda) = |\Lambda(p)|\, \mu_w(\Omega_{\lambda_0}).$$

Analogously to Equation (4), we obtain

$$P_{\mu_w}[r\theta_p \mid \Omega^*_p] = \frac{P_{\mu_w}[r\theta_p, \Omega^*_p]}{P[\Omega^*_p]} = \frac{\sum_{\lambda \in \Lambda(p)} P[r\theta_p, \Omega^*_\lambda]}{|\Lambda(p)|\, \mu_w(\Omega^*_{\lambda_0})} \quad (5)$$

for any fixed $\lambda_0 \in \Lambda(p)$.

For each $j \in \{1, \ldots, m\}$, let:

$$B_{\lambda,j} = \{\lambda(j)\} \cap \left(\bigcap_{\{i \mid w_i \in p_j\}} B_{w,i} \right).$$

We then conclude that, for all $\lambda \in \Lambda(p)$,

$$r \cap \Omega_\lambda = r^k_{i=1}(w_i, B_{\lambda,j(w_i)}) \cap r^{n-k}_{i=1}(s_i, B_{s,i})$$

and

$$r\theta_p \cap \Omega^*_\lambda = r^m_{j=1}(a^j, B_{\lambda,j}) \cap r^{n-k}_{i=1}(s_i, B_{s,i}).$$

For all $\lambda \in \Lambda(p)$, we have

$$K(r \cap \Omega_\lambda) = K(r\theta \cap \Omega^*_\lambda) = \emptyset,$$

and thus

$$I(K(r \cap \Omega_\lambda)) = I(K(r\theta \cap \Omega^*_\lambda)) = \{\emptyset\}.$$

Thus, for all $\iota \in I(K(r \cap \Omega_\lambda))$:

- if $i \in \{1, \ldots, k\}$, then

  $$[\![supp_{\iota(r \cap \Omega_\lambda)}]\!](w_i) = B_{\lambda,j(w_i)}$$

  and

  $$[\![supp_{\iota(r \cap \Omega_\lambda),PS}]\!](w_i) = [\![type(w_i)]\!];$$

- if $i \in \{1, \ldots, n-k\}$, then

  $$[\![supp_{\iota(r \cap \Omega_\lambda)}]\!](s_i) = B_{s,i} [\![type(s_i)]\!]$$

  and

  $$[\![supp_{\iota(r \cap \Omega_\lambda),PS}]\!](s_i) = [\![type(s_i)]\!].$$

Analogously, for $\iota \in I(K(r\theta_p \cap \Omega^*_\lambda))$:

- if $j \in \{1, \ldots, m\}$, then

  $$[\![supp_{\iota(r\theta_p \cap \Omega^*_\lambda)}]\!](w^j) = B_{\lambda,j)}$$

  and

  $$[\![supp_{r\theta_p \cap \Omega^*_\lambda),PS}]\!](w^j) = [\![type(w^j)]\!];$$

- if $i \in \{1, \ldots, n-k\}$, then

  $$[\![supp_{\iota(r\theta_p \cap \Omega^*_\lambda)}]\!](s_i) = B_{s,i} \cap [\![type(s_i)]\!]$$

  and

  $$[\![supp_{\iota(r\theta_p \cap \Omega^*_\lambda),PS}]\!](s_i) = [\![type(s_i)]\!].$$

By letting

$$B_S = \prod_{i=1}^{n-k} \frac{|B_{s,i} \cap [\![type(s_i)]\!]|}{|[\![type(s_i)]\!]|}$$

and

$$\mathbf{1}_\lambda = \begin{cases} 1 & \text{if } \lambda(j) \in \bigcap_{\{i \mid w_i \in p_j\}} B_{w,i} \\ 0 & \text{otherwise} \end{cases},$$

we obtain, for any fixed $\lambda_0 \in \Lambda(p)$,

$$\begin{aligned} &\mu_w(r \cap \Omega_\lambda) \\ &= B_S \left(\textstyle\prod_{i=1}^{k} \frac{|B_{\lambda,j(w_i)}|}{|[\![type(w_i)]\!]|} \right) \\ &= B_S \cdot \mu_w(\Omega_{\lambda_0}) \cdot \mathbf{1}_\lambda \end{aligned} \tag{6}$$

and

$$\begin{aligned} &\mu_w(r\theta_p \cap \Omega^*_\lambda) \\ &= B_S \left(\textstyle\prod_{j=1}^{m} \frac{|B_{\lambda,j}|}{|[\![type(w^j)]\!]|} \right) \\ &= B_S \cdot \mu_w(\Omega^*_{\lambda_0}) \cdot \mathbf{1}_\lambda. \end{aligned} \tag{7}$$

Combining Equations (4) and (6), we obtain

$$P_{\mu_w}[r \mid \Omega_p] = \frac{B_S}{|\Lambda(p)|} \sum_{\lambda \in \Lambda(p)} \mathbf{1}_\lambda.$$

Similarly, combining Equations (5) and (7),

$$P_{\mu_w}[r\theta_p \mid \Omega^*_p] = \frac{B_S}{|\Lambda(p)|} \sum_{\lambda \in \Lambda(p)} \mathbf{1}_\lambda.$$

Thus,

$$P_{\mu_w}[r \mid \Omega_p] = P_{\mu_w}[r\theta_p \mid \Omega^*_p].$$

Now (1) follows from the total probability theorem, because

$$\begin{aligned} &P_\mu[\textstyle\bigwedge_{i=1}^{n}(\boldsymbol{a_i}, B_i)] \\ &= \textstyle\sum_{p \in N_w(r)} P_{\mu_w}[r\theta_p \mid \Omega^*_p]\mu_w(\Omega_p) \\ &= \textstyle\sum_{p \in N_w(r)} P_{\mu_w}[r \mid \Omega_p]\mu_w(\Omega_p) \\ &= P_{\mu_w}[r] = \textstyle\prod_{i=1}^{n} \frac{|B_i \cap [\![T_i]\!]|}{|[\![T_i]\!]|}, \end{aligned}$$

by Lemma V.5.

(2): The fact that

$$\mu \models [\![\cdot]\!] \tag{8}$$

follows trivially from property (1) of this lemma and property (1) of Lemma V.5.

To see that μ satisfies PS, define, for each $f(t_1, \ldots, t_n) \in T_\Sigma(\mathcal{N})$ and each $ps \in PS_f$,

$$r(f(t_1, \ldots, t_n), ps) = r^{n+1}_{i=1}(t_i, B_i),$$

where $t_{n+1} = f(t_1, \ldots, t_n)$, $(B_1, \ldots, B_n) = [\![dom(ps)]\!]$, and $B_{n+1} = \mathcal{B}_\perp \setminus [\![ran(ps)]\!]$, as in the corresponging proof for μ_w.

Fix $f(t_1, \ldots, t_n)$ and ps, and let $r = r(f(t_1, \ldots, t_n)), ps$. We have:

$$\begin{aligned} \mu(r) &= \textstyle\sum_{p \in P(N_w(r))} P_{\mu_w}[r\theta_p \mid \Omega^*_p]\mu_w(\Omega_p) \\ &= \textstyle\sum_{p \in P(N_w(r))} P_{\mu_w}[r(f(t_1\theta_p, \ldots, t_n\theta_p), ps) \mid \Omega^*_p]\mu_w(\Omega_p) \\ &= 0. \end{aligned} \tag{9}$$

Thus:

$$\begin{aligned} &\mu(\{\omega \mid \omega \not\models PS\}) \\ &= \mu\left(\textstyle\bigcup_{\substack{f(t_1,\ldots,t_n) \in T_\Sigma(\mathcal{N}) \\ ps \in PS_f}} r(f(t_1, \ldots, t_n), ps) \right) \\ &= \textstyle\sum_{\substack{f(t_1,\ldots,t_n) \in T_\Sigma(\mathcal{N}), \\ ps \in PS_f}} \mu(r(f(t_1, \ldots, t_n), ps)) \\ &= 0. \end{aligned}$$

We conclude that

$$\mu \models PS. \tag{10}$$

To prove that μ satisfies $\approx_R$, let $t_1, t_2 \in T_\Sigma(\mathcal{N})$ be such that $t_1 \approx_R t_2$, and $b_1, b_2 \in \mathcal{B}$ be two distinct bitstrings. Letting $r = r^2_{i=1}(t_i, \{b_i\})$, we have

$$\begin{aligned} \mu(r) &= \textstyle\sum_{p \in P(N_w(r))} P_{\mu_w}[r\theta_p \mid \Omega^*_p]\mu_w(\Omega_p) \\ &= \textstyle\sum_{p \in P(N_w(r))} P_{\mu_w}[\left(r^2_{i=1}(t_i\theta_p, \{b_i\})\right) \mid \Omega^*_p]\mu_w(\Omega_p) \\ &= 0, \end{aligned} \tag{11}$$

where the last equality holds because the the equivalence $t_1 \approx_R t_2$ is preserved by any substitution on names θ: that is, for all $\theta \colon \mathcal{N} \to \mathcal{N}$, we have $t_1\theta \approx_R t_2\theta$.

We conclude that

$$\begin{aligned} &\mu(\{\omega \mid \omega \not\models \approx_R\}) \\ &= \mu\left(\textstyle\bigcup_{\substack{t_1 \approx_R t_2 \\ b_1, b_2 \in \mathcal{B}, b_1 \neq b_2}} r^2_{i=1}(t_i, \{b_i\}) \right) \\ &= 0. \end{aligned} \tag{12}$$

Thus,

$$\mu \models \approx_R, \tag{13}$$

and (2) follows from Equations (8), (10) and (13).

(3): Let

$$\Omega_{t_1 \neq t_2} = \{\omega \mid \omega(t_1) \neq \omega(t_2), \omega(t_1) \neq \perp, \omega(t_2) \neq \perp\}.$$

We have

$$\Omega_{t_1 \neq t_2} = \bigcup_{b_1 \in \mathcal{B}} \bigcup_{b_2 \in \mathcal{B} \setminus \{b_1\}} r^2_{i=1}(t_i, \{b_1\}).$$

Thus, it is sufficient to prove that

$$P_\mu[r^2_{i=1}(t_i, \{b_i\}) \mid \boldsymbol{a_1} = \boldsymbol{a_2}] = 0$$

whenever b_1, b_2 are distinct non-$\perp$ bitstrings. [18]

■

Theorem VII.8. *Suppose that:*

[18] **(FLAG:)**

(1) $\approx$ *is generated by a subterm convergent rewriting system R such that, for each rewrite rule $(l \to r) \in R$, if $l = f(l_1, \ldots, l_n)$, then f does not occur in any other rewrite rule or any strict subterm of l — that is,*

$$(l' \to r') \in R \setminus \{l \to r\} \Rightarrow l' \in T_{\Sigma\setminus\{f\}}(\mathcal{N})$$

and

$$l_1, \ldots, l_n \in T_{\Sigma\setminus\{f\}}(\mathcal{N}).$$

(2) $(\approx, PS, (\llbracket\cdot\rrbracket_\eta)_{\eta\in\mathbb{N}})$ *is an admissible model of* $(\widehat{\llbracket\cdot\rrbracket}_\eta)_{\eta\in\mathbb{N}}$.
(3) *Our assumptions on names and functions are satisfied.*
(4) $\phi = \nu\tilde{n}.\sigma$ *is such that* $names(ran(\sigma)) \cap \mathcal{N}_w = \{s\}$ *and, for all equational verifiers* $(t, t') \in eqv(\phi, s)$,

$$P_{\mu_\eta}[\boldsymbol{t} = \bot], P_{\mu_\eta}[\boldsymbol{t'} = \bot], P_{\widehat{\mu}_\eta}[\boldsymbol{t} = \bot], P_{\widehat{\mu}_\eta}[\boldsymbol{t'} = \bot]$$

are negligible as a function of η. [19]

Let

$$\Omega_{\phi,\eta} = \{\omega \mid \omega(t) = \omega(t') \text{ for all } (t, t') \in eqv(\phi, s), \\ \omega(t) \in \llbracket TT\rrbracket_\eta \text{ for all } (t, TT) \in tv(\phi, s)\}$$

Then,

$$\left|P_{\mu_\eta}[\Omega_{\phi,\eta}] - P_{\widehat{\mu}_\eta}[\Omega_{\phi,\eta}]\right|$$

is negligible (as a function of η).

Proof: As in the proof of Lemma V.8, let

$$\Omega_{t=t'} = \{\omega \in \boldsymbol{\Omega} \mid \omega(t) = \omega(t')\}$$

and

$$\Omega_{t\in B} = \{\omega \in \boldsymbol{\Omega} \mid \omega(t) \in B\}$$

for all terms $t, t' \in T_\Sigma(\mathcal{N})$ and all sets of bitstrings $B \subseteq \mathcal{B}_\bot$.

Let w be the fresh name used for generating the sets $eqv(\phi, s)$ and $tv(\phi, s)$ as defined in Section IV, and define

$$\Omega_w = \{\omega \mid \omega(w) \neq \omega(s)\}$$

and

$$\Omega_s = \{\omega \mid \omega(w) = \omega(s)\}.$$

Clearly, $\boldsymbol{\Omega} = \Omega_w \uplus \Omega_s$.

By Lemma V.8 and our assumptions on names, we also have

$$\begin{aligned} P_{\mu_\eta}[\Omega_w] &= \tfrac{|\llbracket T\rrbracket_\eta|-1}{|\llbracket T\rrbracket_\eta|} = P_{\widehat{\mu}_\eta}[\Omega_w], \\ P_{\mu_\eta}[\Omega_s] &= \tfrac{1}{|\llbracket T\rrbracket_\eta|} = P_{\widehat{\mu}_\eta}[\Omega_s]. \end{aligned} \tag{14}$$

For each $\eta \in \mathbb{N}$, let

$$\widehat{\boldsymbol{\Omega}}_\eta = \left\{\llbracket\cdot\rrbracket_\eta^\rho \mid \rho \text{ is samplable from } \llbracket\cdot\rrbracket_\eta\right\}$$

as in Section II-B. If ρ is an assignment samplable from $\llbracket\cdot\rrbracket_\eta$ such that $\llbracket w\rrbracket_\eta^\rho = \llbracket s\rrbracket_\eta^\rho$, then by definition of $\llbracket\cdot\rrbracket_\eta^\rho$ we necessarily have

$$\llbracket t\sigma_s\rrbracket_\eta^\rho = \llbracket t\sigma_w\rrbracket_\eta^\rho$$

for all terms t. Because we assume that $\mu \models \approx_R$, it follows from condition (4) that

$$P_{\widehat{\mu}_\eta}[\boldsymbol{t\sigma_w} = \boldsymbol{t'\sigma_w} \mid \Omega_s] = 1$$

whenever $(t, t') \in eqv(\phi, s)$, and so

$$P_{\widehat{\mu}_\eta}[\Omega_{eqv(\phi,s)}] = P_{\widehat{\mu}_\eta}\left[\bigcap_{(t,t')\in eqv(\phi,s)} \Omega_{t=t'}\right] = 1.$$

Since each $\widehat{\mu}_\eta$ is assumed to satisfy $\llbracket\cdot\rrbracket_\eta$ and PS, we also conclude that

$$P_{\widehat{\mu}_\eta}[\boldsymbol{t\sigma_w} \in \llbracket TT\rrbracket_\eta] \geq P_{\widehat{\mu}_\eta}[\boldsymbol{t\sigma_w} \in \llbracket supp_{PS}\rrbracket_\eta] = 1,$$

whenever $(t, TT) \in tv(\phi, s)$. Thus,

$$P_{\widehat{\mu}_\eta}[\Omega_{tv(\phi,s)}] = P_{\widehat{\mu}_\eta}\left[\bigcap_{(t,TT)\in tv(\phi,s)} \Omega_{t\in\llbracket TT\rrbracket_\eta}\right] = 1,$$

and we conclude

$$P_{\widehat{\mu}_\eta}[\Omega_{\phi,\eta} \mid \Omega_s] = 1. \tag{15}$$

We obtain a similar conclusion about the probability distributions μ_η. If $(t, t') \in eqv(\phi, s)$, then property (3) of Lemma V.8 implies $P_{\mu_\eta}[\boldsymbol{t} = \boldsymbol{t'} \mid \Omega_s] = 1$, and we get

$$P_{\widehat{\mu}_\eta}[\Omega_{eqv(\phi,s)}] = P_{\widehat{\mu}_\eta}\left[\bigcap_{(t,t')\in eqv(\phi,s)} \Omega_{t=t'}\right] = 1.$$

If $(t, TT) \in tv(\phi, s)$, then because μ_η satisfies $\llbracket\cdot\rrbracket_\eta$ and PS, we have $P_{\mu_\eta}[\boldsymbol{t\sigma_w} \in \llbracket TT\rrbracket_\eta]$, from which it follows that

$$P_{\mu_\eta}[\Omega_{tv(\phi,s)}] = P_{\mu_\eta}\left[\bigcap_{(t,TT)\in tv(\phi,s)} \Omega_{t\in\llbracket TT\rrbracket_\eta}\right] = 1.$$

Thus,

$$P_{\mu_\eta}[\Omega_{\phi,\eta} \mid \Omega_s] = 1. \tag{16}$$

Now, if $|\llbracket T\rrbracket|_\eta = 1$, then $P_{\mu_\eta}[\Omega_s] = P_{\widehat{\mu}_\eta}[\Omega_s] = 1$, and equations (16) and (15) imply that

$$\left|P_{\mu_\eta}[\Omega_{\phi,\eta}] - P_{\widehat{\mu}_\eta}[\Omega_{\phi,\eta}]\right| = 0.$$

We thus assume, without loss of generality, that $|\llbracket T\rrbracket|_\eta > 1$ for all η. In this case, we have $P_{\mu_\eta}[\Omega_w] = P_{\widehat{\mu}_\eta}[\Omega_w] > 0$, and, by the total probability theorem,

$$\begin{aligned} P_{\widehat{\mu}_\eta}[\Omega_{\phi,s}] = \;& P_{\widehat{\mu}_\eta}[\Omega_{\phi,s} \mid \Omega_w]P_{\widehat{\mu}_\eta}[\Omega_w] \\ & + P_{\widehat{\mu}_\eta}[\Omega_{\phi,s} \mid \Omega_s]P_{\widehat{\mu}_\eta}[\Omega_s], \\ P_{\mu_\eta}[\Omega_{\phi,s}] = \;& P_{\mu_\eta}[\Omega_{\phi,s} \mid \Omega_w]P_{\mu_\eta}[\Omega_w] \\ & + P_{\mu_\eta}[\Omega_{\phi,s} \mid \Omega_s]P_{\mu_\eta}[\Omega_s]. \end{aligned} \tag{17}$$

[19] **(FLAG: I wonder if, for marketing / beauty purposes, it would not be better to simply require that all terms that occur in the verifiers have negligible probability of being $\bot$ in both measures. Such a condition is prettier, agrees well with intuition, and I conjecture that it can have the psychological effect of making people dismiss the whole $\bot$ thing as a "technicality" (which is mostly what it is, anyway). Let me know...)**

From equations (14), (15) (16) and (17), we conclude that

$$\begin{aligned}&\left|P_{\mu_\eta}[\Omega_{\phi,\eta}] - P_{\widehat{\mu}_\eta}[\Omega_{\phi,\eta}]\right| \\ &= \left|P_{\mu_\eta}[\Omega_{\phi,\eta} \mid \Omega_w]P_{\mu_\eta}[\Omega_w] - P_{\widehat{\mu}_\eta}[\Omega_{\phi,\eta} \mid \Omega_w]P_{\widehat{\mu}_\eta}[\Omega_w]\right| \\ &= \frac{|[\![T]\!]_\eta|-1}{|[\![T]\!]_\eta|}\left|P_{\mu_\eta}[\Omega_{\phi,\eta} \mid \Omega_w] - P_{\widehat{\mu}_\eta}[\Omega_{\phi,\eta} \mid \Omega_w]\right|.\end{aligned}$$

The above expression is negligible if and only if

$$\left|P_{\mu_\eta}[\Omega_{\phi,\eta} \mid \Omega_w] - P_{\widehat{\mu}_\eta}[\Omega_{\phi,\eta} \mid \Omega_w]\right| \tag{18}$$

is negligible. We thus focus on proving this fact.

We first note that, by hypothesis (4) and our construction, the only weak names occurring in our set of verifiers are w and s. Suppose now that $(t, t') \in eqv(\phi, s)$. In this case, $\Omega_w = \Omega_p$, where p is the trivial discrete partition of the weak names occurring in t and t'. Now, we note that

$$\begin{aligned}&P_{\mu_\eta}[\boldsymbol{t} = \boldsymbol{t'} \mid \Omega_w] \\ &\leq P_{\mu_\eta}[\boldsymbol{t} = \boldsymbol{t'}, \boldsymbol{t} \neq \bot, \boldsymbol{t'} \neq \bot \mid \Omega_w] \\ &\qquad + P_{\mu_\eta}[\boldsymbol{t} = \boldsymbol{t'} \text{ and either } \boldsymbol{t} = \bot \text{ or } \boldsymbol{t'} = \bot \mid \Omega_w].\end{aligned}$$

The first of these parcels is negligible by our assumptions on functions; the second is negligible by condition (4). Thus, we conclude that

$$P_{\mu_\eta}[\Omega_{\phi,\eta}] \leq P_{\mu_\eta}[\boldsymbol{t} = \boldsymbol{t'} \mid \Omega_w]$$

is negligible. An entirely analogous reasoning allows us to conclude that

$$P_{\widehat{\mu}_\eta}[\Omega_{\phi,\eta}] \leq P_{\widehat{\mu}_\eta}[\boldsymbol{t} = \boldsymbol{t'} \mid \Omega_w]$$

is also negligible. Thus, we conclude that if $eqv(\phi, s) \neq \emptyset$, then (18) is the difference of two negligile functions, and is therefore also negligible.

Let us now consider the case that $eqv(\phi, s) = \emptyset$. Inspecting the proof of Lemma III.3, it becomes clear that the support $[\![supp_{PS}]\!](\boldsymbol{t})$ of any term t only depends on the structure of function application and the types of names that occur in each position on the term $t\!\downarrow$. In particular, if $\theta = \{w \mapsto s\}$ and $(t\sigma_s)\!\downarrow = ((t\sigma_w)\!\downarrow)theta$ — that is, if the normal form of $t\sigma_s$ is the same as the normal form of $t\sigma_w$ after replacing all occurrences of w by s —, then $[\![supp_{PS}]\!](\boldsymbol{t\sigma_s}) = [\![supp_{PS}]\!](\boldsymbol{t\sigma_s})$.

Now, suppose that $(t, TT) \in tv(\phi, s)$.

■