Computationally Complete Symbolic Attacker and Key Exchange

Gergei Bana
INRIA, Paris, France
bana@math.upenn.edu

Koji Hasebe
Graduate School of Systems and Information Engineering, University of Tsukuba
Tsukuba, Japan
hasebe@iit.tsukuba.ac.jp

Mitsuhiro Okada
Department of Philosophy, Keio University, Tokyo, Japan
mitsu@abelard.flet.keio.ac.jp

ABSTRACT

Recently, Bana and Comon-Lundh [8] introduced the notion of computationally complete symbolic attacker to deliver unconditional computational soundness to symbolic protocol verification. First we explain the relationship between their technique and Fitting's embedding of classical logic into S4 [25]. Then, based on predicates for "key usability", we provide an axiomatic system in their framework to handle secure encryption when keys are allowed to be sent. We examine both IND-CCA2 and KDM-CCA2 encryptions, both symmetric and asymmetric situations. We also consider INT-CTXT ciphertext integrity. This technique does not require the usual limitations of computational soundness such as the absence of dynamic corruption, the absence of key-cycles or unambiguous parsing of bit strings. In particular, if a key-cycle possibly corrupts CCA2 encryption, our technique delivers an attack. If it does not endanger security, the security proof goes through. We illustrate how our notions can be applied in protocol proofs.

1. INTRODUCTION

Approaches to computationally sound automated verification of security protocols can be divided into two groups. Works in one [3, 22, 4, 18, 26, 21] define *symbolic adversaries*, and soundness theorems state that under certain circumstances, if there is no successful symbolic attack, then there is no successful computational attack either. The other group aims to work directly in the computational model [23, 15, 10, 11, 9]. In this latter case, computational soundness means that the properties on which symbolic manipulations are conditioned hold computationally.

The first group, where a symbolic attacker is defined, gives hope that already existing automated tools may be adopted for computationally sound verification, but these soundness theorems require large sets of assumptions. A number of assumptions, as well as reasons why they are not realistic are discussed in [19]. Such assumptions are, for example, that bit strings can be unambiguously parsed into symbolic terms, or, that no key cycles occur, or, that all keys are honestly generated, or, that there is no dynamic corruption. Recently, Backes et al. in [2] showed a way to avoid some of these problems such as key-cycles and badly generated keys, but for the computational implementation of the encryption, they needed to require a very strong notion called PROG-KDM security. Moreover, they still used an entire page to list all the further necessary conditions (such as unambiguous parsing) limiting the computational implementation that they needed for soundness. But PROG-KDM security and the other conditions are necessary only to receive computational guarantees for their symbolic analysis even if computational security of the analyzed protocol holds without these requirements. *Their strong conditions are imposed on the computational implementation not for the security of the protocol, but for the soundness of the analysis.*

Recently, Bana and Comon-Lundh (BC) presented in [8] a new kind of symbolic attacker. They called it *computationally complete symbolic adversary*, as it is capable of doing everything that a computational adversary is capable of. They observed that the discrepancy between symbolic and computational proofs emerges from the following fact: While the usual computational security assumptions on the primitives (such as IND-CCA2 security of the encryption) define what the adversary cannot violate (and the security of the protocol is derived from the security of the primitives), symbolic adversaries are defined by listing all the adversarial capabilities (Dolev-Yao rules). Hence, to adjust the viewpoint of the symbolic analysis to that of the computational, instead of listing every kind of moves a symbolic adversary is allowed to do, Bana and Comon-Lundh list a few rules (axioms) that the symbolic adversary is not allowed to violate. Anything that does not contradict these axioms is allowed for the adversary. Hence, a successful symbolic attack in their case means that the violation of the security property of the protocol is consistent with the axioms. The axioms that are introduced must be computationally sound with respect to the computational interpretation they defined. Their *general soundness* result is the following: *Suppose that the computational implementation satisfies a set of axioms. If there is a successful PPT attacker for which the number of sessions it exploits does not increase indefinitely as the security parameter increases (there is a bound, but it can be arbitrarily high), then there is also a successful symbolic adversary complying with the set of axioms.*

The difference between the original Dolev-Yao (DY) technique and that of BC can be best understood from the following pictures. In the DY technique, as more and more rules are added, the symbolic adversarial capabilities are increasing, the symbolic adversary covers more and more of the computational capabilities. However, no-one has been able to come up with rules that properly cover all possible computational capabilities. As Figure 1 shows, there are always some computational capabilities that are not covered by the DY ones. All computational soundness results that use the DY

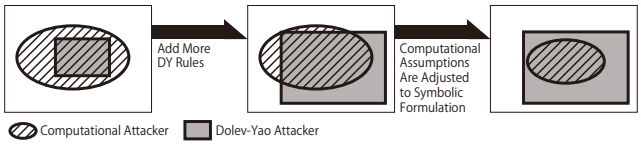

Figure 1: Soundness of the DY Adversary

symbolic adversaries in the end have to impose some significant limitations on the computational implementation.

In the BC approach, without axioms, the symbolic adversary is allowed to do anything. As axioms are added, the symbolic adversary's capabilities are decreasing. Their main theorem is that if the

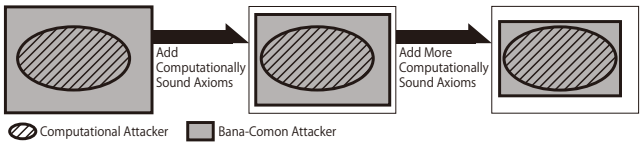

Figure 2: Soundness of the BC Adversary

axioms are computationally sound, the symbolic adversarial capabilities cover all of the computational adversarial capabilities that use bounded number of sessions. This is illustrated in Figure 2. Clearly, if the symbolic adversary is too strong, security of protocols cannot be proven. Therefore, the aim is to create a *library of axioms* that are sound and are sufficient to prove actual protocols.

In [6], Bana et al. introduced several modular, computationally sound axioms, and verified secrecy and authentication of the Needham-Schroeder-Lowe protocol to illustrate that the technique can indeed be used to verify actual protocols. However, as Backes et al. have pointed out in [2], the axioms in [6] were not suitable when decryption keys were sent around in the course of the protocol (under encryptions for example, in a key distribution). The current work aims to address this problem.

Before we describe this current work in detail, it is worth asking ourselves, does it really make sense to develop a new technique when such tools as CryptoVerif and EasyCrypt exist? However, EasyCrypt at its current stage is more for proving properties about primitives, such as CCA2 security from hardness assumptions; it is not suitable for more complex protocols. CryptoVerif is a very powerful tool for protocol analysis, but if it fails to prove a protocol, other than its developer, it is difficult for a user to know what to do. The aim of the BC technique is to construct a relatively simple, intuitive system. If we remove the explanations, the axioms in this paper are just a few lines, perhaps half a page, and most of them are trivial. The BC technique is still lightweight compared to CryptoVerif or EasyCrypt. BC does not use explicit probabilities, neither game reductions. Proofs are readable to human, convenient for human interaction if automated. NSL and the and the symmetric Needham-Schroeder protocols were proven by hand, without any assumption on parsing unambiguity. Available other proofs of the NSL protocol including the one with CryptoVerif all assume unambiguous parsing. The NSL proof with this technique reduces to 2-3 pages with unambiguous parsing. For an initial decidability result, see Comon et al. [20].

1.1 Our Work

In this paper we tackle the problem of key exchange, but along the way we also present various other improvements on the original work of Bana and Comon-Lundh. In their original work [8], the general soundness theorem worked only for certain kinds of first-order formulas, and the non-negligible subsets of the computational execution had to satisfy a certain computability property. Later, in the online version of their paper [7] they presented improved computational semantics for the disjunction and existential quantification, and with that they could make the general soundness proof work for any first-order formula. But they still needed the computability condition on the non-negligible subsets (not an important limitation though, but not pretty). In this work, we remove this limitation with a trick in the soundness proof.

Bana and Comon-Lundh proved their general soundness theorem directly from their definitions. This involved showing that although computational semantics of their compound formulas are not defined as usual in Tarskian semantics, first-order deduction rules and axioms are valid with respect to their semantics too. As it turns out, this actually follows from Fitting's theorem of embedding first-order logic into first-order S4 [25]. After introducing the basics, we detail this relationship in the current work, and show how the BC general soundness theorem follows from Fitting's theorem.

In order to tackle key exchange, the necessary element to incorporate in the framework is *key usability*, an idea introduced in [24] for a different framework. This notion is meant to express whether a properly generated key, at a certain point of the protocol execution, is still usable for secure encryption or whether it has been compromised. If a decryption key (or just a key in the symmetric case) is sent in the clear, the encryption key associated to it cannot be used for secure encryption any more. Or, a key that was sent in a key cycle, may also have lost its capability to encrypt securely if the encryption scheme is only IND-CCA2 secure. More generally, keys can also be compromised in more subtle, non-trace fashion.

For overall consistency of notation, instead of key usability we introduce the opposite, namely *key compromise* as a predicate. We define key compromise predicates both for symmetric and asymmetric encryptions, both for IND-CCA2 [13] and KDM-CCA2 [1, 16] cases, and also for INT-CTXT [14] ciphertext integrity. There is an essential difference from the way key usability was defined in [24], we explain that at our definition. (Furthermore, the axioms in [24] were introduced to work for the Diffie-Hellmann key exchange, they are not helpful with other protocols.)

Further essential innovations of this paper are predicates representing adversarial derivability (computability) with oracle access. This makes the axioms simpler than just using derivability as in [6]. Depending on whether IND-CCA2 or KDM-CCA2 oracles are used, and on whether the encryption is symmetric or asymmetric, we define four such derivability with oracle access predicates.

We introduce axioms and show that they are computationally sound. The axioms are suitable for (but not limited to) inductive reasoning: if something is uncompromised up-to a point, then certain newly sent messages do not destroy this property. They are also entirely *modular*: *Introducing further primitives will not destroy the soundness of these axioms, they do not have to be proved again.* If we want to prove a protocol that uses further primitives such as signatures besides encryption, then we only have to introduce new axioms for the new primitives. For encryption, the current axioms can still be used unchanged. Hence, a library of axioms can be gradually developed by adding more and more axioms.

A nice feature of our new predicates for key usability and derivability with oracles is that we only have a single axiom requiring CCA2 security of the encryption: the axiom stating that fresh keys are uncompromised (with respect to CCA2 security). The rest of the axioms, such as the one expressing that encryption with uncompromised key hides the plaintext, or the one expressing non-malleability are immediate consequences of the computational se-

mantics of the derivability and key-compromise predicates.

We emphasize that we introduce axioms for KDM-CCA2 security to be able to analyze protocols for which KDM security is computationally necessary: Unlike [2], in our case, for those protocols that do not require KDM security for their computational soundness, the use of our IND-CCA2 axioms is sufficient.

After presenting the axioms and their soundness proofs, we look at three simple examples in Section 11 to illustrate how the axioms work with special focus on comparing how the IND-CCA2 and the KDM-CCA2 axioms are applied to key cycles. We show security in a case when there are no key cycles, then we present a case when there is a key cycle and the IND-CCA2 axioms provide an attack while KDM-CCA2 axioms still prove security. Finally we show a case when there is a key cycle, but as it is under another encryption it does not danger security even in the IND-CCA2 case and the security proof goes through. We also show that the axioms in [6] without key compromise are not sufficient to treat these examples.

Finally, we present the result of our proof of the Amended Symmetric Needham-Schroeder Protocol. This protocol first distributes a session key, and then uses the distributed key to share a secret. Using the IND-CCA2 and INT-CTXT axioms, we proved that the key is securely distributed, that the shared nonce remains secret, and that agreement and authentication hold. It is posted online at the first author's homepage.

The technique of [8] and also this work allows to avoid *all* restrictions mentioned before on the computational world. *Once a protocol is proven secure in our symbolic model with respect to a set of axioms, then all properties that the computational implementation has to satisfy for computational security are included in the axioms.* Any number of bad keys are allowed to be generated by the adversary; any number of corrupted, uncorrupted, or dynamically corrupted parties can be present. As for parsing of bit strings into terms, previous soundness results relied on unambiguous parsing. Within this framework, there is no need for such an assumption. We do not even need the condition that encryptions, pairing are length regular (i.e. encryption, pairing of inputs that have the same length output bit strings of the same length).

As long as indistinguishability properties are not concerned, the only significant restriction remains that the technique is not capable to detect computational attacks for which the PPT attacker needs the number of sessions to grow indefinitely as the security parameter increases. (It is the general soundness theorem that requires bounded number of sessions, not the axioms.) However, the usual Dolev-Yao technique is not capable of doing this either. (ProVerif for example is unbounded only in the sense that it works for arbitrary number of sessions, but still, if an attack is found, that, and the corresponding computational attack uses a given number of sessions independently of the security parameter.) Nevertheless, it would still be nice to have conditions under which the absence of successful symbolic BC adversaries means that there are no computational attacks even if unbounded number of sessions are allowed. For example, if only CCA2 encryptions and pairings are used to construct messages, we believe that this statement holds. Analysis of this is left for future work.

The contributions of this paper include (i) relating the BC technique to Fitting's embedding, (ii) syntax and computational semantics of key compromise and derivability with oracle access, (iii) a library of axioms for symmetric and asymmetric IND-CCA2 and KDM-CCA2 encryptions and INT-CTXT encryptions as well, (iv) soundness result of the axioms, (v) short examples to illustrate how the axioms are used with an emphasis on key-cycles, (vi) summary of the verification of the symmetric Needham-Schroeder protocol with this tool as a proof of concept. (The NSL proof in [6] can also be done with the current set of axioms the same way as it was done there, with minor modifications only.)

The paper is organized as follows: In Section 2, we give the intuitive description of our new predicates. In Section 3, we present our first-order language, which is an extension of that of BC. We then summarize in Sections 4 and 5 how Bana and Comon treated the symbolic and computational executions in [8], define computational semantics, and present the general soundness theorem. In Section 6, we show how the BC technique is related to Fitting's embedding of classical logic in S4, and show how the general soundness result of Bana and Comon-Lundh follows from Fitting's theorem. Section 7 is devoted to the semantics of the new derivability predicates with oracles and their axioms, and Section 8 is the same for key usability. In Section 9 we discuss congruence of the equality predicate. In Section 10 we state and prove our soundness theorem for the axioms. In Section 11, we show a few simple examples of how inconsistency of certain formulas with the axioms can be proven. Finally, in Section 14, we state the result of the verification of the amended symmetric NS protocol with our tool.

We thank Pedro Adão, Bruno Blanchet, Rohit Chadha, Hubert Comon and Guillaume Scerri for helpful suggestions and valuable discussions.

2. DERIVABILITY, KEY COMPROMISE

The most important new aspect of the symbolic execution in [8] was to replace the DY technique's fixed definition of what the symbolic adversary can deduce, $x_1, ..., x_n \vdash y$, with a *derivability predicate*[1] $x_1, ..., x_n \rhd y$ for which the symbolic semantics is not fixed. Namely, while in the DY technique, $x_1, ..., x_n \vdash y$ meant that using only the DY rules y can be computed from $x_1, ..., x_n$, in the BC case $x_1, ..., x_n \rhd y$ is given some unfixed symbolic interpretation in an abstract model $\mathcal{M}$ for which they only require to satisfy some (computationally sound) axioms. That is, the axioms express what the symbolic adversary cannot violate. They do imply that from DY deducibility, satisfaction of the derivability predicate follows, for example, $\{y\}_K, K \rhd y$. But in the BC system these rules are not what the adversary can at most do, but what it can certainly at least do (in other words, the adversary is not allowed to be unable to do it). The idea is that symbolic interpretation of $x_1, ..., x_n \rhd y$ should be at least as powerful as computability of y from $x_1, ..., x_n$ by some probabilistic polynomial time algorithm, and so the only limitations that we want to put on symbolic satisfaction of $\rhd$ are limitations that are derived from computational computability.

One of the major innovations we propose here is that axioms become simpler if we allow the use of some oracles for the adversary. For example, considering IND-CCA2 public key encryption, it is better to introduce a new derivability predicate, $x_1, ..., x_n \rhd^{\mathrm{aic2}} y$ with the computational semantics meaning that the interpretation of y can be derived from the interpretation of $x_1, ..., x_n$ by a PPT adversary with the help of decryption oracles that decrypt everything that are not results of encryptions on the left hand side. Similarly, for the symmetric case, we can introduce $x_1, ..., x_n \rhd^{\mathrm{sic2}} y$ meaning that y can be derived from $x_1, ..., x_n$ with the help of decryption oracles and encryption oracles. That is, the algorithm trying to compute y from the $x_1, ..., x_n$ is allowed to submit strings to the oracles for encryption and also for decryption. The decryption oracle decrypts as long as the submitted string is not an encryption from $x_1, ..., x_n$ or an encryption produced by the encryption

[1]Note that in [8] Bana and Comon-Lundh denoted this predicate as $x_1, ..., x_n \vdash y$ although $\vdash$ is usually reserved for denoting deducibility in a proof system. We find that somewhat confusing, so we use the notation $x_1, ..., x_n \rhd y$ to emphasize that we do not mean some specific deducibility by it, it is a predicate.

oracle. The encryption oracles here are needed, because the adversary cannot himself do encryptions (he does not know the key), and for this reason the IND-CCA2 definition for symmetric encryption allows the submission to the encryption oracle multiple times. In fact, for uniformity, we allow it for the public case too, as IND-CCA2 is equivalent for the case of multiple submissions to encryption oracles [12]. Similarly, we will also define derivation with oracle accessibility for KDM-CCA2 encryptions, which is somewhat more tricky. But encryption oracles using which keys? The answer is, keys honestly generated by the agents during the execution. We will use the notation $\rhd^{\mathfrak{O}}$for such derivability with $\mathfrak{O}$ being either aic2, sic2, akc2, skc2 or nothing, depending on whether we want asymmetric or symmetric IND-CCA2 oracles, or asymmetric or symmetric KDM-CCA2 oracles, or no oracles. ($\mathfrak{O}$ is Fraktur O.)

Our next innovation is *key compromise* for the case when keys are sent around. We use the notation $x_1, ..., x_n \blacktriangleright^{\mathfrak{O}} K$, where $\mathfrak{O}$ again indexes whether we are talking about IND-CCA2, KDM-CCA2, symmetric or asymmetric encryption (there is always some oracle here). The intuitive meaning is that K is compromised by the messages $x_1, ..., x_n$ with access to the given oracles. For example, $K, x_2 \blacktriangleright^{\mathfrak{O}} K$. Or, $\{K\}_{K'}, K' \blacktriangleright^{\mathfrak{O}} K$. Or, if $x_1, ..., x_n \rhd^{\mathfrak{O}} K$, then $x_1, ..., x_n \blacktriangleright^{\mathfrak{O}} K$. But, presumably, if x_1 is just the first half of K, then $x_1 \blacktriangleright^{\mathfrak{O}} K$ may still hold, while $x_1 \rhd^{\mathfrak{O}} K$ does not. That is, while $x_1, ..., x_n \rhd^{\mathfrak{O}} K$ clearly implies $x_1, ..., x_n \blacktriangleright^{\mathfrak{O}} K$, the other way is not necessarily true. Nevertheless the two properties, as we will see, behave very similarly, so we chose similar notation for them. We also consider key compromise for INT-CTXT *ciphertext integrity*.

3. LANGUAGE

The core of the framework used in this paper was introduced by Bana and Comon-Lundh in [8]. Along with our new innovations, we present a brief summary of the original system of Bana and Comon-Lundh as well.

3.1 Terms, Predicates, Formulas

Terms are built out of a set of function symbols $\mathcal{F}$ that contains an unbounded set of names $\mathcal{N}$ and an unbounded set of handles $\mathcal{H}$. Names and handles are constants (zero-arity function symbols). We use names to denote items honestly generated by agents, while handles denote inputs of the adversary. Let $\mathcal{X}$ be an unbounded set of variables. A ground term is without variables.

Let $\mathcal{P}$ be a set of predicate symbols over terms. We assume here that $\mathcal{P}$ contains the binary predicate $=$ and is used as $t_1 = t_2$ and the following families of of $(n+1)$-ary predicates meaning various sorts of derivability:

- $t_1, ..., t_n \rhd_n t$ for derivability of the rhs from the lhs
- $t_1, ..., t_n \rhd_n^{\mathrm{aic2}} t$ for derivability of the rhs from the lhs with access to IND-CCA2 oracles in asymmetric case
- $t_1, ..., t_n \rhd_n^{\mathrm{sic2}} t$ for derivability of the rhs from the lhs with access to IND-CCA2 oracles in symmetric case
- $t_1, ..., t_n \rhd_n^{\mathrm{akc2}} t$ for derivability of the rhs from the lhs with access to KDM-CCA2 oracles in asymmetric case
- $t_1, ..., t_n \rhd_n^{\mathrm{skc2}} t$ for derivability of the rhs from the lhs with access to KDM-CCA2 oracles in symmetric case

and the following *key compromise* predicates:

- $t_1, ..., t_n \blacktriangleright_n^{\mathrm{aic2}} K$ meaning the lhs compromises (with oracle access) secure asymmetric IND-CCA2 encryption with K
- $t_1, ..., t_n \blacktriangleright_n^{\mathrm{sic2}} K$ meaning the lhs compromises (with oracle access) secure symmetric IND-CCA2 encryption with K
- $t_1, ..., t_n \blacktriangleright_n^{\mathrm{akc2}} K$ meaning the lhs compromises (with oracle access) secure asymmetric KDM-CCA2 encryption with K
- $t_1, ..., t_n \blacktriangleright_n^{\mathrm{skc2}} K$ meaning the lhs compromises (with oracle access) secure symmetric KDM-CCA2 encryption with K
- $t_1, ..., t_n \blacktriangleright_n^{\mathrm{ic}} K$ meaning the lhs compromises (with oracle access) INT-CTXT ciphertext integrity of encryptions with K.

We always drop the index n: $t_1, ..., t_n \rhd^{\mathfrak{O}} t$ and $t_1, ..., t_n \blacktriangleright^{\mathfrak{O}} K$.

4. SYMBOLIC EXECUTION

In case of the Dolev-Yao adversary, derivability predicates would have fixed interpretations. For example, $\rhd$ holds in case of a DY adversary, if the right-hand side can be computed from the left-hand side with the DY rules. However, *for the symbolic execution, the BC technique allows any interpretation of the predicates (including $=$) that does not contradict some axioms (introduced later).*

Accordingly, let $\mathcal{M}$ be any first-order structure that interprets the function and predicate symbols of the logic. We denote by $D_{\mathcal{M}}$ the domain of interpretation, and by $\rhd^{\mathfrak{O}}_{\mathcal{M}}$, $\blacktriangleright^{\mathfrak{O}}_{\mathcal{M}}$ and $=_{\mathcal{M}}$ the relations on $D_{\mathcal{M}}$ interpreting $\rhd^{\mathfrak{O}}$, $\blacktriangleright^{\mathfrak{O}}$, and $=$ respectively. Given an assignment σ of elements in $D_{\mathcal{M}}$ to the free variables of term t, we write $[\![t]\!]^{\sigma}_{\mathcal{M}}$ for the interpretation of $t\sigma$ in $\mathcal{M}$ ($[\![\_]\!]^{\sigma}_{\mathcal{M}}$ is the unique extension of σ into a homomorphism of $\mathcal{F}$-algebras). For any first-order formula θ, for any first-order structure $\mathcal{M}$ over the functions $\mathcal{F}$ and predicates $\mathcal{P}$, and any assignment σ of the free variables of θ in the domain of $\mathcal{M}$, the satisfaction relation $\mathcal{M}, \sigma \models \theta$ is defined as usual in first-order logic from the satisfaction of predicates.

4.1 Protocols

Bana and Comon-Lundh set up their technique to be convenient for constraint-solving methods as in [17].

Let Q be a set of *control states* (not necessarily finite). A *protocol* is a recursive set of tuples

$$((q, \overline{n}), (q', \overline{n}, \overline{n'}), \langle x_1, ..., x_m \rangle, x, \psi, s)$$

where $q, q' \in Q$, $x_1, ..., x_m, x$ are variables (into which agents read messages from the adversary), $\overline{n}, \overline{n'}$ are disjoint finite sequences of names (corresponding to honestly generated items such as keys, nonces). ψ is some formula corresponding to agent checks on incoming messages. For example, ψ can be a formula such as $dec(x, k) = n$, checking whether the input decrypts to a previously generated nonce n. ψ is over the variables $\{x_1, ..., x_m, x\}$, the names $\overline{n}$, the function symbols $\mathcal{F}$ without the rest of the names and handles, and some subset of the predicate symbols $\mathcal{P}$. Finally, s is the output message, when the transition succeeds. s is built from the variables $\{x_1, ..., x_m, x\}$, the names $\overline{n}, \overline{n}'$, and the function symbols $\mathcal{F}$ without the rest of the names and handles.

4.2 Execution of a Protocol and Attacks

In applied π-calculus, *frames* are sequences of terms with name binders: a frame ϕ can be written $\nu\overline{n}.\langle p_1 \mapsto t_1, ..., p_n \mapsto t_n \rangle$ where $p_1, ...p_n$ are place holders that do not occur in $t_1, ..., t_n$ and $\overline{n}$ is a sequence of names, but we think of a frame simply as a list of terms $\langle t_1, ..., t_n \rangle$ representing the messages that agents have sent over the network, that is, messages that the adversary has seen. The *names*, *variables* of ϕ are the names, variables of $t_1, ..., t_n$.

A *symbolic state* of the network consists of:

- a control state $q \in Q$ together with a sequence of names (randomly) generated so far, $n_1, ..., n_k$
- a sequence of constants called *handles* $h_1, ..., h_n$ (recording the attacker's inputs)

- a ground frame ϕ (the agents outputs)
- a set of closed formulas Θ (all conditions that must be satisfied in order to reach the state).

A *symbolic transition sequence* of a protocol Π is a sequence

$$((q_0, \overline{n_0}), \emptyset, \phi_0, \emptyset) \rightarrow ... \rightarrow ((q_m, \overline{n_m}), \langle h_1, ..., h_m \rangle, \phi_m, \Theta_m)$$

if, for every $m - 1 \geq i \geq 0$, there is a transition rule

$$((q_i, \overline{\alpha_i}), (q_{i+1}, \overline{\alpha_{i+1}}), \langle x_1, ..., x_i \rangle, x, \psi, s)$$

such that $\overline{n} = \overline{\alpha_{i+1}} \setminus \overline{\alpha_i}$, $\phi_{i+1} = (\phi_i, s\sigma_{i+1})$, $\overline{n_{i+1}} = (\overline{n_i}, \overline{n})$, $\Theta_{i+1} = \Theta_i \cup \{\phi_i \rhd h_{i+1}, \psi\sigma_{i+1}\}$ where $\sigma_{i+1} = \{x_1 \mapsto h_1, ..., x_i \mapsto h_i, x \mapsto h_{i+1}\}$. If necessary, some renaming of the sequence $\overline{\alpha_{i+1}}$ ensures the freshness of the names $\overline{n}$: $\overline{n} \cap \overline{n_i} = \emptyset$.

Given an interpretation $\mathcal{M}$, a transition sequence of Π

$$((q_0, \overline{n_0}), \emptyset, \phi_0, \emptyset) \rightarrow ... \rightarrow ((q_m, \overline{n_m}), \langle h_1, ..., h_m \rangle, \phi_m, \Theta_m)$$

is *valid w.r.t.* $\mathcal{M}$ if, for every $m - 1 \geq i \geq 0$, $\mathcal{M} \models \Theta_{i+1}$.

Examples of symbolic executions can be found in [8] and [6].

4.3 Symbolic Satisfaction of Formulas

$\mathcal{M}$ modeled, among others, the predicate $t_1, ..., t_n \rhd t$. In *executions* we also consider a predicate that we write as $\hat{\phi}, t_1, ..., t_n \rhd t$. This is also an $n + 1$-arity predicate. $\hat{\phi}$ is just a symbol, not an argument, and it represents the frame containing the messages that protocol agents sent out, that is, the information available from the protocol to the adversary. We also use a number of different *constraints*: $\mathsf{Handle}(h)$ means h is a handle, $\mathsf{RanGen}(x)$ means that x was honestly, randomly generated (i.e. appears in the $\overline{n}$ of the control state); $x \sqsubseteq \hat{\phi}$ means that x is a subterm of a message sent out by an agent (i.e. listed in the frame ϕ), $x \sqsubseteq \vec{x}$ means x is subterm of $\vec{x}$. $dK \sqsubseteq_{\not d} \hat{\phi}$ means dK occurs somewhere other than in a decryption position $dec(\ , dK)$ in ϕ, and $dK \sqsubseteq_{\not d} \vec{x}$ is analogous (dK may also occur in a decryption position in $\vec{x}$, but it has to occur elsewhere too)[2]. Similarly, let $K \sqsubseteq_{\not ed} \hat{\phi}$ mean that symmetric key K occurs somewhere other than in an encryption or decryption position (as $\{\!|\_|\!\}_K$ or $sdec(\ , K)$) in ϕ, and $K \sqsubseteq_{\not ed} \vec{x}$ is analogous (K may also occur in encryption or decryption position in $\vec{x}$, but it has to occur elsewhere too). Let us introduce the following abbreviations:

- $x \sqsubseteq \hat{\phi}, \vec{x} \ \equiv \ x \sqsubseteq \hat{\phi} \vee x \sqsubseteq \vec{x}$
- $\mathsf{fresh}(x; \hat{\phi}, \vec{x}) \ \equiv \ \mathsf{RanGen}(x) \wedge x \not\sqsubseteq \hat{\phi}, \vec{x}$
- $\mathsf{keyfresh}(K; \hat{\phi}, \vec{x})$ for asymmetric key:
  $\mathsf{keyfresh}(K; \hat{\phi}, \vec{x}) \ \equiv \ \mathsf{RanGen}(K) \wedge dK \not\sqsubseteq_{\not d} \hat{\phi}, \vec{x}$
- $\mathsf{keyfresh}(K; \hat{\phi}, \vec{x})$ for symmetric key:
  $\mathsf{keyfresh}(K; \hat{\phi}, \vec{x}) \ \equiv \ \mathsf{RanGen}(K) \wedge K \not\sqsubseteq_{\not ed} \hat{\phi}, \vec{x}$
- $x \preccurlyeq \hat{\phi}, \vec{x} \ \equiv \ \forall h(h \sqsubseteq x \wedge \mathsf{Handle}(h) \rightarrow \hat{\phi}, \vec{x} \rhd h)$
  $\vec{x} \preccurlyeq \hat{\phi}, \vec{y} \ \equiv \ \bigvee_p (x_{p_1} \preccurlyeq \hat{\phi}, \vec{y} \ \wedge \ x_{p_2} \preccurlyeq \hat{\phi}, \vec{y}, x_{p_1} \ \wedge \ ... \wedge x_{p_n} \preccurlyeq \hat{\phi}, \vec{y}, x_{p_1}, ..., x_{p_{n-1}})$

Where p runs through all permutations of $1, ..., n$. Further, for symmetric encryption we also require for $x \preccurlyeq \hat{\phi}, \vec{x}$ that if any R is a random input of an encryption in $\hat{\phi}, \vec{x}, x$ then the only way it can appear in x is within that same encryption.

If $\mathcal{M}$ is a first-order model, satisfaction of predicates and constraints in a *symbolic* execution (denoted by $\stackrel{s}{\models}$) is defined recursively: Let $\overline{n} = (n_1, ..., n_k)$ be a list of names and $\phi = \langle t_1, ..., t_m \rangle$ a list of closed terms. Let σ be a substitution of free variables of the rhs of $\stackrel{s}{\models}$ with elements in the domain of $\mathcal{M}$.

[2]In this paper, we will use the notation $\{x\}^R_{eK}$ and $dec(y, dK)$ for both symmetric and asymmetric encryptions with random input R, where in the symmetric case, $eK = dK = K$. We use $\{\!|x|\!\}^R_K$ and $sdec(y, K)$ for symmetric encryption and decryption only.

- Satisfactions of predicates by $\mathcal{M}, \sigma, \overline{n}, \phi$ (depends on $\mathcal{M}$):
  - $\mathcal{M}, \sigma, \overline{n}, \phi \stackrel{s}{\models} t = t'$ if $\mathcal{M}, \sigma \models t = t'$
  - $\mathcal{M}, \sigma, \overline{n}, \phi \stackrel{s}{\models} \hat{\phi}, s_1, ..., s_n \rhd^{\mathfrak{D}} t$ if $\mathcal{M}, \sigma \models t_1, ..., t_m, s_1, ..., s_n \rhd^{\mathfrak{D}} t$.
  - $\mathcal{M}, \sigma, \overline{n}, \phi \stackrel{s}{\models} \hat{\phi}, s_1, ..., s_n \blacktriangleright^{\mathfrak{D}} t$ if $\mathcal{M}, \sigma \models t_1, ..., t_m, s_1, ..., s_n \blacktriangleright^{\mathfrak{D}} t$.
- Satisfactions of constraints by $\mathcal{M}, \sigma, \overline{n}, \phi$ are independent of $\mathcal{M}$ and σ so we define them as satisfaction by $\overline{n}, \phi$:
  - $\mathsf{Handle}(h)$ for h closed term: $\overline{n}, \phi \stackrel{s}{\models} \mathsf{Handle}(h)$ if $h \in \mathcal{H}$.
  - $\mathsf{RanGen}(s)$ for s closed term: $\overline{n}, \phi \stackrel{s}{\models} \mathsf{RanGen}(s)$ if $s \in \mathcal{N}$ and $\mathcal{M}, \sigma \models s = n_1 \vee ... \vee s = n_k$.
  - $t \sqsubseteq \hat{\phi}$, where t is closed term: $\overline{n}, \phi \stackrel{s}{\models} t \sqsubseteq \hat{\phi}$ if t is a subterm of some t_i
  - $t \sqsubseteq s_1, ..., s_n$, where $s_1, ..., s_n, t$ are closed terms: $\overline{n}, \phi \stackrel{s}{\models} t \sqsubseteq s_1, ..., s_n$ if t is a subterm of some s_i
- Satisfaction of any FOL formula by $\mathcal{M}, \sigma, \overline{n}, \phi$:
  - $\theta_1 \wedge \theta_2$, $\theta_1 \vee \theta_2$, and $\neg\theta$ are interpreted as usual in FOL.
  - If x is not under a constraint in θ, interpretations of $\forall x\theta$ and $\exists x\theta$ are defined as usual in FOL.
  - If x occurs under a constraint in θ, then
    - $\mathcal{M}, \sigma, \overline{n}, \phi \stackrel{s}{\models} \forall x\theta$ iff for every ground term t, $\mathcal{M}, \sigma, \overline{n}, \phi \stackrel{s}{\models} \theta\{x \mapsto t\}$
    - $\mathcal{M}, \sigma, \overline{n}, \phi \stackrel{s}{\models} \exists x\theta$ iff there is a ground term t, $\mathcal{M}, \sigma, \overline{n}, \phi \stackrel{s}{\models} \theta\{x \mapsto t\}$
- Satisfaction at step m: $\mathcal{M}, \sigma, ((q, \overline{n}), \langle h_1, ..., h_m \rangle, \phi_m, \Theta) \stackrel{s}{\models} \theta$ iff $\mathcal{M}, \sigma, \overline{n}, \phi_m \stackrel{s}{\models} \theta$.

We say there is a *successful symbolic attack* against the security property θ (a first-order formula) of the protocol if there is a model $\mathcal{M}$ and state of an execution $((q, \overline{n}), \langle h_1, ..., h_m \rangle, \phi_m, \Theta)$ such that $\mathcal{M}, ((q, \overline{n}), \langle h_1, ..., h_m \rangle, \phi_m, \Theta) \stackrel{s}{\models} \neg\theta$ holds, and moreover, $\mathcal{M}, ((q, \overline{n}), \langle h_1, ..., h_m \rangle, \phi_m, \Theta)$ also satisfies the computationally sound axioms that we introduce in the rest of the paper. This is the same as saying that there is a successful symbolic attack if at a certain point of some symbolic execution, the axioms, the agent checks and the negation of the security property are all consistent.

EXAMPLE 4.1. We show the beginning of a possible branch in the symbolic execution of a single session of the NSL protocol.

$$(q_0, \emptyset, \phi_0, \emptyset) \quad (q_1, H_1, \phi_1, \Theta_1) \quad (q_2, H_2, \phi_2, \Theta_2) \quad (q_3, H_3, \phi_3, \Theta_3)$$

where with q_j^A, q_j^B counting the states of the A and B, $q_0 = (q_0^A, q_0^B, \overline{n_0})$, and $q_1 = (q_1^A, q_0^B, \overline{n_1})$, and $q_2 = (q_1^A, q_1^B, \overline{n_2})$, and $q_3 = (q_2^A, q_1^B, \overline{n_3})$ and $q_4 = (q_2^A, q_2^B, \overline{n_4})$. In other words, we interleave the actions of A and B, as in an expected execution and assume that the two processes were first activated. Let W denote the unary predicate that tells if its argument is an agent name.

- $\overline{n_0} = ()$, $\quad \phi_0 = \langle\rangle$, $\quad \Theta_0 = \emptyset$
- $\overline{n_1} = (K_A, K_B)$, $\quad H_1 = \emptyset$, $\phi_1 = \langle A, B, eK_A, eK_B \rangle$, $\quad \Theta_1 = \emptyset$
- $\overline{n_2} = (K_A, K_B, N_1, R_1)$, $H_2 = \langle h_1 \rangle$, ϕ_2 extends ϕ_1 with $\{\langle N_1, A \rangle\}^{R_1}_{eK_B}$, $\Theta_2 = \{\phi_1 \rhd h_1\}$

- $\overline{n_3} = (K_A, K_B, N_1, R_1, N_2, R_2)$
  $H_3 = \langle h_1, h_2 \rangle$,
  ϕ_3 extends ϕ_2 with
  $\{\langle \pi_1 (dec(h_2, dK_B)), \langle N_2, B \rangle \rangle\}^{R_2}_{eK_{\pi_2(dec(h_2,dK_B))}}$,
  $\Theta_3 = \Theta_2 \cup \{\phi_2 \rhd h_2, W(\pi_2 (dec(h_2, dK_B)))\}$
- $\overline{n_4} = (K_A, K_B, N_1, R_1, N_2, R_2, R_3)$,
  $H_4 = \langle h_1, h_2, h_3 \rangle$,
  ϕ_4 extends ϕ_3 with $\{\pi_1 (\pi_2 (dec(h_3, dK_A)))\}^{R_3}_{eK_B}$,
  $\Theta_4 = \Theta_3 \cup \{\phi_3 \rhd h_3, \pi_1 (dec(h_3, dK_h)) = N_1,$
  $\pi_2 (\pi_2 (dec(h_3, dK_A))) = B\}$,
- $H_5 = \langle h_1, h_2, h_3, h_4 \rangle$,
  $\phi_5 = \phi_4$,
  $\Theta_5 = \Theta_4 \cup \{\phi_4 \rhd h_4, dec(h_4, dK_B) = N_2\}$,

Let $\mathcal{M}$ be a model such that $\pi_2 (dec(h_2, dK_B)) = A$, and $h_2 =_{\mathcal{M}} \{\langle N_1, A \rangle\}^{R_1}_{eK_B}$, and $h_3 =_{\mathcal{M}} \{\langle N_1, \langle N_2, B \rangle \rangle\}^{R_2}_{eK_A}$, and $h_4 =_{\mathcal{M}} \{N\}^{R_3}_{eK_B}$, and $\rhd_{\mathcal{M}}$ is simply the classical Dolev-Yao deduction relation. Then the execution sequence above is valid w.r.t. $\mathcal{M}$, and this corresponds to the correct execution of the NSL protocol between A and B.

EXAMPLE 4.2. Consider again Example 4.1, and a model $\mathcal{M}$ in which $N_0, \{N_1, N_2, B\}^{R_2}_{eK_A} \rhd_{\mathcal{M}} \{N_1, N_0, B\}^{r}_{eK_A}$ for an honestly generated nonce N_0 that can be chosen by the attacker; the transition sequence of the previous example is also valid w.r.t. this model. This however yields an attack, using a malleability property of the encryption scheme. Discarding such attacks requires some properties of the encryption scheme (for instance IND-CCA). It can be ruled out by the axioms that we will introduce. From this example, we see that unexpected attacks can be found when some assumption is not explicitly stated as an axiom to limit adversarial capabilities.

5. COMPUTATIONAL EXECUTION

We now summarize the computational semantics. Short proofs of Theorems 5.2 and 5.3 are in the Section 6 using Fitting's embedding of classical logic into S4 [25].

5.1 Computational Execution

Following Bana and Comon, we consider a family of computational algebras, parametrized by a security parameter η, in which each function symbol is interpreted as a polynomially computable function on bit strings (that may return an error message). Given a sample τ of names, every ground term t can be interpreted as a bit string $[\![t]\!]_\tau$ in such a way that $[\![\_]\!]_\tau$ is a homomorphism of $\mathcal{F}$-algebras (a name n is interpreted as a bit string $\tau(n)$). More generally, if σ is an assignment of the variables of t to bit strings, $[\![t]\!]^\sigma_\tau$ is the (unique) extension of τ (on names) and σ (on variables) as a homomorphism of $\mathcal{F}$-algebras.

Given a set of transition rules, a *computational state* consists of

- a symbolic state s (that is itself a tuple $((q, \overline{n}), \overline{h}, \phi, \Theta)$)
- a sequence of bit strings $\langle b_1, ..., b_m \rangle$ (attacker outputs)
- a sequence $\langle b'_1, ..., b'_m \rangle$ of bit strings (agents' outputs)
- the configuration γ of the attacker.

Given a PPT interactive Turing machine $\mathcal{M}^c$ and a sample τ, a sequence of transitions

$$(s_0, \emptyset, \vec{b'_0}, \gamma_0) \rightarrow ... \rightarrow (s_m, \langle b_1, ..., b_m \rangle, \langle b'_1, ..., b'_m \rangle, \gamma_m)$$

is *(computationally) valid with respect to $\mathcal{M}^c$ and τ* if

- $s_0 \rightarrow \cdots \rightarrow s_m$ is a transition sequence of the protocol
- for all $i = 0, ..., m-1$, $s_i = ((q_i, \overline{n_i}), \overline{h_i}, \phi_i, \Theta_i)$, $\phi_{i+1} = (\phi_i, u_i)$, $[\![u_i]\!]_\tau = b'_{i+1}$
- for every $i = 0, ..., m-1$, there is a configuration γ'_i of the machine $\mathcal{M}^c$ such that $\gamma_i \vdash^*_M \gamma'_i \vdash^*_M \gamma_{i+1}$ and γ'_i is in a sending state, the sending tape containing b_{i+1}, γ_{i+1} is in a receiving state, the receiving tape containing b'_{i+1}
- for all $i = 0, ..., m-1$, the bit strings τ, $\{h_1 \mapsto b_1, ..., h_{i+1} \mapsto b_{i+1}\}$ satisfy all agent checks listed in Θ_{i+1}.

Here $\vdash^*_M$ means what the machine (in whatever model it is defined) can compute via a sequence of computational steps.

5.2 Computational satisfaction of formulas

We recall the computational interpretation of the original predicates, $=$ and $\rhd$ here and the semantics of compound formulas. The difference between our presentation here and that of [7] is that we do not assume any computability condition on non-negligible sets any more, as we apply a trick in the soundness proof that makes it unnecessary. Interpretations of the new predicates are presented in later sections.

Let $(\Omega_0, \Sigma_0, \mathbf{Prob}_0)$ be the probability space of infinite fair coin tosses, Ω_0 being the the set of infinite bit strings, Σ_0 the measurable sets generated by fixing finitely many outcomes, and $\mathbf{Prob}_0$ the probability measure assigning the probabilities to the sets of Σ_0. For a finite bit string $b \in \{0,1\}^*$ of length n, let $\overline{b} \subset \Omega_0$ denote the set of infinite bit strings for which the initial n bits are exactly b. Let Σ_f be the set generated by finite unions intersections, and subtractions of sets of the form $\overline{b}$ (including Ω). Σ_0 is the σ-closure of Σ_f.

Let $\mathcal{M}^c$ be an interactive PPT Turing machine with a special challenge control state $q_{\rm ch}$. We may regard this machine as an attacker, who moves to the state $q_{\rm ch}$ when he thinks that he is ready to break the security property. As usual, the machine takes the security parameter 1^η as an initial input. $\mathcal{M}^c$ interacts with the protocol agents, which are also assumed to be interactive PPT Turing machines, and they respond to the calls of the adversary. Since once η is fixed, such an execution is probabilistic, and for each security parameter η, we denote underlying probability space by $(\Omega^\eta, \Sigma^\eta, \mathbf{Prob}^\eta)$, which is just a copy of the $(\Omega_0, \Sigma_0, \mathbf{Prob}_0)$ above. We denote the elements of Ω^η by ω^η. (Actually, the adversary's random string and the agent random strings are separate, but as there are finitely many of them, they can be thought to be on a single string) Each ω^η is one particular random string. Let $\Omega = (\Omega^\eta)_{\eta \in \mathbb{N}}$. Let $\tau(\omega^\eta)$ be the assignment of all fixed bit string evaluations $\tau(\overline{n})$ of names given for ω^η. For a given n name, we just use simply $n(\omega^\eta)$ for the bit string $\tau(\omega^\eta)(n)$.

By a *non-negligible set of coins* S, we mean $S = (S^\eta)_{\eta \in \mathbb{N}}$, where for all $\eta \in \mathbb{N}$, $S^\eta \in \Sigma^\eta_f$, and $\mathbf{Prob}^\eta\{S^\eta\}$ is non-negligible function of η. We use the notation $S_1 \subseteq S_2$ for $S_1 = (S^\eta_1)_{\eta \in \mathbb{N}}$ and $S_2 = (S^\eta_2)_{\eta \in \mathbb{N}}$ non-negligible sets of coins if for all $\eta \in \mathbb{N}$, $S^\eta_1 \subseteq S^\eta_2$. In what follows, S is any such non-negligible set of coins. The domain of interpretation $\mathfrak{D}(S) = \mathfrak{D}$ is the same for all S: PPT algorithms that take as input η, read from the random tape ω^η, and output a bit string. (As ω^η is infinite coin tosses, the algorithms of course do not read it all, they terminate in polynomial time.)

We recall the interpretations of $=$ and $\rhd$ from [8]: Let σ be a sequence of PPT machines (e.g. one for each free variable x_i of θ): $\mathcal{A}_{x_1}, ..., \mathcal{A}_{x_n} \in \mathfrak{D}$. For example, and $\mathcal{A}_x$ can be the evaluation of any name (in which case $\mathcal{A}_x(\eta, \omega^\eta) = n(\omega^\eta)$, or any value for a handle computed by the adversary, or some more complex object. Let $\sigma(\omega^\eta)$ denote the assignments $x_1 \mapsto \mathcal{A}_{x_1}(\eta, \omega^\eta), ..., x_n \mapsto \mathcal{A}_{x_n}(\eta, \omega^\eta)$. If $st(\eta, \omega^\eta)$ is a statement, then for any fixed $S = (S^\eta)_{\eta \in \mathbb{N}}$, instead of "for all $\eta \in \mathbb{N}$ and all $\omega^\eta \in S^\eta, st(\eta, \omega^\eta)$",

we simply write "for all $\omega \in S, st(\omega)$".

- For the equality predicate, $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} t_1 = t_2$ iff there is a subset $S' \subseteq S$ such that $S \setminus S'$ is negligible, and for all $\omega \in S'$, $[\![t_1]\!]^{\sigma(\omega)}_{\tau(\omega)} = [\![t_2]\!]^{\sigma(\omega)}_{\tau(\omega)}$.
- For the derivability predicate, $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \hat{\phi}, t_1, ..., t_n \rhd t$ if for all non-negligible $S' \subseteq S$, there is a non-negligible $S'' \subseteq S'$ and a PPT Turing machine $\mathcal{A}$ such that for all $\omega \in S''$, $\mathcal{A}([\![\phi_{m(\omega)}]\!]^{\sigma(\omega)}_{\tau(\omega)}, [\![t_1]\!]^{\sigma(\omega)}_{\tau(\omega)}, ..., [\![t_n]\!]^{\sigma(\omega)}_{\tau(\omega)}, a(\omega), r(\omega)) = [\![t]\!]^{\sigma(\omega)}_{\tau(\omega)}$ where $m(\omega)$ is the step at which $\mathcal{M}^c$ reached the challenge state, $a(\omega)$ stands for the protocol adversary's output and $r(\omega)$ is some fresh input from the random tape.
- If P is a constraint, $\vec{t}$ are closed terms then $\mathcal{M}^c, \Pi, S, \sigma \models P(\vec{t})$ iff there is $S' \subseteq S$ such that, $S \setminus S'$ is negligible, and for all $\omega \in S'$, the unique valid computation of Π with respect to $\mathcal{M}^c, \tau(\omega)$ yields a state $(((q, \overline{n}), \overline{h}, \phi, \Theta), \overline{b}, \overline{b'}, \gamma)$ in the control state q_{ch} such that $\overline{n}, \phi \stackrel{s}{\models} P(\vec{t})$.

About the fresh $r(\omega)$, note we assumed for any non-negligible set S that $S^\eta \subseteq \Sigma^\eta_f$ and not $S^\eta \subseteq \Sigma^\eta_0$, so there can always be fresh random bits generated inside S.

Satisfaction of compound formulas are defined the following way.

- $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \theta_1 \wedge \theta_2$ iff $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \theta_1$ and $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \theta_2$.
- $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \theta_1 \vee \theta_2$ iff for any $S' \subseteq S$ non-negligible, there is a $S'' \subseteq S'$ non-negligible such that either $\mathcal{M}^c, \Pi, S'', \sigma \stackrel{c}{\models} \theta_1$ or $\mathcal{M}^c, \Pi, S'', \sigma \stackrel{c}{\models} \theta_2$.
- $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \theta_1 \to \theta_2$ iff for all $S' \subseteq S$ non-negligible, $\mathcal{M}^c, \Pi, S', \sigma \stackrel{c}{\models} \theta_1$ implies $\mathcal{M}^c, \Pi, S', \sigma \stackrel{c}{\models} \theta_2$
- $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \neg\theta$ iff for all $S' \subseteq S$ non-negligible, $\mathcal{M}^c, \Pi, S', \sigma \not\stackrel{c}{\models} \theta$
- $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \exists x.\theta$ iff for any $S' \subseteq S$ non-negligible, there is a $S'' \subseteq S'$ non-negligible and a PT machine $\mathcal{A}_x$ such that $\mathcal{M}^c, \Pi, S'', \sigma, \mathcal{A}_x \stackrel{c}{\models} \theta$
- $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \forall x.\theta$ iff for any probabilistic polynomial time machine $\mathcal{A}_x$, $\mathcal{M}^c, \Pi, S, \sigma, \mathcal{A}_x \stackrel{c}{\models} \theta$
- If x is a constrained variable, the interpretation of $\exists x.\theta$ is analogous to the symbolic case: $\mathcal{M}, \Pi, S, \sigma \stackrel{c}{\models} \exists x.\theta$ if and only if for every non-negligible $S' \subseteq S$ there is a non-negligible $S'' \subseteq S'$ and a ground term t, such that the satisfaction $\mathcal{M}, \Pi, S'', \sigma \stackrel{c}{\models} \theta\{x \mapsto t\}$ holds.
- If x is a constrained variable, the interpretation of $\forall x.\theta$ is analogous to the symbolic case: $\mathcal{M}, \Pi, S, \sigma \stackrel{c}{\models} \forall x.\theta$ if and only if for every ground term t, the satisfaction $\mathcal{M}, \Pi, S, \sigma \stackrel{c}{\models} \theta\{x \mapsto t\}$ holds.

$\mathcal{M}^c, \Pi \stackrel{c}{\models} \theta$ iff $\mathcal{M}^c, \Pi, \Omega \stackrel{c}{\models} \theta$ and $\Pi \stackrel{c}{\models} \theta$ if $\mathcal{M}^c, \Pi \stackrel{c}{\models} \theta$ for every $\mathcal{M}^c$ and q_{ch}.

Given a protocol Π, we say that there is a *successful computational attack* against the security property θ (a first-order formula) of the protocol if there is a an attacker $\mathcal{M}^c$ and a non-negligible set of coins S such that $\mathcal{M}^c, \Pi, S \stackrel{c}{\models} \neg\theta$ (which is the same as $\Pi \not\stackrel{c}{\models} \theta$).

Despite that semantics of the compound formulas is not as usual in first-order logic, we prove in Section 6 that as a consequence of Fitting's embedding [25] of classical logic into S4, the following theorems hold.

THEOREM 5.1 (FITTING'S EMBEDDING). *With the above semantics, first-order deduction rules are sound.*

THEOREM 5.2 (TRACE MAPPING). *Let Π be a protocol, $s_1 \to ... \to s_m$ be a symbolic transition sequence of Π and $\mathcal{M}^c$ be a probabilistic polynomial time interactive Turing machine. If there is a non-negligible set of coins S such that, for any $\omega \in S$, there is a sequence of transitions $(s_0, \vec{b_0}, \vec{b'_0}, \gamma_0) \to \cdots \to (s_m, \vec{b_m}, \vec{b'_m}, \gamma_m)$ that is computationally valid w.r.t. $\mathcal{M}^c$, $\tau(\omega)$ and γ_m is in the challenge state q_{ch}, then for any set of FOL formulas Φ, $\mathcal{M}^c, \Pi, S \stackrel{c}{\models} \Phi$ implies there is a symbolic model $\mathcal{M}$ such that $s_0 \to \cdots \to s_m$ is a valid symbolic execution w.r.t. $\mathcal{M}$ and $\mathcal{M}, s_m \stackrel{s}{\models} \Phi$.*

THEOREM 5.3 (GENERAL SOUNDNESS). *If there is a successful computational attack such that the number of sessions of honest agents are bounded in the security parameter, then there is also a successful symbolic attack.*

5.3 Axioms for Equality and Derivability

We recall the core axioms presented in [6]. As usual, unquantified variables are universally quantified. Unless noted otherwise, they are always sound.

Equality is a Congruence:
- $x = x$, and the substitutability (congruence) property of equal terms holds for $=$ and $\rhd$.

Core Axioms for the Derivability Predicate:
- Self derivability: $\hat{\phi}, \vec{x}, x \rhd x$
- Increasing capabilities: $\hat{\phi}, \vec{x} \rhd y \longrightarrow \hat{\phi}, \vec{x}, x \rhd y$
- Commutativity: If $\vec{x'}$ is a permutation of $\vec{x}$, then $\hat{\phi}, \vec{x} \rhd y \longrightarrow \hat{\phi}, \vec{x'} \rhd y$
- Transitivity of derivability: $\hat{\phi}, \vec{x} \rhd \vec{y} \wedge \hat{\phi}, \vec{x}, \vec{y} \rhd \vec{z} \longrightarrow \hat{\phi}, \vec{x} \rhd \vec{z}$
- Functions are derivable: $\hat{\phi}, \vec{x} \rhd f(\vec{x})$ This axiom is sound as long as functions are interpreted as PT computable algorithms.

Axioms for Freshly Generated Items:
- No telepathy: $\mathsf{fresh}(x; \hat{\phi}) \longrightarrow \hat{\phi} \not\rhd x$ This axiom is sound as long as $\mathsf{RanGen}()$ items are generated so that they can only be guessed with negligible probability. A more general version is also possible as $\mathsf{fresh}(x; \hat{\phi}, \vec{x}) \wedge \vec{x} \preccurlyeq \hat{\phi} \longrightarrow \hat{\phi}, \vec{x} \not\rhd x$
- Fresh items do not help to compute: $\mathsf{fresh}(x; \hat{\phi}, \vec{x}, y) \wedge \vec{x}, y \preccurlyeq \hat{\phi} \wedge \hat{\phi}, \vec{x}, x \rhd y \longrightarrow \hat{\phi}, \vec{x} \rhd y$

Equations for the fixed function symbols: For example, for symmetric encryption $sdec(\{\!|x|\!\}^{\vec{R}}_K, K) = x$, and for pairing, $\pi_1(\langle x, y\rangle) = x$; $\pi_2(\langle x, y\rangle) = y$. Function of error is error $f(..., \bot, ...) = \bot$, etc.

6. THE FITTING CONNECTION

The trace mapping and the general soundness theorems for arbitrary first-order formulas were proven directly from their definitions by an elaborate argument in [7]. We have realized however, that they are rather easy consequences of Fitting's embedding of first-order logic into first-order S4 [25]. The non-Tarskian computational semantics of first-order formulas that naturally arouse in the BC technique turns out to be a special kind of Kripke semantics of first-order S4 composed with Fitting's embedding of FOL into first-order S4. We detail this connection here, and show how trace mapping and general soundness follow from Fitting's theorem. This section assumes basic familiarity with S4 modal logic and its first-order extension as well as Kripke semantics.

For any first-order formula θ, consider the Fitting transformation $\theta \mapsto \theta^*$, where θ^* is a formula of first-order S4, and is defined recursively as follows:

- For any atomic formula θ, let $\theta^* \equiv \Box\Diamond\theta$.
- $(\neg\theta)^* \equiv \Box\neg\theta^*$
- $(\theta_1 \rightarrow \theta_2)^* \equiv \Box(\theta_1^* \rightarrow \theta_2^*)$
- $(\theta_1 \wedge \theta_2)^* \equiv (\theta_1^* \wedge \theta_2^*)$
- $(\theta_1 \vee \theta_2)^* \equiv \Box\Diamond(\theta_1^* \vee \theta_2^*)$
- $(\forall x\theta)^* \equiv \forall x\theta^*$
- $(\exists x\theta)^* \equiv \Box\Diamond\exists x\theta^*$

Fitting in [25] put $\Box\Diamond$ everywhere and noted that it is redundant in front of the conjunction. It is also easy to check that if the Barcan formula and its converse ($\forall x\Box\theta \leftrightarrow \Box\forall x\theta$) are assumed (that is, when the domain does not change from possible world to possible world in the Kripke structure), then $\Box\Diamond$ is also redundant in front of the universal quantification (as $\theta^* \leftrightarrow \Box\Diamond\theta^*$ holds in our definitions for all θ). In our computational situation the domain is unchanged as we show below. So in this paper we assume the Barcan formula and its converse.

Fitting's theorem says that any formula θ is derivable in first-order logic if and only if θ^* it is derivable in S4 with the Barcan formulas. (Without the Barcan formulas, $(\forall x\theta)^* \equiv \Box\Diamond\forall x\theta^*$ has to be written above).

Observe that if we think of non-negligible sets as possible worlds, and the subset relation as accessibility (that is, if S' is accessible from S iff $S' \subseteq S$), then we can define a computational Kripke semantics: For our predicates, consider the S4 satisfaction relation $\mathcal{M}^c, \Pi, S, \sigma \models^{s4}$ that we define almost the same way as the BC computational satisfaction $\models^c$ of Section 5.2 is defined, except that we drop the "for all non-negligible $S' \subseteq S$, there is a non-negligible $S'' \subseteq S'$" phrase, and replace S'' with S in the remaining of the definition. For example, the satisfaction of derivability becomes:

- For the derivability predicate, $\mathcal{M}^c, \Pi, S, \sigma \models^{s4} \hat{\phi}, t_1, ..., t_n \rhd t$ if there is a PPT Turing machine $\mathcal{A}$ such that for all $\omega \in S$, $\mathcal{A}([\![\phi_{m(\omega)}]\!]^{\sigma(\omega)}_{\tau(\omega)}, [\![t_1]\!]^{\sigma(\omega)}_{\tau(\omega)}, ..., [\![t_n]\!]^{\sigma(\omega)}_{\tau(\omega)}, a(\omega), r(\omega)) = [\![t]\!]^{\sigma(\omega)}_{\tau(\omega)}$ where $m(\omega)$ is the step at which $\mathcal{M}^c$ reached the challenge state, $a(\omega)$ stands for the protocol adversary's output and $r(\omega)$ is some fresh input from the random string.

For an arbitrary θ first-order S4 formula, $\mathcal{M}^c, \Pi, S, \sigma \models^{s4} \Box\theta$ is defined to hold if and only if $\mathcal{M}^c, \Pi, S', \sigma \models^{s4} \theta$ holds for all non-negligible $S' \subseteq S$, and $\mathcal{M}^c, \Pi, S, \sigma \models^{s4} \Diamond\theta$ is defined to hold if and only if $\mathcal{M}^c, \Pi, S', \sigma \models^{s4} \theta$ holds for some non-negligible $S' \subseteq S$. Taking θ to be $\hat{\phi}, t_1, ..., t_n \rhd t$, and applying $\Box\Diamond$ to the above definition of S4 satisfaction, we receive the computational satisfaction of Bana and Comon-Lundh. That is, $\mathcal{M}^c, \Pi, S, \sigma \models^c \hat{\phi}, t_1, ..., t_n \rhd t$ if and only if $\mathcal{M}^c, \Pi, S, \sigma \models^{s4} \Box\Diamond\hat{\phi}, t_1, ..., t_n \rhd t$.

Note also that for the equality predicate and for the constraints, "for all non-negligible $S' \subseteq S$, there is a non-negligible $S'' \subseteq S'$" can be freely inserted in the definition, as the resulting definition is equivalent with the original: for example, if for all non-negligible $S' \subseteq S$, there is a non-negligible $S'' \subseteq S'$ such that $[\![t_1]\!]^{\sigma(\omega)}_{\tau(\omega)} = [\![t_2]\!]^{\sigma(\omega)}_{\tau(\omega)}$ holds on S'', then it also holds up to negligible probability on S, because if there were a non-negligible subset $S' \subseteq S$ on which $[\![t_1]\!]^{\sigma(\omega)}_{\tau(\omega)} \neq [\![t_2]\!]^{\sigma(\omega)}_{\tau(\omega)}$ were true, then this S' would not have some non-negligible subset S'' on which they are equal, a contradiction. So BC could have defined satisfaction of equality as

- For the equality predicate, $\mathcal{M}^c, \Pi, S, \sigma \models^c t_1 = t_2$ iff for all non-negligible $S' \subseteq S$, there is a non-negligible subset $S'' \subseteq S'$ such that for all $\omega \in S''$, $[\![t_1]\!]^{\sigma(\omega)}_{\tau(\omega)} = [\![t_2]\!]^{\sigma(\omega)}_{\tau(\omega)}$.

That is, $\mathcal{M}^c, \Pi, S, \sigma \models^c t_1 = t_2$ iff $\mathcal{M}^c, \Pi, S, \sigma \models^{s4} \Box\Diamond t_1 = t_2$. The same is true for constraints. Hence we have this for all atomic formulas. $\models^{s4}$ for compound formulas is defined as usual in Kripke semantics. For example, $\mathcal{M}^c, \Pi, S, \sigma \models^{s4} \theta_1 \vee \theta_2$ if and only if $\mathcal{M}^c, \Pi, S, \sigma \models^{s4} \theta_1$ or $\mathcal{M}^c, \Pi, S, \sigma \models^{s4} \theta_2$.

Comparing the definition of $\models^c$ in Section 5.2 for compound formulas with Fitting's embedding, for any first-order formula θ,

$$\mathcal{M}^c, \Pi, S, \sigma \models^c \theta \iff \mathcal{M}^c, \Pi, S, \sigma \models^{s4} \theta^*.$$

For a set of first-order formulas Φ, let Φ^* mean the set that we get by applying the Fitting transformation to all formulas in Φ. Since with $\models^{s4}$, our computational semantics is a special kind of Kripke semantics, Fitting's theorem implies that if $\mathcal{M}^c, \Pi, S, \sigma \models^{s4} \Phi^*$ holds and if $\Phi \vdash^{FOL} \theta$, then $\mathcal{M}^c, \Pi, S, \sigma \models^{s4} \theta^*$. So we also have that if $\mathcal{M}^c, \Pi, S, \sigma \models^c \Phi$ holds and if $\Phi \vdash^{FOL} \theta$, then $\mathcal{M}^c, \Pi, S, \sigma \models^c \theta$. This is exactly Theorem 5.1.

The Barcan formula and its converse hold, as the domain $\mathfrak{D}$ does not depend on the non-negligible sets.

For Theorem 5.2, note that it is assumed in the theorem that for all $\omega \in S$ the computational execution has the same length m and the symbolic part of their traces, s_i agree. Remember that the symbolic states have the transition conditions Θ_i in them. So at the challenge state, we have $\mathcal{M}^c, \Pi, S \models^c \Theta_m \wedge \Phi$. Note that since on the traces, s_i agree, the terms that are in the frame also agree for all trace $\omega \in S$. Therefore, $\hat{\phi}$ in the formulas of θ_m and Φ can be replaced by the list of terms in the frames. Let θ'_m and Φ' denote the formulas we receive this way. Then, $\mathcal{M}^c, \Pi, S \models^c \Theta'_m \wedge \Phi'$. This is the same as $\mathcal{M}^c, \Pi, S \models^{s4} (\Theta'_m \wedge \Phi')^*$, that, by Fitting's theorem means that $\Theta'_m \wedge \Phi'$ is first-order satisfiable (because $(\Theta'_m \wedge \Phi')^*$ is S4 satisfiable). Hence there is a symbolic model $\mathcal{M}$ with $\mathcal{M} \models^s \Theta'_m \wedge \Phi'$. As Θ'_m and Φ' have no frames in them, it is easy to see from the symbolic satisfaction that $\mathcal{M}, s_m \models^s \Theta'_m \wedge \Phi'$ is also satisfied as satisfaction does not depend on the state. Finally, as in s_m, the frames contain exactly the terms with which we replaced $\hat{\phi}$, we can now write them back and receive $\mathcal{M}, s_m \models^s \Theta_m \wedge \Phi$. Which also means that Θ_m and Φ are first-order consistent.

Finally, for proving Theorem 5.3, consider the following. If there is a computational attack, that is, if the negation of the security formula θ_s is computationally satisfied by some $\mathcal{M}^c, \Pi, S$, then, as long as only bounded number (in the security parameter) of sessions are allowed, the maximum number of different (with respect to ω) symbolic transitions $s_1 \rightarrow \cdots \rightarrow s_m$ does not depend on the security parameter. Therefore, S can be split (up to negligible probability) into a fixed number of non-negligible subsets on each of which $s_1 \rightarrow \cdots \rightarrow s_m$ is independent of ω. Pick one, call it S'. Then $\mathcal{M}^c, \Pi, S'$ computationally satisfies $\neg\theta_s$ as well as all computationally sound axioms (denote the set by Φ_A), and Θ_m also: $\mathcal{M}^c, \Pi, S' \models^c \Theta_m \wedge \Phi_A \wedge \theta_s$. Hence, By Theorem 5.2 there is a symbolic model $\mathcal{M}$ such that $\mathcal{M}, s_m \models^s \Theta_m \wedge \Phi_A \wedge \theta_s$, which exactly means that there is a symbolic attack.

Note, although by Fitting's theorem, the first-order deduction rules are computationally sound, it is not possible in the current formulation to deduce security properties from the axioms only by first-order deduction rules and nothing else. The BC technique has no formulas expressing the transition system, no formulas saying the agents follow the protocol roles. Explicit time and some axiom for induction would also be needed as in [9]. These are taken care by the symbolic execution.

7. DERIVABILITY WITH ORACLES

Syntax of various types of derivability with oracles was introduced in Section 3. Here we define their computational semantics and list a number of axioms that are computational sound.

7.1 Computational Semantics of Derivability with Oracles

Let $\mathcal{O}^{\text{sic2}}$ be the following oracle: It first takes a list of honestly generated keys $\mathcal{K}$ and some additional list $\mathcal{C}$ of ciphertexts. The oracle can be called for encryption by submitting a string to encrypt along with the encrypting key's place number in $\mathcal{K}$. The oracle honestly generates the desired encryption, returns the result and adds it to the list $\mathcal{C}$. The oracle can also be called for a decryption again with specifying the key and providing a ciphertext. If the ciphertext is not one of those in $\mathcal{C}$, the oracle outputs the decryption. Let $\mathcal{O}^{\text{skc2}}$ take $\mathcal{K}$ and $\mathcal{C}$ as above, but also $\tau(\mathcal{L})$ assignment of bit strings to a finite set of name symbols $\mathcal{L}$. The oracle accepts descriptions of functions of the names $\mathcal{L}$, the ciphers $\mathcal{C}$ and the keys $\mathcal{K}$, into which he substitutes the corresponding bit strings, computes the function and then encrypts the result with the specified key and adds it to $\mathcal{C}$. Decryption works as for $\mathcal{O}^{\text{sic2}}$. Let $\mathcal{O}^{\text{aic2}}$ and $\mathcal{O}^{\text{akc2}}$ be the analogous notions for public key encryption. PPT algorithms with oracle access will be written as $\mathcal{A}^{\mathcal{O}}$, $\mathcal{B}^{\mathcal{O}^{\mathfrak{O}}}$.

The meaning of the definition is that $\mathcal{M}^c, \Pi, S, \sigma \models^{\text{s4}} \hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} x$ holds if there is a PPT algorithm $\mathcal{A}^{\mathcal{O}}$ that, for $\omega \in S$, receiving the bit strings $[\![\hat{\phi}, \vec{x}]\!]^{\sigma(\omega)}_{\tau(\omega)}$, it outputs the bit string $[\![x]\!]^{\sigma(\omega)}_{\tau(\omega)}$. In the computation $\mathcal{A}^{\mathcal{O}}$ can request encryption and decryption oracles corresponding to the honest keys, but it will only receive a decryption if the submitted bit string is not a bit string corresponding to an encryption in $\hat{\phi}, \vec{x}$ or a bit string received from the encryption oracle. Outside S, nothing is required.

DEFINITION 7.1. *Semantics of Derivability with Oracles: Let $\mathcal{M}^c, \Pi, S, \sigma$ be as before. Let $a(\omega^\eta)$ denote the protocol adversary output as it reaches the challenge state on the random input ω^η, and let $m(\omega^\eta)$ denote the number of moves till then. We write $\mathcal{M}^c, \Pi, S, \sigma \models^{\text{s4}} \hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} x$ iff there is a PPT Turing machine $\mathcal{A}^{\mathcal{O}}$ such that for all $\omega \in S$,*

$$\mathcal{A}^{\mathcal{O}}\big([\![\phi_{m(\omega)}]\!]^{\sigma(\omega)}_{\tau(\omega)}, [\![\vec{x}]\!]^{\sigma(\omega)}_{\tau(\omega)}, a(\omega), r(\omega)\big) = [\![x]\!]^{\sigma(\omega)}_{\tau(\omega)}$$

where $r(\omega)$ is some fresh (not used for the computation of $\hat{\phi}, \vec{x}, x$) random input from the random string. On each ω, if the tuple $(((q, \overline{n}), \overline{h}, \phi, \Theta), \overline{b}, \overline{b'}, \gamma)$ denotes the state yielded by the unique valid computation of Π with respect to $\mathcal{M}^c$ and $\tau(\omega)$, then the oracles receive in $\mathcal{K}$ all keys (bit strings) corresponding to the keys in $\overline{n}$, and in $\mathcal{C}$ all strings of the form $[\![\{z\}^R_{eK}]\!]^{\sigma(\omega)}_{\tau(\omega)}$ with R and K names in $\overline{n}$, and $\overline{n}, \phi \models^{\text{s}} \{z\}^R_{eK} \sqsubseteq \hat{\phi}, \vec{x}$. In $\tau(\mathcal{L})$, the KDM oracles receive all assignments of names in $\bar{n}$ to bit strings, except (in the symmetric case), for those that occur as random inputs R to the encryptions in $\mathcal{C}$.

We shorten this as

$$\mathcal{M}^c, \Pi, S, \sigma \models \mathcal{A}^{\mathcal{O}^{\mathfrak{O}}}(\hat{\phi}, \vec{x}) = x,$$

implicitly assuming the algorithm has access to the protocol adversary's knowledge and to random bits. Let

$$\mathcal{M}^c, \Pi, S, \sigma \models^{c} \hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} x \quad \textit{iff} \quad \mathcal{M}^c, \Pi, S, \sigma \models^{\text{s4}} \Box\Diamond(\hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} x).$$

Note that in the KDM case, the submitted functions may depend on randomly generated items that differ from the secret keys but are not accessible to the protocol adversary. For example, it is allowed to depend on a secret nonce. This is necessary for receiving nice axioms for the KDM case, and we explain the reason at the axioms. Still, assuming the usual KDM security is enough to prove that an unsent key is uncompromised with such oracle access. The reason is that when a KDM attack is constructed from the protocol attack, the KDM attacker has access to the items generated by the honest agents except for the secret keys and for the random inputs to the encryptions.

7.2 Axioms for Derivability with Oracles

The following axioms (except for the second and last entry of the core axioms for derivability predicates) are very similar to the ones in Section 5.3, and are just as trivial. The second entry of the core axioms for derivability predicates with oracles is also trivially computationally sound.

Core Axioms for the Derivability Predicate with Oracles.

- Let $\mathsf{SameEnc}(\vec{x}; \vec{y})$ be the constraint that there is a one-to-one correspondence between the honest encryption terms of $\vec{x}$ and $\vec{y}$ such that the corresponding encryption terms are equal (with respect to the equality predicate). Then $\mathsf{SameEnc}(\vec{x}; \vec{y}) \wedge \vec{x}, x = \vec{y}, y \longrightarrow (\hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} x \leftrightarrow \hat{\phi}, \vec{y} \rhd^{\mathfrak{O}} y)$.
- More oracles help more: If the oracles of $\mathfrak{O}$ are more powerful than the oracles of $\mathfrak{O}'$, then $\hat{\phi}, \vec{x} \rhd^{\mathfrak{O}'} x \longrightarrow \hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} x$. In particular, $\hat{\phi}, \vec{x} \rhd x \longrightarrow \hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} x$ and $\hat{\phi}, \vec{x} \rhd^{\text{aic2}} x \to \hat{\phi}, \vec{x} \rhd^{\text{akc2}} x$ and $\hat{\phi}, \vec{x} \rhd^{\text{sic2}} x \to \hat{\phi}, \vec{x} \rhd^{\text{skc2}} x$
- Increasing capabilities: $\hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} y \longrightarrow \hat{\phi}, \vec{x}, x \rhd^{\mathfrak{O}} y$
- Commutativity: If $\vec{x}'$ is a permutation of $\vec{x}$, then $\hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} y \longrightarrow \hat{\phi}, \vec{x}' \rhd^{\mathfrak{O}} y$
- Transitivity: $\hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} \vec{y} \wedge \hat{\phi}, \vec{x}, \vec{y} \rhd^{\mathfrak{O}} \vec{z} \longrightarrow \hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} \vec{z}$
- Decryption Oracles help:

$$\begin{aligned} &\mathsf{RanGen}(K) \wedge \hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} y \\ &\wedge \forall x R(y = \{x\}^R_{eK} \to \{x\}^R_{eK} \not\sqsubseteq \hat{\phi}, \vec{x}) \\ &\longrightarrow \hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} \mathit{dec}(y, dK). \end{aligned}$$

  This expresses that if y is computable and is not an encryption in $\hat{\phi}, \vec{x}$, then $\mathit{dec}(y, dK)$ is also computable from the same items as the decryption oracle can be called. We do not have to require that y is none of the encryptions done by the oracles, because if they were, then the decryption is known to the submitter. Again, this follows purely from the definition of $\rhd^{\mathfrak{O}}$, CCA2 security of the encryption is not required. This axiom together with the transitivity axiom easily imply

$$\begin{aligned} &\mathsf{RanGen}(K) \wedge \hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} y \wedge \hat{\phi}, \vec{x}, \mathit{dec}(y, dK) \rhd^{\mathfrak{O}} z \\ &\wedge \forall x R(y = \{x\}^R_{eK} \to \{x\}^R_{eK} \not\sqsubseteq \hat{\phi}, \vec{x}) \longrightarrow \hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} z. \end{aligned}$$

  This replaces the non-malleability axiom of [6] for the derivability predicate. With tiny modifications, it is possible to rewrite the NSL proof presented in [6] for using the $\rhd^{\mathfrak{O}}$predicate and this simpler axiom instead of the $\rhd$ predicate with the non-malleability axiom there.

Axioms for Freshly Generated Items. These axioms are sound for the same reason as the corresponding ones for $\rhd$ were:

- No telepathy: $\mathsf{fresh}(x; \hat{\phi}, \vec{x}) \wedge \vec{x} \preccurlyeq \hat{\phi} \longrightarrow \hat{\phi}, \vec{x} \not\rhd^{\mathfrak{O}} x$ (implies no-telepathy axiom without oracles). This is sound as long as $\mathsf{RanGen}()$ means generation with negligible guessing probability only.
- Fresh items do not help to compute: $\mathsf{fresh}(x; \hat{\phi}, \vec{x}, y) \wedge \vec{x}, y \preccurlyeq \hat{\phi} \wedge \hat{\phi}, \vec{x}, x \rhd^{\mathfrak{O}} y \longrightarrow \hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} y$

$\hat{\phi}, \vec{x}, x \rhd^{\mathfrak{O}} x$ is implied by the more oracles help more axiom and the self-derivability axiom of derivability predicate. Also, $\hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} f(\vec{x})$.

8. KEY USABILITY

Syntax of various types of key usability was introduced in Section 3. Here we define their computational semantics and list a number of axioms that are computational sound.

8.1 IND-CCA2 and KDM-CCA2 cases

8.1.1 Semantics of IND-CCA2 Key Compromise

The idea of key usability is that a key has been uncompromised, that is, it can be used for safe encryption. To match the computability predicate, we define the negation of it, that is, key compromise. The intuitive meaning of $\hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$ is that $\hat{\phi}, \vec{x}$ compromises the key (with oracles) and it cannot be used for safe encryption any more.

The first thought here would be to define the compromise so that from $\hat{\phi}, \vec{x}$, an x can be computed such that the encryption of x and of $0^{|x|}$ are *computationally distinguishable*. However, the major difficulty here (and the major difference from [24]) is that we have to define our notion for any S. Computational distinguishability on an arbitrary set has no meaning: even in the usual CCA2 game with CCA2 secure encryption, there can be non-negligible sets of coins defined on which the CCA2 attacker returns 1 if the real bit string is encrypted, while 1 with probability $1/2$ if the 0's are encrypted; it is very easy to find sets like this.

What we came up with is a notion of observational inequivalence: encryptions of x and $0^{|x|}$ have to be observationally inequivalent on S, where PPT algorithms with oracle access provide the contexts and equality on S provides the equivalence.

DEFINITION 8.1 (KEY COMPROMISE). *The define the relation $\mathcal{M}^c, \Pi, S, \sigma \stackrel{s4}{\models} \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$ to hold if either $\mathcal{M}^c, \Pi, S, \sigma \stackrel{s4}{\models} \vec{x} \not\preccurlyeq \hat{\phi} \vee \neg \mathsf{RanGen}(K)$, or there are R, PPT algorithms $\mathcal{A}_{21}^{\mathcal{O}}$, $\mathcal{A}_{22}^{\mathcal{O}}$, and $\mathcal{A}_1^{\mathcal{O}}$ (in the IND case) or x (in the KDM case) such that:*

In the the IND case

- *R is generated honestly, statistically independently of the interpretations of $\hat{\phi}$, $\vec{x}$, $\mathcal{A}_1^{\mathcal{O}^{\mathfrak{D}}}(\hat{\phi}, \vec{x})$ and*
- *Either $\mathcal{M}^c, \Pi, S, \sigma \models$*

  $$\mathcal{A}_{21}^{\mathcal{O}}(\varphi, \vec{x}, \{\mathcal{A}_1^{\mathcal{O}}(\varphi, \vec{x})\}_{eK}^{R}) = \mathcal{A}_{22}^{\mathcal{O}}(\varphi, \vec{x}, \{\mathcal{A}_1^{\mathcal{O}}(\varphi, \vec{x})\}_{eK}^{R})$$

  and for some (hence for all) R' fresh random input generated inside S, $\mathcal{M}^c, \Pi, S, \sigma \models$

  $$\mathcal{A}_{21}^{\mathcal{O}}(\varphi, \vec{x}, \{0^{|\mathcal{A}_1^{\mathcal{O}}(\varphi, \vec{x})|}\}_{eK}^{R'}) \neq \mathcal{A}_{22}^{\mathcal{O}}(\varphi, \vec{x}, \{0^{|\mathcal{A}_1^{\mathcal{O}}(\varphi, \vec{x})|}\}_{eK}^{R'})$$

- *Or $\mathcal{M}^c, \Pi, S, \sigma \models$*

  $$\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{\mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, \vec{x})\}_{eK}^{R}) \neq \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{\mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, \vec{x})\}_{eK}^{R})$$

  and for some (hence for all) R' fresh random input generated inside S, $\mathcal{M}^c, \Pi, S, \sigma \models$

  $$\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{0^{|\mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, \vec{x})|}\}_{eK}^{R'}) = \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{0^{|\mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, \vec{x})|}\}_{eK}^{R'})$$

In the KDM case

- *R is generated honestly, statistically independently of the interpretations of $\hat{\phi}$, $\vec{x}$, x, and $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} x \preccurlyeq \hat{\phi}, \vec{x}$ and*
- *Either $\mathcal{M}^c, \Pi, S, \sigma \models$*

  $$\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{x\}_{eK}^{R}) = \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{x\}_{eK}^{R})$$

  and for some (hence for all) R' fresh random input generated inside S, $\mathcal{M}^c, \Pi, S, \sigma \models$

  $$\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{0^{|x|}\}_{eK}^{R'}) \neq \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{0^{|x|}\}_{eK}^{R'})$$

- *Or $\mathcal{M}^c, \Pi, S, \sigma \models$*

  $$\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{x\}_{eK}^{R}) \neq \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{x\}_{eK}^{R})$$

  and for some (hence for all) R' fresh random input generated inside S, $\mathcal{M}^c, \Pi, S, \sigma \models$

  $$\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{0^{|x|}\}_{eK}^{R'}) = \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{0^{|x|}\}_{eK}^{R'})$$

Let $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} x$ iff $\mathcal{M}^c, \Pi, S, \sigma \stackrel{s4}{\models} \Box\Diamond(\hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} x)$.

Recall from Section 6 that $\Box\Diamond$ means "there is a non-negligible $S' \subseteq S$ such that for all S''..." Also note that as we required our non-negligible sets to be in Σ_f, it is always possible to toss fresh coins inside the non-negligible sets.

Note that while R does not have to be generated inside S, R' does. In particular, S may actually depend on R, which is essential for the usability of the axioms, because the non-negligible sets on which we need to apply the axioms may depend on values of an encryption, and hence values of R. On the other hand, S is not allowed to depend on R', which is essential for proving that a freshly generated key is not compromised.

Note, in the KDM case, x does not have to be computed from $\hat{\phi}, \vec{x}$, it could be a secret nonce. This corresponds to the fact that in the semantics of $\rhd^{\mathfrak{D}}$ in the KDM case (as we noted after the definition) we allowed the functions submitted to the oracles to depend on such items not known to the protocol adversary.

8.1.2 Axioms for CCA2 Key Compromise

We now present the axioms for key compromise. First the core axioms for which soundness does not need CCA2 security.

Core Axioms for the Key Compromise Predicate.

- Let $\mathsf{SameEnc}(\vec{x}; \vec{y})$ be the constraint as before. Then $\mathsf{SameEnc}(\vec{x}; \vec{y}) \wedge \vec{x}, x = \vec{y}, y \longrightarrow (\hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} x \leftrightarrow \hat{\phi}, \vec{y} \blacktriangleright^{\mathfrak{D}} y)$.
- Derivability implies compromise: $\hat{\phi}, \vec{x} \rhd^{\mathfrak{D}} K \longrightarrow \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$ If K is computable for the adversary, then it is compromised. Note, this axiom and the self derivability axiom (from 7.2) imply that $\hat{\phi}, \vec{x}, K \blacktriangleright^{\mathfrak{D}} K$
- Increasing capabilities for key compromise: $\hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K \longrightarrow \hat{\phi}, \vec{x}, x \blacktriangleright^{\mathfrak{D}} K$
- Commutativity: If $\vec{x}'$ is a permutation of $\vec{x}$, then $\hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K \longrightarrow \hat{\phi}, \vec{x}' \blacktriangleright^{\mathfrak{D}} K$
- Transitivity: $\hat{\phi}, \vec{x} \rhd^{\mathfrak{D}} \vec{y} \wedge \hat{\phi}, \vec{x}, \vec{y} \blacktriangleright^{\mathfrak{D}} K \longrightarrow \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$ The intuitive reason is very clear: $\vec{y}$ just contains extra information, that can be computed from $\hat{\phi}, \vec{x}$, so it is not actually needed in the compromise. This, and the functions are derivable axiom imply $\hat{\phi}, \vec{x}, f(\vec{x}) \blacktriangleright^{\mathfrak{D}} K \longrightarrow \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$. With the increasing capabilities axiom, we get $\hat{\phi}, f(\vec{x}) \blacktriangleright^{\mathfrak{D}} K \longrightarrow \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$. We refer to these as *function application*.
- Uncompromised keys securely encrypt:
  - If $\mathfrak{D}$ is either aic2 or sic2, then

    $$\begin{aligned} &\mathsf{RanGen}(K) \wedge \mathsf{fresh}(R; \hat{\phi}, \vec{x}, x, y, K) \\ &\wedge \vec{x}, x, y \preccurlyeq \hat{\phi} \wedge \hat{\phi}, \vec{x}, \{x\}_{eK}^{R} \rhd^{\mathfrak{D}} y \\ &\longrightarrow \hat{\phi}, \vec{x}, x \blacktriangleright^{\mathfrak{D}} K \vee \hat{\phi}, \vec{x} \rhd^{\mathfrak{D}} y \end{aligned}$$

    This formula means that if the key is uncompromised, that is, $\hat{\phi}, \vec{x}, x \not\blacktriangleright^{\mathfrak{D}} K$, then $\{x\}_{eK}^{R}$ cannot help in deriving y. In other words, if it is possible to derive y with $\{x\}_{eK}^{R}$, then it is also possible to derive it without $\{x\}_{eK}^{R}$. The freshness and random generation conditions ensure that $\{x\}_{eK}^{R}$ is indeed a good encryption

(e.g. $\{N\}^N_{eK}$ or $\{N\}^{eK}_{eK}$ are not good), and also that y cannot depend on $\{x\}^R_{eK}$ (e.g. $y = \{x\}^R_{eK}$ is not good). Moreover, $\vec{x}, x, y \preccurlyeq \hat{\phi}$ ensures that handles in these terms are given values the adversary can compute (otherwise e.g. taken $x = h$, the handle h cannot be dK if dK was never sent, and it cannot be R either).
This formula is completely analogous to the secrecy axiom in [6] but $dK \sqsubseteq \hat{\phi}, \vec{x}, x$ there is replaced now with $\hat{\phi}, \vec{x}, x \blacktriangleright^{\text{aic2}} K$ as we can now allow dK to appear inside a secure encryption for example.

– If $\mathfrak{O}$ is either akc2 or skc2, then

$$\begin{aligned}&\mathsf{RanGen}(K) \wedge \mathsf{fresh}(R; \hat{\phi}, \vec{x}, x, y, K)\\ &\wedge \vec{x}, x, y \preccurlyeq \hat{\phi} \wedge \hat{\phi}, \vec{x}, \{x\}^R_{eK} \rhd^{\mathfrak{O}} y\\ &\longrightarrow \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{O}} K \vee \hat{\phi}, \vec{x} \rhd^{\mathfrak{O}} y\end{aligned}$$

The difference here from the axiom for IND-CCA2 security is that in $\hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{O}} K$ now there is no x. This corresponds to the fact that the encrypted message x is allowed to contain the decryption key, or it may leak it somehow together with $\hat{\phi}, \vec{x}$. For more, see Section 11.

It may be surprising however that these core axioms *do not require any security of the encryption*. It is purely a consequence of the definition of key compromise and derivability predicates. (The axiom that requires CCA2 security is the fresh keys are uncompromised axiom later.)
Here we want to allow x to be any secret thing, such as a nonce, so only $x \preccurlyeq \hat{\phi}$ was assumed. That is why in the KDM definition of key compromise we needed to allow functions depending on such secret items to be submitted to the encryption oracles in the semantics of $\rhd^{\mathfrak{O}}$and $\blacktriangleright^{\mathfrak{O}}$.

- Encryptions with uncompromised keys do not compromise:
  – IND-CCA2 case. If $\mathfrak{O}$ is either aic2 or sic2, then

$$\begin{aligned}&\mathsf{RanGen}(K) \wedge \mathsf{RanGen}(K') \wedge \mathsf{fresh}(R; \hat{\phi}, \vec{x}, x, K, K')\\ &\wedge \vec{x}, x \preccurlyeq \hat{\phi} \wedge \hat{\phi}, \vec{x}, \{x\}^R_{eK'} \blacktriangleright^{\mathfrak{O}} K\\ &\longrightarrow \hat{\phi}, \vec{x}, x \blacktriangleright^{\mathfrak{O}} K' \vee \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{O}} K\end{aligned}$$

That is, if $\hat{\phi}, \vec{x}, \{x\}^R_{eK'}$ compromised K, then either K is already compromised without $\{x\}^R_{eK'}$, or K' was already compromised by $\hat{\phi}, \vec{x}, x$. Note that this includes x, the encrypted term. This means that x itself (with $\hat{\phi}, \vec{x}$) should not compromise K' if we want $\{x\}^R_{eK'}$ to be safe. This is the generalization of that key cycles may compromise CCA2 encryption. In Section 11 we will see how this axiom deals with key cycles.

  – KDM-CCA2 case. If $\mathfrak{O}$ is akc2 or skc2, then

$$\begin{aligned}&\mathsf{RanGen}(K) \wedge \mathsf{RanGen}(K') \wedge \mathsf{fresh}(R; \hat{\phi}, \vec{x}, x, K, K')\\ &\wedge \vec{x}, x \preccurlyeq \hat{\phi} \wedge \hat{\phi}, \vec{x}, \{x\}^R_{eK'} \blacktriangleright^{\mathfrak{O}} K\\ &\longrightarrow \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{O}} K' \vee \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{O}} K\end{aligned}$$

This is basically the same as the previous one, except again that $\hat{\phi}, \vec{x} \blacktriangleright K'$ does not contain x.

Again, soundness of these axioms follow directly from the definition of key usability, and it does not depend on what encryption is used.
In the above formulas, K and K' could be allowed to encrypt different kinds of encryptions, not necessarily the same, we just did not want to overload our formulas.

- **Axioms for Freshly Generated Items.**
  – Fresh keys are not compromised: The intuition of this axiom is that if K is fresh, then it can be used for secure encryption: $\mathsf{keyfresh}(K; \hat{\phi}, \vec{x}) \wedge \vec{x} \preccurlyeq \hat{\phi} \longrightarrow \hat{\phi}, \vec{x} \not\blacktriangleright^{\mathfrak{O}} K$
  This axiom is sound if the encryption for which K is generated (correctly) is CCA2 secure. Depending on which $\mathfrak{O}$ is in the axiom, the encryption needs to have the corresponding level of security. *This is the only axiom where the security of the encryption is necessary.*
  The reader may wonder that proving the KDM case, what happens to the variables not known to the protocol adversary in the submitted functions as the standard KDM encryption oracle only fills in the gaps of keys, not other unknown items. However, in a KDM attack created by the failure of the axiom, the attacker simulates the protocol, and all honestly generated items except for the keys and random inputs to the encryptions in question are available to him.
  – Fresh items do not compromise: they were generated independently and as they have not been sent out, they have not had a chance to compromise other items:
  $\mathsf{fresh}(x; \hat{\phi}, \vec{x}, y) \wedge \vec{x}, y \preccurlyeq \hat{\phi} \wedge \hat{\phi}, \vec{x}, x \blacktriangleright^{\mathfrak{O}} y \rightarrow \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{O}} y$

8.2 Ciphertext Integrity

8.2.1 Semantics of Ciphertext Integrity

DEFINITION 8.2. *We define INT-CTXT compromise as: We say that* $\mathcal{M}^c, \Pi, S, \sigma \models^{s4} \hat{\phi}, \vec{x} \blacktriangleright^{\text{ic}} K$, *if and only if* $\mathcal{M}^c, \Pi, S, \sigma \models^{s4} \vec{x} \not\preccurlyeq \hat{\phi} \vee \neg \mathit{RanGen}(K)$, *or there is a PPT algorithm* $\mathcal{A}^{\mathcal{O}^{sic2}}$, *with*

$$\mathcal{M}^c, \Pi, S, \sigma \models \mathit{sdec}(\mathcal{A}^{\mathcal{O}^{sic2}}(\hat{\phi}, \vec{x}), K) \neq \bot$$

And on S, *the output of* $\mathcal{A}^{\mathcal{O}^{sic2}}(\hat{\phi}, \vec{x})$ *is not equal any of the outputs of the encryption oracles, and it is not any of the bit strings corresponding to the honest encryptions in* $\hat{\phi}, \vec{x}$. *Let*
$\mathcal{M}^c, \Pi, S, \sigma \models^{c} \hat{\phi}, \vec{x} \blacktriangleright^{\text{ic}} x$ *iff* $\mathcal{M}^c, \Pi, S, \sigma \models^{s4} \Box\Diamond(\hat{\phi}, \vec{x} \blacktriangleright^{\text{ic}} x)$.

The reason for using oracle $\mathcal{O}^{\text{sic2}}$ is that the definition of INT-CTXT security [14] allows the use of encryption and decryption oracles.

8.2.2 Axioms for Ciphertext Integrity Key Compromise

- Let $\mathsf{SameEnc}(\vec{x}; \vec{y})$ be the constraint as before. Then $\mathsf{SameEnc}(\vec{x}; \vec{y}) \wedge \vec{x}, x = \vec{y}, y \longrightarrow (\hat{\phi}, \vec{x} \blacktriangleright^{\text{ic}} x \leftrightarrow \hat{\phi}, \vec{y} \blacktriangleright^{\text{ic}} y)$.
- Derivability implies compromise: $\hat{\phi}, \vec{x} \rhd K \longrightarrow \hat{\phi}, \vec{x} \blacktriangleright^{\text{ic}} K$
- Increasing capabilities for key compromise: $\hat{\phi}, \vec{x} \blacktriangleright^{\text{ic}} K \longrightarrow \hat{\phi}, \vec{x}, x \blacktriangleright^{\text{ic}} K$
- Commutativity: If $\vec{x}'$ is a permutation of $\vec{x}$, then $\hat{\phi}, \vec{x} \blacktriangleright^{\text{ic}} K \longrightarrow \hat{\phi}, \vec{x}' \blacktriangleright^{\text{ic}} K$.
- Transitivity: $\hat{\phi}, \vec{x} \rhd \vec{y} \wedge \hat{\phi}, \vec{x}, \vec{y} \blacktriangleright^{\text{ic}} K \longrightarrow \hat{\phi}, \vec{x} \blacktriangleright^{\text{ic}} K$
- Uncompromised key's encryption cannot be faked:

$$\begin{aligned}&\mathsf{RanGen}(K) \wedge \hat{\phi}, \vec{x} \rhd y \wedge \mathit{dec}(y, dK) \neq \bot\\ &\wedge \forall x R(y = \{x\}^R_{eK} \rightarrow \{x\}^R_{eK} \not\sqsubseteq \hat{\phi}, \vec{x}) \longrightarrow \hat{\phi}, \vec{x} \blacktriangleright^{\text{ic}} K\end{aligned}$$

This means the adversary cannot compute a y which decrypts to something meaningful. This is exactly what we need from the INT-CTXT property, namely, that the encryption cannot be faked. Again, soundness of this axiom does not need INT-CTXT encryption, it is immediate from our semantics.

- Encryptions with uncompromised keys do not compromise:
  - For the IND case, we have

$$\mathsf{RanGen}(K) \wedge \mathsf{RanGen}(K') \wedge \mathsf{fresh}(R; \hat{\phi}, \vec{x}, x, K, K') \\ \wedge\ \vec{x}, x \preccurlyeq \hat{\phi}\ \wedge\ \hat{\phi}, \vec{x}, \{|x|\}^R_{K'} \blacktriangleright^{\mathrm{ic}} K \\ \longrightarrow\ \hat{\phi}, \vec{x}, x \blacktriangleright^{\mathrm{sic2}} K' \ \vee\ \hat{\phi}, \vec{x} \blacktriangleright^{\mathrm{ic}} K$$

  - For the KDM case, we have

$$\mathsf{RanGen}(K) \wedge \mathsf{RanGen}(K') \wedge \mathsf{fresh}(R; \hat{\phi}, \vec{x}, x, K, K') \\ \wedge\ \vec{x}, x \preccurlyeq \hat{\phi}\ \wedge\ \hat{\phi}, \vec{x}, \{|x|\}^R_{K'} \blacktriangleright^{\mathrm{ic}} K \\ \longrightarrow\ \hat{\phi}, \vec{x} \blacktriangleright^{\mathrm{skc2}} K' \ \vee\ \hat{\phi}, \vec{x} \blacktriangleright^{\mathrm{ic}} K$$

  Soundness of these follow from the compromise definitions.
- Fresh keys are not INT-CTXT compromised if encryption is INT-CTXT secure:
  - $\mathsf{keyfresh}(K; \hat{\phi}) \longrightarrow \hat{\phi} \not\blacktriangleright^{\mathrm{ic}} K$. The intuition of this axiom is that if the encryption is INT-CTXT secure and if K is fresh, then the adversary cannot fake encryptions with this key.
- Fresh items do not compromise: $\mathsf{fresh}(x; \hat{\phi}, \vec{x}, K) \ \wedge\ \vec{x} \preccurlyeq \hat{\phi} \ \wedge\ \hat{\phi}, \vec{x}, x \blacktriangleright^{\mathrm{ic}} K \ \longrightarrow\ \hat{\phi}, \vec{x} \blacktriangleright^{\mathrm{ic}} K$

9. ON CONGRUENCE OF EQUALITY

Note that the semantics of the equality predicate $=$ is not defined as identity in the domain. In fact, on the left-hand sides of the predicates $\rhd^{\mathfrak{D}}$and $\blacktriangleright^{\mathfrak{D}}$, equal terms cannot be freely substituted. This might cause problems with decidability, the result in [20] heavily builds on the fact that $\rhd$ is invariant under substitution with respect to equal terms. There is, however a solution if we observe that CCA2 security implies that encryptions cannot be faked: for CCA2 secure encryption schemes, we can define the semantics of $\hat{\phi}, \vec{x} \rhd^{\mathfrak{D}} x$ such that only those ciphers are not decrypted by the decryption oracles that are *necessary* for the computation of $[\![\hat{\phi}, \vec{x}]\!]$. Here, when we say an encryption is necessary, we mean the encryption cannot be omitted and the number of encryptions reduced by this in the process computing $[\![\hat{\phi}, \vec{x}]\!]$. The reason we did not define our predicates this way is that we did not want the well-definedness of our predicates depend on whether the encryption satisfies CCA2 security. But if the encryption does satisfy CCA2 security, then $\rhd^{\mathfrak{D}}$ and $\blacktriangleright^{\mathfrak{D}}$can be defined in the above way and all axioms that we have listed are also valid for those new definitions, and on the top of it, $=$ would be a congruence relation.

The symmetric Needham-Schroeder protocol proof as well as the NSL proof work either way, and we believe this issue does not make a big difference in protocol proofs in general. But for automation, the two definitions might make a big difference.

10. SOUNDNESS OF AXIOMS

THEOREM 10.1 (SOUNDNESS). *With the computational interpretations of derivability and key compromise predicates, the axioms are computationally sound. For the "fresh keys are not compromised", it is necessary that the implementation of the encryption satisfies the corresponding (symmetric or asymmetric, IND or KDM-CCA2 security, or INT-CTXT ciphertext integrity). Soundness of the other axioms do not require that. Furthermore, the no-telepathy axiom requires that freshly generated items are guessable only with negligible probability.*

Note, unlike for general soundness (Theorem 5.3), here the number of sessions in the computational execution does not have to be bounded in the security parameter.

We detail the proofs in the IND case, the KDM case is sketched, details will be included in the long version.

PROOF. As the soundness proofs of the axioms for derivability with oracles are essentially the same as the soundness proofs for key compromise below, we skip that and focus on key compromise.

- Substitutability of equal terms: The reason is that according to the definition of key compromise, compromise of the item on the right hand side of $\blacktriangleright^{\mathfrak{D}}$only depends on the bit string that is associated to the term there, and not on the structure of the term. This is in contrast with the left hand side. The notion that anything can be submitted to the decryption oracle that is not an encryption on the left clearly depends on the term structure on the left, so we have to make sure that in $\vec{x}$ and $\vec{y}$ the same encryption values occur.
- Derivability implies compromise: Soundness of this axiom is rather trivial, but we write it out for clarity. In order to show that in any protocol execution and n.n. set S, we have $\mathcal{M}^c, \Pi, S \models^{\mathrm{c}} \forall \vec{x} K(\hat{\phi}, \vec{x} \rhd^{\mathfrak{D}} K \longrightarrow \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K)$, by the computational semantics we have to show that for any evaluation σ of the variables, and for any $S' \subseteq S$ non-negligible set, $\mathcal{M}^c, \Pi, S', \sigma \models^{\mathrm{c}} \hat{\phi}, \vec{x} \rhd^{\mathfrak{D}} K$ implies $\mathcal{M}^c, \Pi, S', \sigma \models^{\mathrm{c}} \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$. So suppose $\mathcal{M}^c, \Pi, S', \sigma \models^{\mathrm{c}} \hat{\phi}, \vec{x} \rhd^{\mathfrak{D}} K$ holds. To show $\mathcal{M}^c, \Pi, S', \sigma \models^{\mathrm{c}} \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$, let us take any n.n. $S'' \subseteq S'$. By $\mathcal{M}^c, \Pi, S', \sigma \models^{\mathrm{c}} \hat{\phi}, \vec{x} \rhd^{\mathfrak{D}} K$, there is a n.n. $S''' \subseteq S''$ and an algorithm $\mathcal{B}^{\mathcal{O}^{\mathfrak{D}}}$ such that $\mathcal{M}^c, \Pi, S''', \sigma \models \mathcal{B}^{\mathcal{O}^{\mathfrak{D}}}(\hat{\phi}, \vec{x}) = K$. We can chose $\mathcal{A}_1^{\mathcal{O}}$ in the IND definition of key compromise, and x in the KDM definition, simply to be a random bit string $r(\omega)$ of length η. $\mathcal{A}_{21}^{\mathcal{O}}$ can take the key and decrypt its third input. $\mathcal{A}_{22}^{\mathcal{O}}$ can be chosen to be identically 0^η. If the third input is the encryption of $r(\omega)$ then $r(\omega)$ is received after the decryption, so the output of $\mathcal{A}_{21}^{\mathcal{O}}$ and $\mathcal{A}_{22}^{\mathcal{O}}$ differ overwhelmingly. On the other hand, if it is the encryption $0^{|r(\omega)|}$, then the outputs of $\mathcal{A}_{21}^{\mathcal{O}}$ and $\mathcal{A}_{22}^{\mathcal{O}}$ always agree. So by the definition of key compromise, $\mathcal{M}^c, \Pi, S', \sigma \not\models^{\mathrm{c}} \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$ holds as there is such an S''' for all S''.
- Increasing capabilities for key compromise: If $\hat{\phi}, \vec{x} \not\blacktriangleright^{\mathfrak{D}} K$ holds, there are $\mathcal{A}_1^{\mathcal{O}}$, R, etc in the definition of key compromise. The same items are good for $\hat{\phi}, \vec{x}, x' \not\blacktriangleright^{\mathfrak{D}} K$, ignoring x'.
- Commutativity: Trivial, the definition of key compromise is invariant under the change of the order of the list $\vec{x}$.
- Transitivity: For any S, assuming $\mathcal{M}^c, \Pi, S, \sigma \not\models^{\mathrm{c}} \hat{\phi}, \vec{x} \rhd^{\mathfrak{D}} \vec{y}$ and $\mathcal{M}^c, \Pi, S, \sigma \not\models^{\mathrm{c}} \hat{\phi}, \vec{x}, \vec{y} \blacktriangleright^{\mathfrak{D}} K$, we have to show that $\mathcal{M}^c, \Pi, S, \sigma \not\models^{\mathrm{c}} \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$. Take an arbitrary $S' \subseteq S$. By $\mathcal{M}^c, \Pi, S, \sigma \not\models^{\mathrm{c}} \hat{\phi}, \vec{x} \rhd^{\mathfrak{D}} \vec{y}$, there is a $S'' \subseteq S'$ and an $\mathcal{A}^{\mathcal{O}}$ algorithm such that $\mathcal{M}^c, \Pi, S'', \sigma \not\models \mathcal{A}^{\mathcal{O}}(\hat{\phi}, \vec{x}) = \vec{y}$. By $\mathcal{M}^c, \Pi, S, \sigma \not\models^{\mathrm{c}} \hat{\phi}, \vec{x}, \vec{y} \blacktriangleright^{\mathfrak{D}} K$, there is a $S''' \subseteq S''$ such that for this S''', the conditions in the definition of key compromise (in place of S'') hold. Also, we have $\mathcal{M}^c, \Pi, S''', \sigma \not\models \mathcal{A}^{\mathcal{O}}(\hat{\phi}, \vec{x}) = \vec{y}$. So in the key compromise definition applied to the satisfaction $\mathcal{M}^c, \Pi, S, \sigma \not\models^{\mathrm{c}} \hat{\phi}, \vec{x}, \vec{y} \blacktriangleright^{\mathfrak{D}} K$, the algorithms $\mathcal{A}_1^{\mathcal{O}}$, $\mathcal{A}_{21}^{\mathcal{O}}$ and $\mathcal{A}_{22}^{\mathcal{O}}$ can run $\mathcal{A}^{\mathcal{O}}$ as a subroutine to compute $\vec{y}$, so they do not need it as an input. Since there is such a $S''' \subseteq S'$ for all $S' \subseteq S$, we have $\mathcal{M}^c, \Pi, S, \sigma \not\models^{\mathrm{c}} \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$.

- Secrecy of CCA2 encryption: For the IND case, we have

$$\begin{aligned}&\mathsf{RanGen}(K) \ \wedge\ \mathsf{fresh}(R;\hat{\phi},\vec{x},x,y,K)\\ &\wedge\ \vec{x},x,y \preccurlyeq \hat{\phi}\ \wedge\ \hat{\phi},\vec{x},\{x\}^R_{eK} \rhd^{\mathfrak{D}} y\\ &\longrightarrow\ \hat{\phi},\vec{x},x \blacktriangleright^{\mathfrak{D}} K\ \vee\ \hat{\phi},\vec{x} \rhd^{\mathfrak{D}} y\end{aligned}$$

where $\mathfrak{D}$ is either aic2 or sic2. Soundness of this follows easily from our definition of key compromise and derivability with oracle access. Note that CCA2 security of the encryption is not needed in the following argument: We move $\hat{\phi},\vec{x} \rhd^{\mathfrak{D}} y$ to the premise, it becomes $\hat{\phi},\vec{x} \not\rhd^{\mathfrak{D}} y$. Let us denote by θ the premise received this way:

$$\begin{aligned}\theta \equiv\ &\mathsf{RanGen}(K)\ \wedge\ \mathsf{fresh}(R;\hat{\phi},\vec{x},x,y,K)\\ &\wedge\ \vec{x},x,y \preccurlyeq \hat{\phi}\ \wedge\ \hat{\phi},\vec{x},\{x\}^R_{eK} \rhd^{\mathfrak{D}} y\ \wedge\ \hat{\phi},\vec{x} \rhd^{\mathfrak{D}} y\end{aligned}$$

We have to show that for any σ evaluation of free variables and S non-negligible set, if $\mathcal{M}^c,\Pi,S,\sigma \models^c \theta$ holds, then $\mathcal{M}^c,\Pi,S,\sigma \models^c \hat{\phi},\vec{x},x \blacktriangleright^{\mathfrak{D}} K$ is also satisfied. So suppose $\mathcal{M}^c,\Pi,S,\sigma \models^c \theta$. For $\mathcal{M}^c,\Pi,S,\sigma \models^c \hat{\phi},\vec{x},x \blacktriangleright^{\mathfrak{D}} K$, take any non-negligible set $S' \subseteq S$. As $\mathcal{M}^c,\Pi,S,\sigma \models^c \theta$ implies that $\mathcal{M}^c,\Pi,S',\sigma \models^c \hat{\phi},\vec{x},\{x\}^R_{eK} \rhd^{\mathfrak{D}} y$ by the semantics of compound formulas (one conjunct in θ is $\hat{\phi},\vec{x},\{x\}^R_{eK} \rhd^{\mathfrak{D}} y$, and the property is preserved under taking subsets), for such an S', by the definition of derivability, there is a non-negligible subset $S'' \subseteq S'$ and an algorithm $\mathcal{A}^{\mathcal{O}}$ such that $\mathcal{M}^c,\Pi,S'',\sigma \models \mathcal{A}^{\mathcal{O}}(\hat{\phi},\vec{x},\{x\}^R_{eK}) = y$. Let $\mathcal{B}^{\mathcal{O}^{\mathfrak{D}}}$ be the algorithm that takes $\hat{\phi},\vec{x},x$ as input, submits x to its encryption oracle to receive $\{x\}^{R'}_{eK}$, and then applies $\mathcal{A}^{\mathcal{O}}$ on $(\hat{\phi},\vec{x},\{x\}^{R'}_{eK})$. As this was y with non-negligible probability for R, it is also y with non-negligible probability for R', because y does not depend on either of them, and because the random inputs of the algorithms are required to be fresh. Hence R' is just as good an independent item as R was. So there is a $S''' \subseteq S''$ such that $\mathcal{M}^c,\Pi,S''',\sigma \models \mathcal{A}^{\mathcal{O}}(\hat{\phi},\vec{x},\{x\}^{R'}_{eK}) = y$, and by the definition of $\mathcal{B}^{\mathcal{O}^{\mathfrak{D}}}$, we also have $\mathcal{M}^c,\Pi,S''',\sigma \models \mathcal{B}^{\mathcal{O}^{\mathfrak{D}}}(\hat{\phi},\vec{x},x) = y$. Therefore, $\mathcal{M}^c,\Pi,S''',\sigma \models \mathcal{A}^{\mathcal{O}}(\hat{\phi},\vec{x},\{x\}^{R'}_{eK}) = \mathcal{B}^{\mathcal{O}^{\mathfrak{D}}}(\hat{\phi},\vec{x},x) = y$. Now observe that since x was PPT generated, for any non-negligible S, there must be a length function $\ell(\eta)$ such that the probability that $|x| = \ell(\eta)$ is non-negligible on S. This, means that $\mathcal{A}^{\mathcal{O}}(\hat{\phi},\vec{x},\{0^{|x|}\}^{R''}_{eK}) = \mathcal{B}^{\mathcal{O}^{\mathfrak{D}}}(\hat{\phi},\vec{x},x) = y$ cannot hold on any non-negligible subset of S''', because if it did, then $\mathcal{A}^{\mathcal{O}}(\hat{\phi},\vec{x},\{0^{\ell}\}^{R''}_{eK}) = y$ would also hold non-negligible in S''' contradicting $\mathcal{M}^c,\Pi,S,\sigma \models^c \hat{\phi},\vec{x} \not\rhd^{\mathfrak{D}} y$. So,

$$\mathcal{M}^c,\Pi,S''',\sigma \models \mathcal{A}^{\mathcal{O}}(\hat{\phi},\vec{x},\{x\}^R_{eK}) = \mathcal{B}^{\mathcal{O}^{\mathfrak{D}}}(\hat{\phi},\vec{x},x),$$

but

$$\mathcal{M}^c,\Pi,S''',\sigma \models \mathcal{A}^{\mathcal{O}}(\hat{\phi},\vec{x},\{0^{|x|}\}^{R''}_{eK}) \neq \mathcal{B}^{\mathcal{O}^{\mathfrak{D}}}(\hat{\phi},\vec{x},x),$$

which means $\mathcal{M}^c,\Pi,S,\sigma \models^c \hat{\phi},\vec{x},x \blacktriangleright^{\mathfrak{D}} K$.

The argument for the KDM case is completely analogous, except that from the satisfaction of the same $\mathcal{M}^c,\Pi,S'',\sigma \models^c \theta$, we have to derive $\mathcal{M}^c,\Pi,S'',\sigma \models^c \hat{\phi},\vec{x} \blacktriangleright^{\mathfrak{D}} K$ there. This means that the analogous $\mathcal{B}^{\mathcal{O}^{\mathfrak{D}}}$ in the KDM case is not allowed to use x as an input, so it cannot submit x to the encryption oracle. Instead, in the definition of key compromise for the KDM case we allowed the algorithms to use the encryptions functions of items depending on the names generated that far. Hence $x \preccurlyeq \hat{\phi}$ can be submitted to the oracle in the form of such a function. The only items in x that are neither function symbols nor names generated thus far are new names and handles. But new names can be generated by the submitter, and handles were computed by the adversary from sent messages earlier, the handles of which were again computed earlier from earlier messages. So ultimately, all handles were computed by earlier names and functions, so they can be submitted to the oracle as functions of earlier names. The rest of the proof is exactly the same.

- Next, we have to show that encryptions with uncompromised keys do not compromise. Note again in the proof below that we do not need CCA2 security of the encryption, we only need the definition of key compromise. Instead of the original formula, we show the following in the IND-CCA2 case:

$$\begin{aligned}&\mathsf{RanGen}(K) \wedge \mathsf{RanGen}(K') \wedge \mathsf{fresh}(R;\hat{\phi},\vec{x},x,K,K')\\ &\wedge \vec{x},x \preccurlyeq \hat{\phi}\ \wedge \hat{\phi},\vec{x},\{x\}^R_{eK'} \blacktriangleright^{\mathfrak{D}} K \wedge \hat{\phi},\vec{x},x \not\blacktriangleright^{\mathfrak{D}} K'\\ &\longrightarrow\ \hat{\phi},\vec{x} \blacktriangleright^{\mathfrak{D}} K.\end{aligned}$$

We have to show that for all S non-negligible sets and σ evaluations of variables, if $\mathcal{M}^c,\Pi,S,\sigma$ satisfies the premise, then it satisfies the conclusion as well. So let us suppose it satisfies the premise. We want to conclude $\mathcal{M}^c,\Pi,S,\sigma \models^c \hat{\phi},\vec{x} \blacktriangleright^{\mathfrak{D}} K$. Following the definition of key compromise, take any subset $S' \subseteq S$. By the definition of key compromise applied to $\mathcal{M}^c,\Pi,S,\sigma \models^c \hat{\phi},\vec{x},\{x\}^R_{eK'} \blacktriangleright^{\mathfrak{D}} K$, there are $S'' \subseteq S'$, R', R_1 $\mathcal{A}^{\mathcal{O}}_1$, $\mathcal{A}^{\mathcal{O}}_{21}$ and $\mathcal{A}^{\mathcal{O}}_{22}$ such that, taking the first possibility,

$$\begin{aligned}&\mathcal{M}^c,\Pi,S'',\sigma \models\\ &\mathcal{A}^{\mathcal{O}}_{21}(\hat{\phi},\vec{x},\{x\}^R_{eK'},\{\mathcal{A}^{\mathcal{O}}_1(\hat{\phi},\vec{x},\{x\}^R_{eK'})\}^{R_1}_{eK}))\\ &= \mathcal{A}^{\mathcal{O}}_{22}(\hat{\phi},\vec{x},\{x\}^R_{eK'},\{\mathcal{A}^{\mathcal{O}}_1(\hat{\phi},\vec{x},\{x\}^R_{eK'})\}^{R_1}_{eK})).\end{aligned} \tag{1}$$

but

$$\begin{aligned}&\mathcal{M}^c,\Pi,S'',\sigma \models\\ &\mathcal{A}^{\mathcal{O}}_{21}(\hat{\phi},\vec{x},\{x\}^R_{eK'},\{0^{|\mathcal{A}^{\mathcal{O}}_1(\hat{\phi},\vec{x},\{x\}^R_{eK'})|}\}^{R'_1}_{eK}))\\ &\neq \mathcal{A}^{\mathcal{O}}_{22}(\hat{\phi},\vec{x},\{x\}^R_{eK'},\{0^{|\mathcal{A}^{\mathcal{O}}_1(\hat{\phi},\vec{x},\{x\}^R_{eK'})|}\}^{R'_1}_{eK})).\end{aligned} \tag{2}$$

By $\mathcal{M}^c,\Pi,S'',\sigma \models^c \hat{\phi},\vec{x},x \not\blacktriangleright^{\mathfrak{D}} K'$, from Equation 5, we have that there is an R', and a subset (by restricting R') $S''' \subseteq S''$ such that

$$\begin{aligned}&\mathcal{M}^c,\Pi,S''',\sigma \models\\ &\mathcal{A}^{\mathcal{O}}_{21}(\hat{\phi},\vec{x},\{0^{|x|}\}^{R'}_{eK'},\{\mathcal{A}^{\mathcal{O}}_1(\hat{\phi},\vec{x},\{0^{|x|}\}^{R'}_{eK'})\}^{R_1}_{eK}))\\ &= \mathcal{A}^{\mathcal{O}}_{22}(\hat{\phi},\vec{x},\{0^{|x|}\}^{R'}_{eK'},\{\mathcal{A}^{\mathcal{O}}_1(\hat{\phi},\vec{x},\{0^{|x|}\}^{R'}_{eK'})\}^{R_1}_{eK})).\end{aligned} \tag{3}$$

Equation 6 still holds on S''' and for all R'_1, as R'_1 and R' are independent. By $\mathcal{M}^c,\Pi,S''',\sigma \models^c \hat{\phi},\vec{x},x \not\blacktriangleright^{\mathfrak{D}} K'$, from Equation 6, we get that there is an R'', a subset $S'''' \subseteq S'''$ just by restricting R'', such that

$$\begin{aligned}&\mathcal{M}^c,\Pi,S'''',\sigma \models\\ &\mathcal{A}^{\mathcal{O}}_{21}(\hat{\phi},\vec{x},\{0^{|x|}\}^{R''}_{eK'},\{0^{|\mathcal{A}^{\mathcal{O}}_1(\hat{\phi},\vec{x},\{0^{|x|}\}^{R''}_{eK'})|}\}^{R'}_{eK}))\\ &\neq \mathcal{A}^{\mathcal{O}}_{22}(\hat{\phi},\vec{x},\{0^{|x|}\}^{R''}_{eK'},\{0^{|\mathcal{A}^{\mathcal{O}}_1(\hat{\phi},\vec{x},\{0^{|x|}\}^{R''}_{eK'})|}\}^{R'}_{eK})),\end{aligned} \tag{4}$$

still for all R'. Again, there is a length function $\ell(\eta)$ such that the probability that $|x| = \ell(\eta)$ is non-negligible on S''''. Let the subset of S'''' on which they are equal be S'''''. Since 0^{ℓ} is easily computable, we get that there are algorithms $\mathcal{A}^{\mathcal{O}}_3$,

$\mathcal{A}^{\mathcal{O}}_{41}$ and $\mathcal{A}^{\mathcal{O}}_{42}$ such that

$$\mathcal{M}^c, \Pi, S'''''', \sigma \models$$
$$\mathcal{A}^{\mathcal{O}}_{41}(\hat{\phi}, \vec{x}, \{\mathcal{A}^{\mathcal{O}}_3(\hat{\phi}, \vec{x})\}^{R_1}_{eK}) = \mathcal{A}^{\mathcal{O}}_{42}(\hat{\phi}, \vec{x}, \{\mathcal{A}^{\mathcal{O}}_3(\hat{\phi}, \vec{x})\}^{R_1}_{eK}))$$

but

$$\mathcal{M}^c, \Pi, S'''''', \sigma \models$$
$$\mathcal{A}^{\mathcal{O}}_{41}(\hat{\phi}, \vec{x}, \{0^{|\mathcal{A}^{\mathcal{O}}_3(\hat{\phi}, \vec{x})|}\}^{R'_1}_{eK}) \neq \mathcal{A}^{\mathcal{O}}_{42}(\hat{\phi}, \vec{x}, \{0^{|\mathcal{A}^{\mathcal{O}}_3(\hat{\phi}, \vec{x})|}\}^{R'_1}_{eK})).$$

This exactly means that $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$. If we switch $=$ and $\neq$, we receive the proof of the other case of key compromise.

Here too, the KDM case is proven entirely analogously, but we write the details out for more clarity: We have to show

$$\mathsf{RanGen}(K) \wedge \mathsf{RanGen}(K') \wedge \mathsf{fresh}(R; \hat{\phi}, \vec{x}, x, K, K')$$
$$\wedge \vec{x}, x \preccurlyeq \hat{\phi} \wedge \hat{\phi}, \vec{x}, \{x\}^R_{eK'} \blacktriangleright^{\mathfrak{D}} K \wedge \hat{\phi}, \vec{x} \not\blacktriangleright^{\mathfrak{D}} K'$$
$$\longrightarrow \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K.$$

We have to show that for all S non-negligible sets and σ evaluations of variables, if $\mathcal{M}^c, \Pi, S, \sigma$ satisfies the premise, then it satisfies the conclusion as well. So let us suppose it satisfies the premise. We want to conclude $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$. Following the definition of key compromise, take any subset $S' \subseteq S$. By the definition of key compromise applied to $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \hat{\phi}, \vec{x}, \{x\}^R_{eK'} \blacktriangleright^{\mathfrak{D}} K$, there are $S'' \subseteq S'$, R', R_1 x', $\mathcal{A}^{\mathcal{O}}_{21}$ and $\mathcal{A}^{\mathcal{O}}_{22}$ such that, taking the first possibility,

$$\mathcal{M}^c, \Pi, S'', \sigma \models$$
$$\mathcal{A}^{\mathcal{O}}_{21}(\hat{\phi}, \vec{x}, \{x\}^R_{eK'}, \{x'\}^{R_1}_{eK})) \quad (5)$$
$$= \mathcal{A}^{\mathcal{O}}_{22}(\hat{\phi}, \vec{x}, \{x\}^R_{eK'}, \{x'\}^{R_1}_{eK})).$$

but

$$\mathcal{M}^c, \Pi, S'', \sigma \models$$
$$\mathcal{A}^{\mathcal{O}}_{21}(\hat{\phi}, \vec{x}, \{x\}^R_{eK'}, \{0^{|x'|}\}^{R'_1}_{eK})) \quad (6)$$
$$\neq \mathcal{A}^{\mathcal{O}}_{22}(\hat{\phi}, \vec{x}, \{x\}^R_{eK'}, \{0^{|x'|}\}^{R'_1}_{eK})).$$

By $\mathcal{M}^c, \Pi, S'', \sigma \stackrel{c}{\models} \hat{\phi}, \vec{x} \not\blacktriangleright^{\mathfrak{D}} K'$, from Equation 5, we have that there is an R', and a subset (by restricting R') $S''' \subseteq S''$ such that

$$\mathcal{M}^c, \Pi, S''', \sigma \models$$
$$\mathcal{A}^{\mathcal{O}}_{21}(\hat{\phi}, \vec{x}, \{0^{|x|}\}^{R'}_{eK'}, \{x''\}^{R_1}_{eK})) \quad (7)$$
$$= \mathcal{A}^{\mathcal{O}}_{22}(\hat{\phi}, \vec{x}, \{0^{|x|}\}^{R'}_{eK'}, \{x''\}^{R_1}_{eK}))$$

where $x'' \preccurlyeq \hat{\phi}, \vec{x}, \{0^{|x|}\}^{R'}_{eK'}$ is received from x by computing the handles in it using $\{0^{|x|}\}^{R'}_{eK'}$ instead of $\{x\}^R_{eK'}$. Equation 6 still holds on S''' and for all R'_1, as R'_1 and R' are independent. By $\mathcal{M}^c, \Pi, S''', \sigma \stackrel{c}{\models} \hat{\phi}, \vec{x} \not\blacktriangleright^{\mathfrak{D}} K'$, from Equation 6, we get that there is an R'', a subset $S'''' \subseteq S'''$ just by restricting R'', such that

$$\mathcal{M}^c, \Pi, S'''', \sigma \models$$
$$\mathcal{A}^{\mathcal{O}}_{21}(\hat{\phi}, \vec{x}, \{0^{|x|}\}^{R''}_{eK'}, \{0^{|x''|}\}^{R'}_{eK}) \quad (8)$$
$$\neq \mathcal{A}^{\mathcal{O}}_{22}(\hat{\phi}, \vec{x}, \{0^{|x|}\}^{R''}_{eK'}, \{0^{|x''|}\}^{R'}_{eK})),$$

still for all R'. Again, there is a length function $\ell(\eta)$ such that the probability that $|x| = \ell(\eta)$ is non-negligible on S''''. Let the subset of S'''' on which they are equal be S'''''. Since 0^ℓ is easily computable, we get that there are $x''' \preccurlyeq \hat{\phi}, \vec{x}$ and algorithms $\mathcal{A}^{\mathcal{O}}_{41}$ and $\mathcal{A}^{\mathcal{O}}_{42}$ such that

$$\mathcal{M}^c, \Pi, S''''', \sigma \models$$
$$\mathcal{A}^{\mathcal{O}}_{41}(\hat{\phi}, \vec{x}, \{x'''\}^{R_1}_{eK}) = \mathcal{A}^{\mathcal{O}}_{42}(\hat{\phi}, \vec{x}, \{x'''\}^{R_1}_{eK}))$$

but

$$\mathcal{M}^c, \Pi, S''''', \sigma \models$$
$$\mathcal{A}^{\mathcal{O}}_{41}(\hat{\phi}, \vec{x}, \{0^{|x'''|}\}^{R'_1}_{eK}) \neq \mathcal{A}^{\mathcal{O}}_{42}(\hat{\phi}, \vec{x}, \{0^{|x'''|}\}^{R'_1}_{eK})).$$

This exactly means that $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$. If we switch $=$ and $\neq$, we receive the proof of the other case of key compromise.

- Fresh keys are not compromised: It is here where IND-CCA2 or KDM-CCA2 security of the encryption is used. Let us first consider the IND-CCA2 case. We define our CCA2 attacker against the CCA2 oracle that allows multiple submissions for encryptions, which is equivalent with the original definition [12]. Let us consider the IND-CCA2 case. What we have to prove is that if $\mathcal{M}^c, \Pi, S, \sigma$ satisfies freshness of key K, then $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$ leads to a CCA2 attack to the encryption. Let $\mathcal{M}^c, \Pi, S, \sigma \stackrel{c}{\models} \hat{\phi}, \vec{x} \blacktriangleright^{\mathfrak{D}} K$ hold. That is, for every $S' \subseteq S$, there are $S'' \subseteq S'$, etc. such that (consider the first case),

$$\mathcal{M}^c, \Pi, S'', \sigma \models \mathcal{A}^{\mathcal{O}}_{21}(\hat{\phi}, \vec{x}, \{\mathcal{A}^{\mathcal{O}}_1(\hat{\phi}, \vec{x})\}^R_{eK}) \quad (9)$$
$$= \mathcal{A}^{\mathcal{O}}_{22}(\hat{\phi}, \vec{x}, \{\mathcal{A}^{\mathcal{O}}_1(\hat{\phi}, \vec{x})\}^R_{eK})$$

and

$$\mathcal{M}^c, \Pi, S'', \sigma \models \mathcal{A}^{\mathcal{O}}_{21}(\hat{\phi}, \vec{x}, \{0^{|\mathcal{A}^{\mathcal{O}}_1(\hat{\phi}, \vec{x})|}\}^{R'}_{eK}) \quad (10)$$
$$\neq \mathcal{A}^{\mathcal{O}}_{22}(\hat{\phi}, \vec{x}, \{0^{|\mathcal{A}^{\mathcal{O}}_1(\hat{\phi}, \vec{x})|}\}^{R'}_{eK})$$

for all fresh R'. Note that here we continue the convention that for easier readability, we drop notating the computational interpretations of $\hat{\phi}$, $\vec{x}$, R, etc. Of course, the algorithms act on the computational interpretations and not on the symbolic terms. Note further that $\mathsf{keyfresh}(K; \hat{\phi}, \vec{x}) \wedge \vec{x} \preccurlyeq \hat{\phi}$ means the decryption key or any function of it was never used in $\hat{\phi}, \vec{x}$ (except for decrypting, and in case of symmetric keys encrypting). What the CCA2 attacker has to do is to simulate the protocol execution such that

  - except for K, the CCA2 attacker generates all other keys
  - encryptions (except for that of $\mathcal{A}^{\mathcal{O}}_1(\hat{\phi}, \vec{x})$) with K are done by requesting the encryption oracle (in case of asymmetric encryptions, the adversary can also compute them himself)
  - the attacker keeps a table recording which encryption belongs to which plaintext
  - decryptions of ciphertexts provided by the encryption oracle are done by looking it up in the table
  - decryptions of strings not provided by the oracle are done by submitting to the decryption oracle
  - when the challenge state is reached, the interpretations of $\vec{x}$ and $\mathcal{A}^{\mathcal{O}}_1(\hat{\phi}, \vec{x})$ are computed
  - $\mathcal{A}^{\mathcal{O}}_1(\hat{\phi}, \vec{x})$ is submitted to the encryption oracle along with a string of 0's of the same length. Let c_0 denote the encryption that is received from the oracle. Note that the adversary does not know if this is the encryption of $\mathcal{A}^{\mathcal{O}}_1(\hat{\phi}, \vec{x})$ or of the 0's.

– apply $\mathcal{A}_{21}^{\mathcal{O}}$ and $\mathcal{A}_{22}^{\mathcal{O}}$ to $\hat{\phi}, \vec{x}, c_0$.
– because of (9) and (10) on S'', if the correct bit string was encrypted, the two computations are equal, and if the 0's were encrypted, the two are different.

However, the attacker does not necessarily know when he is inside S'' and when outside. Outside S'' the attacker cannot be sure that equality means the correct encryption was encrypted. To overcome this problem, we finish the above CCA2 attack the following way. Let $\mu(\eta)$ be a function (in the security parameter) of natural numbers.

– the adversary submits pairs of $0^{|\mathcal{A}_1^{\mathcal{O}}(\hat{\phi},\vec{x})|}$ for encryption μ times. Let c_i denote $(i = 1, ..., \mu)$ the encryptions received back from the oracle. Note that these encryptions are known to be encryptions of $0^{|\mathcal{A}_1^{\mathcal{O}}(\hat{\phi},\vec{x})|}$
– applies $\mathcal{A}_{21}^{\mathcal{O}}$ and $\mathcal{A}_{22}^{\mathcal{O}}$ on all of $\hat{\phi}, \vec{x}, c_i$
– CASE 1: if $\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_0) = \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_0)$ but $\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i) \neq \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i)$ for all $i = 1, ..., \mu$, then the CCA2 attacker outputs 1, meaning that his guess is that the oracle encrypted the correct bit string.
– CASE 2: otherwise, the adversary tosses a coin and outputs the result.

We can think of the probability space of the CCA2 attack as $\{0,1\} \times \Omega^\eta$, where $\{0,1\}$ represents the internal bit of the oracles. Even if the internal bit is 0, the simulation of the protocol execution can be done according to the above rules. If b denotes the internal bit of the oracles and b' denotes the output of the CCA2 adversary, the following holds.

$$
\begin{aligned}
&Adv(\mathcal{A}_{\mathrm{CCA2}}) = \mathbf{Prob}\{b = b'\} - \frac{1}{2} \\
&= \frac{1}{2} \cdot \mathbf{Prob}\{b = b' | b = 1\} + \frac{1}{2} \cdot \mathbf{Prob}\{b = b' | b = 0\} - \frac{1}{2} \\
&= \frac{1}{2} \cdot \Big(\mathbf{Prob}\{b = b' | b = 1\} + \mathbf{Prob}\{b = b' | b = 0\} - 1\Big) \\
&= \frac{1}{2} \cdot \Big(\mathbf{Prob}\{b = b' | b = 1\} - \frac{1}{2} + \mathbf{Prob}\{b = b' | b = 0\} - \frac{1}{2}\Big)
\end{aligned}
$$

Let $S_\mu^1 \subseteq \Omega$ be the set where

$$
\begin{aligned}
&\mathcal{M}^c, \Pi, S_\mu^1, \sigma \Vdash= \\
&\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{\mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, \vec{x})\}_{eK}^R) = \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{\mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, \vec{x})\}_{eK}^R)
\end{aligned}
$$

but where

$$\mathcal{M}^c, \Pi, S_\mu^1, \sigma \Vdash \mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i) \neq \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i)$$

for all $i = 1, ..., \mu$, (that is, CASE 1 happens when the real plaintext is encrypted in the CCA2 attack). Note that S_μ^1 depends on the function μ, but still, $S'' \subseteq S_\mu^1$, so S_μ^1 is also non-negligible. Suppose, the internal bit of the oracle is 1. Then, on $\{1\} \times S_\mu^1$, according to our setup of the CCA2 attacker, he outputs 1, giving the correct guess. That is, denoting by $\mathcal{A}_{\mathrm{CCA2}}$ the CCA2 attacker as described above, the output of $\mathcal{A}_{\mathrm{CCA2}}$ is 1 on $\{1\} \times S_\mu^1$. On $\{1\} \times (\Omega \setminus S_\mu^1)$, $\mathcal{A}_{\mathrm{CCA2}}$ tosses a coin, so the probabilities there balance out.
Let $S_\mu^0 \subseteq \Omega$ be the set where

$$
\begin{aligned}
&\mathcal{M}^c, \Pi, S_\mu^0, \sigma \Vdash= \\
&\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{0^{|\mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, \vec{x})|}\}_{eK}^R) = \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, \{0^{|\mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, \vec{x})|}\}_{eK}^R)
\end{aligned}
$$

but where

$$\mathcal{M}^c, \Pi, S_\mu^0, \sigma \Vdash \mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i) \neq \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i)$$

for all $i = 1, ..., \mu$, (that is, CASE 1 happens when the 0's are encrypted in the CCA2 attack). Suppose now the internal bit is 0. Then, on $\{0\} \times S_\mu^0$, according to our setup of the CCA2 attacker, he outputs 1, giving the wrong guess. That is, the output of $\mathcal{A}_{\mathrm{CCA2}}$ is 1 on $\{0\} \times S_\mu^0$. On $\{0\} \times (\Omega \setminus S_\mu^0)$, $\mathcal{A}_{\mathrm{CCA2}}$ tosses a coin, so the probabilities there balance out. The advantage of the attacker is essentially the difference of the probabilities of S_μ^1 and S_μ^0:

$$
\begin{aligned}
&Adv(\mathcal{A}_{\mathrm{CCA2}}) = \\
&= \frac{1}{2} \cdot \Big(\mathbf{Prob}\{b = b' | b = 1\} - \frac{1}{2} + \mathbf{Prob}\{b = b' | b = 0\} - \frac{1}{2}\Big) \\
&= \frac{1}{2} \cdot \Big(\mathbf{Prob}\{S_\mu^1\} \cdot \big(\mathbf{Prob}\{b = b' | b = 1 \wedge S_\mu^1\} - \frac{1}{2}\big) \\
&\quad + \mathbf{Prob}\{\Omega \setminus S_\mu^1\} \cdot \big(\mathbf{Prob}\{b = b' | b = 1 \wedge \Omega \setminus S_\mu^1\} - \frac{1}{2}\big) \\
&\quad + \mathbf{Prob}\{S_\mu^0\} \cdot \big(\mathbf{Prob}\{b = b' | b = 1 \wedge S_\mu^0\} - \frac{1}{2}\big) \\
&\quad + \mathbf{Prob}\{\Omega \setminus S_\mu^0\} \cdot \big(\mathbf{Prob}\{b = b' | b = 1 \wedge \Omega \setminus S_\mu^0\} - \frac{1}{2}\big)\Big) \\
&= \frac{1}{2} \cdot \Big(\mathbf{Prob}\{S_\mu^1\} \cdot \big(1 - \frac{1}{2}\big) + \mathbf{Prob}\{\Omega \setminus S_\mu^1\} \cdot \big(\frac{1}{2} - \frac{1}{2}\big) \\
&\quad + \mathbf{Prob}\{S_\mu^0\} \cdot \big(0 - \frac{1}{2}\big) + \mathbf{Prob}\{\Omega \setminus S_\mu^0\} \cdot \big(\frac{1}{2} - \frac{1}{2}\big)\Big) \\
&= \frac{1}{4} \cdot \Big(\mathbf{Prob}\{S_\mu^1\} - \mathbf{Prob}\{S_\mu^0\}\Big)
\end{aligned}
$$

We did not write it out, but of course each step holds for all fixed values η of the security parameter:

$$Adv^\eta(\mathcal{A}_{\mathrm{CCA2}}) = \frac{1}{4} \cdot \Big(\mathbf{Prob}^\eta\{S_\mu^{1\,\eta}\} - \mathbf{Prob}^\eta\{S_\mu^{0\,\eta}\}\Big).$$

It is clear from the definitions that for all η, we have $S''^\eta \subseteq S_\mu^{1\,\eta}$, and $\mathbf{Prob}^\eta\{S''^\eta\} \leq \mathbf{Prob}^\eta\{S_\mu^{1\,\eta}\}$. Let us observe that the probability that $\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i) \neq \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i)$ holds for a fixed i is:

$$
\begin{aligned}
&\sum_w \mathbf{Prob}^\eta\{\hat{\phi}, x, \mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, x) = w\} \cdot \\
&\cdot \mathbf{Prob}^\eta\{\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i) \neq \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i) \,|\hat{\phi}, x, \mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, x) = w\}
\end{aligned}
$$

Where w runs through all possible outcomes values of the interpretation of the triple $\hat{\phi}, x, \mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, x)$. If w is fixed, the rest of the randomness for $\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i) \neq \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i)$ with differing i's are independent. Hence for any finite index set I,

$$
\begin{aligned}
&\mathbf{Prob}^\eta\{\bigwedge_{i \in I} \mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i) \neq \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i) \,|\hat{\phi}, x, \mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, x) = w\} \\
&\quad = \\
&\prod_{i \in I} \mathbf{Prob}^\eta\{\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i) \neq \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i) \,|\hat{\phi}, x, \mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, x) = w\}
\end{aligned}
$$

Further note that since the only difference between the c_i's is that they have independent random inputs (and this holds also for c_0 when the internal bit is 0), these probabilities are actually all the same:

$$
\begin{aligned}
&\mathbf{Prob}^\eta\{\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i) \neq \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_i) \,|\hat{\phi}, x, \mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, x) = w\} \\
&\quad = \\
&\mathbf{Prob}^\eta\{\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_j) \neq \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_j) \,|\hat{\phi}, x, \mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, x) = w\}
\end{aligned}
$$

Let's call this probability $p_w^\eta{}'$, while let

$$p_w^\eta = \mathbf{Prob}^\eta\{\hat{\phi}, x, \mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, x) = w\}.$$

Note also that when the internal bit is 0,

$$\mathbf{Prob}^\eta\{\mathcal{A}_{21}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_0) = \mathcal{A}_{22}^{\mathcal{O}}(\hat{\phi}, \vec{x}, c_0) \mid \hat{\phi}, x, \mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, x) = w\} = 1 - p_w^\eta{}'.$$

With all the above notation, we have that

$$\mathbf{Prob}^\eta\{S_\mu^{0\,\eta}\} = \sum_w (1 - p_w^\eta{}') \cdot (p_w^\eta{}')^{\mu(\eta)} \cdot p_w^\eta$$

where $(p_w^\eta{}')^\mu$ is the $\mu(\eta)$'th power of $p_w^\eta{}'$. Remember, we assumed that $\mathbf{Prob}\{S''\}$ was non-negligible. This means that there is an $a \in \mathbb{N}$, and a strictly increasing sequence of naturals $\eta \mapsto n(\eta)$, such for all η,

$$\mathbf{Prob}^\eta\{S_\mu^{1\,n(\eta)}\} \geq \mathbf{Prob}^\eta\{S''^{n(\eta)}\} > \frac{1}{n(\eta)^a}. \qquad (11)$$

In the rest of the proof, we show that there is an $\eta \mapsto \mu(\eta)$ polynomial function such that

$$\mathbf{Prob}^\eta\{S_\mu^{0\,\eta}\} \leq \frac{1}{2 \cdot \eta^a} + f_{\mathrm{negl}}(\eta)$$

where f_{negl} is some negligible function. This will mean that because of Equation (11), $\mathbf{Prob}^\eta\{S_\mu^{1\,\eta}\} - \mathbf{Prob}^\eta\{S_\mu^{0\,\eta}\}$ is not negligible for this μ, and we will be done. To this end, consider the set

$$\hat{S}^\eta := \bigcup_{\substack{w\,:\\ \frac{1}{2\eta^a} \leq p_w^\eta{}' \leq 1 - \frac{1}{2\eta^a}}} \Big\{\omega \in \Omega^\eta \,\Big|\, \mathcal{M}^c, \Pi, \{\omega\}, \sigma \models \hat{\phi}, x, \mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, x) = w\Big\}$$

Clearly,

$$\mathbf{Prob}^\eta\{S_\mu^{0\,\eta}\} = \mathbf{Prob}^\eta\{S_\mu^{0\,\eta} \cap \hat{S}^\eta\} + \mathbf{Prob}^\eta\{S_\mu^{0\,\eta} \cap (\Omega^\eta \setminus \hat{S}^\eta)\}.$$

But

$$\begin{aligned}\mathbf{Prob}^\eta\{S_\mu^{0\,\eta} \cap (\Omega^\eta \setminus \hat{S}^\eta)\} &= \sum_{\substack{w\,:\\ p_w^\eta{}' < \frac{1}{2\eta^a}\\ p_w^\eta{}' > 1 - \frac{1}{2\eta^a}}} (1 - p_w^\eta{}') \cdot (p_w^\eta{}')^{\mu(\eta)} \cdot p_w^\eta \\ &\leq \sum_w 1 \cdot \frac{1}{2\eta^a} \cdot p_w^\eta \\ &\leq \frac{1}{2\eta^a} \cdot \sum_w p_w^\eta = \frac{1}{2\eta^a}.\end{aligned}$$

Hence, if we can prove that $\mathbf{Prob}^\eta\{S_\mu^{0\,\eta} \cap \hat{S}^\eta\}$ is negligible, then we are done. For this, we have

$$\begin{aligned}\mathbf{Prob}^\eta\{S_\mu^{0\,\eta} \cap \hat{S}^\eta\} &= \sum_{\substack{w\,:\\ \frac{1}{2\eta^a} \leq p_w^\eta{}' \leq 1 - \frac{1}{2\eta^a}}} (1 - p_w^\eta{}') \cdot (p_w^\eta{}')^{\mu(\eta)} \cdot p_w^\eta \\ &\leq \sum_{\substack{w\,:\\ \frac{1}{2\eta^a} \leq p_w^\eta{}' \leq 1 - \frac{1}{2\eta^a}}} (p_w^\eta{}')^{\mu(\eta)} \cdot p_w^\eta\end{aligned}$$

Let us chose $\mu(\eta) := \eta^{a+1}$. Then, on the set in question,

$$(p_w^\eta{}')^{\mu(\eta)} \leq (1 - \frac{1}{2\eta^a})^{\eta^{a+1}} = ((1 - \frac{1}{2\eta^a})^{\eta^a})^\eta$$

Now, we know that $(1 - \frac{1}{2\eta})^\eta$ and hence $(1 - \frac{1}{2\eta^a})^{\eta^a}$ go to $e^{-1/2}$ from below as η goes to infinity, so we have

$$(p_w^\eta{}')^{\mu(\eta)} \leq e^{-\frac{1}{2}\eta},$$

and

$$\mathbf{Prob}^\eta\{S_{\eta^2}^{0\,\eta} \cap \hat{S}^\eta\} \leq e^{-\frac{1}{2}\eta},$$

which is a negligible function. This means that it is enough for the adversary to encrypt 0's himself $\mu(\eta) = \eta^{a+1}$ many times, so the adversary is still polynomial.

The proof for KDM-CCA2 is exactly analogous. The only difference is that instead of $\mathcal{A}_1^{\mathcal{O}}(\hat{\phi}, \vec{x})$, there is an x there, and the oracles accept the functions to be submitted. When it comes to computing the encryptions of x, the KDM encryption oracle is requested. It is not directly x that is submitted, but a description of a function of the keys instead. Since the KDM adversary is simulating the protocol, all items except for the secret keys and the random inputs to the encryptions in the symmetric case are available to him.

- Fresh items do not compromise: The idea is exactly the same as in case of the derivability predicate. A fresh item can just as well be created by the adversary, it cannot help him.

We now turn to the case of INT-CTXT key compromise.

Proofs of the first six axioms and the last one are entirely identical to the proofs for CCA2 key compromise. The soundness of the "encryptions with uncompromised keys do not compromise" axiom is also analogous: We again show the following:

$$\begin{aligned}&\mathsf{RanGen}(K) \wedge \mathsf{RanGen}(K') \wedge \mathsf{fresh}(R; \hat{\phi}, \vec{x}, x, K, K') \\ &\quad \wedge\ \vec{x}, x \preccurlyeq \hat{\phi} \ \wedge \hat{\phi}, \vec{x}, \{\!|x|\!\}_{K'}^R \blacktriangleright^{\mathrm{ic}} K \ \wedge\ \hat{\phi}, \vec{x}, x \blacktriangleright^{\mathrm{sic2}} K' \\ &\longrightarrow\ \hat{\phi}, \vec{x} \blacktriangleright^{\mathrm{ic}} K\end{aligned}$$

Again, we have to show that for all S non-negligible sets and σ evaluations of variables, if $\mathcal{M}^c, \Pi, S, \sigma$ satisfies the premise, then it satisfies the conclusion as well. So let us suppose it satisfies the premise. We want to show $\mathcal{M}^c, \Pi, S, \sigma \models^{\mathrm{c}} \hat{\phi}, \vec{x} \blacktriangleright^{\mathrm{ic}} K$. Following the definition of key compromise, take any subset $S' \subseteq S$. By the definition of key compromise, $\mathcal{M}^c, \Pi, S, \sigma \models^{\mathrm{c}} \hat{\phi}, \vec{x}, \{\!|x|\!\}_{K'}^R \blacktriangleright^{\mathrm{ic}} K$ implies there is a $S'' \subseteq S$, and a PT algorithm $\mathcal{A}^{\mathcal{O}^{\mathrm{sic2}}}$, such that

$$\begin{aligned}&\mathcal{M}^c, \Pi, S'', \sigma \models \mathit{sdec}(\mathcal{A}^{\mathcal{O}^{\mathrm{sic2}}}(\hat{\phi}, \vec{x}, \{\!|x|\!\}_{K'}^R), K) \neq \bot \\ &\wedge \forall z R'(\mathcal{A}^{\mathcal{O}^{\mathrm{sic2}}}(\hat{\phi}, \vec{x}, \{\!|x|\!\}_{K'}^R) = \{\!|z|\!\}_K^{R'} \rightarrow \{\!|z|\!\}_K^{R'} \not\sqsubseteq \hat{\phi}, \vec{x}, \{\!|x|\!\}_{K'}^R)\end{aligned}$$

and on S'', $\mathcal{A}^{\mathcal{O}^{\mathrm{sic2}}}(\hat{\phi}, \vec{x}, , \{\!|x|\!\}_{K'}^R)$ is not equal any of the outputs of the encryption oracles. Now, we also have that $\mathcal{M}^c, \Pi, S'', \sigma \models^{\mathrm{c}} \hat{\phi}, \vec{x}, x \blacktriangleright^{\mathrm{sic2}} K'$ from the satisfaction of the premise. This gives us that there is some S''' non-negligible subset of S'' such that

$$\begin{aligned}&\mathcal{M}^c, \Pi, S''', \sigma \models \mathit{sdec}(\mathcal{A}^{\mathcal{O}^{\mathrm{sic2}}}(\hat{\phi}, \vec{x}, \{\!|0^{|x|}|\!\}_{K'}^R), K) \neq \bot \ \wedge \\ &\forall z R'(\mathcal{A}^{\mathcal{O}^{\mathrm{sic2}}}(\hat{\phi}, \vec{x}, \{\!|0^{|x|}|\!\}_{K'}^R) = \{\!|z|\!\}_K^{R'} \rightarrow \{\!|z|\!\}_K^{R'} \not\sqsubseteq \hat{\phi}, \vec{x}, \{\!|0^{|x|}|\!\}_{K'}^R)\end{aligned}$$

and on S'', $\mathcal{A}^{\mathcal{O}^{\mathrm{sic2}}}(\hat{\phi}, \vec{x}, , \{\!|0^{|x|}|\!\}_{K'}^R)$ is not equal any of the outputs of the encryption oracles. Again, as the length of x can be guessed, there is a non-negligible $S'''' \subseteq S'''$ and a $\mathcal{B}^{\mathcal{O}^{\mathrm{sic2}}}$ such that

$$\begin{aligned}&\mathcal{M}^c, \Pi, S'''', \sigma \models \mathit{sdec}(\mathcal{B}^{\mathcal{O}^{\mathrm{sic2}}}(\hat{\phi}, \vec{x}), K) \neq \bot \\ &\wedge \forall z R'(\mathcal{B}^{\mathcal{O}^{\mathrm{sic2}}}(\hat{\phi}, \vec{x}) = \{\!|z|\!\}_K^{R'} \rightarrow \{\!|z|\!\}_K^{R'} \not\sqsubseteq \hat{\phi}, \vec{x})\end{aligned}$$

and on S'''', $\mathcal{B}^{\mathcal{O}^{\mathrm{sic2}}}(\hat{\phi}, \vec{x})$ is not equal any of the outputs of the encryption oracles. And that exactly means $\mathcal{M}^c, \Pi, S, \sigma \models^{\mathrm{c}} \hat{\phi}, \vec{x} \blacktriangleright^{\mathrm{ic}} K$. Again, the KDM case is entirely analogous.

Proof of the uncompromised key's encryption cannot be faked axiom is immediate from the semantics of $\blacktriangleright^{\text{ic}}$. If

$$\mathcal{M}^c, \Pi, S, \sigma \models^{\underline{c}} \mathsf{RanGen}(K) \ \wedge \ \hat{\phi}, \vec{x} \rhd y \ \wedge \ dec(y, dK) \neq \bot$$
$$\wedge \ \forall x R(y = \{x\}^R_{eK} \rightarrow \{x\}^R_{eK} \not\sqsubseteq \hat{\phi}, \vec{x})$$

Then for all $S' \subseteq S$, there is a $S'' \subseteq S'$ and an algorithm $\mathcal{A}$ such that $\mathcal{M}^c, \Pi, S'', \sigma \models \mathcal{A}(\hat{\phi}, \vec{x}) = y$. Furthermore, the last conjunct means that the output of the algorithm is not any of the encryptions in $\hat{\phi}, \vec{x}$, and the third conjunct means the decryption does not fail. This is exactly means that $\mathcal{M}^c, \Pi, S, \sigma \models^{\underline{c}} \hat{\phi}, \vec{x} \blacktriangleright^{\text{ic}} K$.

The only remaining axiom is the fresh keys are not compromised axiom for the INT-CTXT case. But that is rather easy. Suppose, the encryption is INT-CTXT secure. $\mathcal{M}^c, \Pi, S, \sigma \models^{\underline{c}} \hat{\phi} \blacktriangleright^{\text{ic}} K$ means there is a $S'' \subseteq S$, and a PT algorithm $\mathcal{A}^{\mathcal{O}^{\text{sic2}}}$, such that

$$\mathcal{M}^c, \Pi, S'', \sigma \models sdec(\mathcal{A}^{\mathcal{O}^{\text{sic2}}}(\hat{\phi}), K) \neq \bot$$
$$\wedge \forall z R'(\mathcal{A}^{\mathcal{O}^{\text{sic2}}}(\hat{\phi}) = \{\!|z|\!\}^{R'}_K \rightarrow \{\!|z|\!\}^{R'}_K \not\sqsubseteq \hat{\phi})$$

and on S'', $\mathcal{A}^{\mathcal{O}^{\text{sic2}}}(\hat{\phi})$ is not equal any of the outputs of the encryption oracles. But that exactly means that there is a non-negligible set (namely S''), on which $\mathcal{A}^{\mathcal{O}^{\text{sic2}}}$ can produce a ciphertext, contradicting the INT-CTXT property. □

11. SIMPLE EXAMPLES

Now let us see on a few simple examples how inconsistency can be shown with the above axioms. In [6], the authors presented some of the most basic examples, therefore the ones we analyze here are a little more complex, all are related to sending keys around. We use symmetric encryption in these examples.

EXAMPLE 11.1. Suppose the first messages in a frame are

$$\phi_3 \equiv \langle (A, B), \{\!|K|\!\}^{R_1}_{K_{AB}}, \{\!|h_2, N|\!\}^{R_2}_K \rangle,$$

with names $KK_{AB}NR_1R_2$, and where the symmetric encryption is IND- (or KDM-) CCA2 secure. We want to show that $\phi_3 \rhd N$ is inconsistent with the axioms, that is, N remains secret. Let now $\mathfrak{O}$ denote either sic2 or skc2. Suppose $\phi_3 \rhd N$ holds. Then we have $\phi_3 \rhd^{\mathfrak{O}} N$ by the more oracles help more axiom. That is the same as $\phi_2, \{\!|h_2, N|\!\}^{R_2}_K \rhd^{\mathfrak{O}} N$. By the no-telepathy axiom, $\phi_2 \not\rhd^{\mathfrak{O}} N$ as $\mathsf{fresh}(N; \phi_2)$ holds (which follows directly from the definition of the freshness constraint, not from axioms). By the 'uncompromised key securely encrypts' axiom for CCA2 symmetric case, with the roles $\vec{x} \equiv \langle\rangle$, $x \equiv \langle h_2, N\rangle$, $y \equiv N$, since we assumed $\phi_2, \{\!|h_2, N|\!\}^{R_2}_K \rhd^{\mathfrak{O}} N$, we also have that either $\phi_2 \rhd^{\mathfrak{O}} N$ (already ruled out) or (depending on $\mathfrak{O}$) $\phi_2, h_2, N \blacktriangleright^{\text{sic2}} K$ or $\phi_2 \blacktriangleright^{\text{skc2}} K$. In the IND-CCA2 case, by the 'fresh items do not compromise' axiom, we then have $\phi_2, h_2 \blacktriangleright^{\text{sic2}} K$ as N does not appear in ϕ_2. Since the handle is always derived from the frame, $\phi_2 \rhd h_2$ holds, hence $\phi_2 \rhd^{\text{sic2}} h_2$ and by the transitivity axiom applied for $\phi_2, h_2 \blacktriangleright^{\text{sic2}} K$ and $\phi_2 \rhd^{\text{sic2}} h_2$, we have $\phi_2 \blacktriangleright^{\text{sic2}} K$, just as we had in the KDM case earlier. But that is the same as (now for both IND and KDM cases) $\phi_1, \{\!|K|\!\}^{R_1}_{K_{AB}} \blacktriangleright^{\mathfrak{O}} K$. By the 'encryptions with uncompromised keys do not compromise' axiom, with roles $K' \equiv K_{AB}$, $\vec{x} \equiv \langle\rangle$ and $x \equiv K$, we have that either $\phi_1 \blacktriangleright^{\mathfrak{O}} K$, or $\phi_1, K \blacktriangleright^{\text{sic2}} K_{AB}$ or $\phi_1 \blacktriangleright^{\text{skc2}} K_{AB}$. However, $\phi_1 \not\blacktriangleright^{\mathfrak{O}} K$ because of the 'fresh keys are not compromised' axiom, and the same is true for $\phi_1 \not\blacktriangleright^{\mathfrak{O}} K_{AB}$. So for the KDM case we have a contradiction and we are done. For the IND case, again by the 'fresh items do not compromise' axiom, $\phi_1, K \blacktriangleright^{\text{sic2}} K_{AB}$ together with $\mathsf{fresh}(K; \phi_1, K_{AB})$ implies $\phi_1 \blacktriangleright^{\text{sic2}} K_{AB}$, hence again we arrived at a contradiction.

It may seem to the reader that the axioms provided in [6] (that is, without key compromise) could also be sufficient to prove that $\phi_3 \rhd N$ is inconsistent with the axioms there by removing items from ϕ_3 in a different order from what we just did in Example 11.1. Namely, the idea would be to proceed the following way: given $\phi_3 \rhd N$, first remove the first encryption (by the secrecy axiom as K_{AB} was never sent out), and receive $\langle (A, B), \{\!|h_2, N|\!\}^{R_2}_K \rangle \rhd N$. Then remove the second encryption (by secrecy as K is not in the frame any more after removing the first encryption) receiving $(A, B) \rhd N$ contradicting the no-telepathy axiom. However, application of the secrecy axiom in [6] (and also in this paper) to $\langle (A, B), \{\!|h_2, N|\!\}^{R_2}_K \rangle \rhd N$ requires $h_2 \preccurlyeq (A, B)$, which means $(A, B) \rhd h_2$, but that does not hold, because h_2 was computed from $\langle (A, B), \{\!|K|\!\}^{R_1}_{K_{AB}} \rangle$. Secrecy axiom can only be used if the frame contains all necessary information for the computation of handles in the plaintext. It is in fact possible to show that $\phi_3 \rhd N$ is consistent with the axioms of [6]. Although those axioms are inconsistent with $h_2 \rhd K$, they do allow h_2 to carry partial information about K sufficient to compromise the second encryption. Without the handle h_2 in the second encryption, the axioms of [6] are sufficient to prove inconsistency, but except for initial ones, protocol messages are normally responses to agent inputs and contain handles.

EXAMPLE 11.2. Now suppose

$$\phi_3 \equiv \langle (A, B), \{\!|K|\!\}^{R_1}_{K_{AB}}, \{\!|K_{AB}, h_2, N|\!\}^{R_2}_K \rangle$$

and let us try to show that $\phi_3 \rhd^{\mathfrak{O}} N$ contradicts the axioms. Note that there is a key cycle in this example, K and K_{AB} encrypt each other. So assume $\phi_3 \rhd^{\mathfrak{O}} N$. For IND-CCA2 security, from the 'uncompromised key securely encrypts' axiom we get $\phi_2, K_{AB}, h_2, N \blacktriangleright^{\text{sic2}} K$ if we follow the same steps as we did in Example 11.1. Then the same way as before, we can remove h_2 and N, and since $\phi_2 \equiv \phi_1, \{\!|K|\!\}^{R_1}_{K_{AB}}$, receive $\phi_1, \{\!|K|\!\}^{R_1}_{K_{AB}}, K_{AB} \blacktriangleright^{\text{sic2}} K$. But this does not lead to a contradiction! According to the equational theory, $K = sdec(\{\!|K|\!\}^{R_1}_{K_{AB}}, K_{AB})$, and by the 'functions are computable' axiom, we get $\phi_1, \{\!|K|\!\}^{R_1}_{K_{AB}}, K_{AB} \rhd^{\text{sic2}} K$. So we always have $\phi_1, \{\!|K|\!\}^{R_1}_{K_{AB}}, K_{AB} \blacktriangleright^{\text{sic2}} K$ too by the 'derivability implies compromise' axiom, there is no contradiction. However, if we have KDM security, then just as in the previous example, using the 'uncompromised key securely encrypts' axiom, $\phi_3 \rhd^{\text{skc2}} N$ immediately leads to $\phi_2 \blacktriangleright^{\text{skc2}} K$, and the rest of the derivation is the same as in the previous example. So in this case, while $\phi_3 \rhd^{\text{sic2}} N$ is consistent with the axioms, $\phi_3 \rhd^{\text{skc2}} N$ is inconsistent.

EXAMPLE 11.3. Now consider

$$\phi_3 \equiv \langle (A, B), \{\!|K|\!\}^{R_1}_{K_{AB}}, \{\!|\{\!|K_{AB}|\!\}^{R_2}_{K'}, h_2, N|\!\}^{R_3}_K \rangle$$

with names $KK'K_{AB}NR_1R_2, R_3$ Strictly speaking, K and K_{AB} are still in cycles, but they do not disturb each other because of K'. Again, assuming IND-CCA2 security, from $\phi_3 \rhd^{\text{sic2}} N$ first $\phi_2, \{\!|K_{AB}|\!\}^{R_2}_{K'}, h_2, N \blacktriangleright^{\text{sic2}} K$ is derived using the 'uncompromised key securely encrypts' axiom as in Example 11.1. As in Example 11.1, h_2 and N are removed: $\phi_2, \{\!|K_{AB}|\!\}^{R_2}_{K'} \blacktriangleright^{\text{sic2}} K$. At this point, the 'encryptions with uncompromised keys do not compromise' axiom implies that either $\phi_2 \blacktriangleright^{\text{sic2}} K$ or $\phi_2, K_{AB} \blacktriangleright^{\text{sic2}} K'$. In the former case, we are back at the situation of Example 11.1 and we arrive at a contradiction. In the latter case, by function (encryption) application on $\phi_1, \{\!|K|\!\}^{R_1}_{K_{AB}}, K_{AB} \blacktriangleright^{\text{sic2}} K'$, we receive that $\phi_1, K, K_{AB}, R_1 \blacktriangleright^{\text{sic2}} K'$, but by 'the fresh items do not compromise' axiom all of K, K_{AB}, R_1 can be removed, and receive that $\phi_1 \blacktriangleright^{\text{sic2}} K'$ contradicting the 'fresh keys are uncompromised' axiom.

12. AN NEW ATTACK ON NSL

Using our technique, we found an attack on the NSL protocol, that, to our knowledge had not been found by others. It was published in [6], but for the sake of a complete presentation, we include in this long version as well.

With the notations of Example 4.1, if we assume that $\mathsf{RanGen}(N) \wedge W(\pi_2(N))$ is computationally satisfiable (where $W(\pi_2(N))$ means $\pi_2(N)$ is an agent name), then we have the following computational attack on the NSL protocol. $\mathsf{RanGen}(N) \wedge W(\pi_2(N))$ is the same as saying that with non-negligible probability, it is possible to choose a name (bit string) Q for an agent such that for the output N of some honest nonce generation, there is a bit string n such that $\langle n, Q\rangle = N$. To show that this is not at all unrealistic, suppose that the pairing $\langle\cdot,\cdot\rangle$ is concatenation, and the length of agent names does not depend on the security parameter, say always 8 bits. Then for any name Q, n can be chosen with $\langle n, Q\rangle = N$ as long as the last four digits of N equals Q, which, if N is evenly generated, is of just $1/2^8$, i.e. non-negligible probability. So this situation is realistic. Now, the attack is the following, it needs two sessions:

1. The adversary choses a name Q as above.
2. The adversary catches the last message $\{N_2\}_{eK_B}$ in a session between A and B, two honest agents.
3. The adversary, acting as agent Q initiates a new session with B, sending $\{N_2\}_{eK_B}$ to him.
4. Since B believes this is a new session with Q, it will parse $\{N_2\}_{eK_B}$ according to its role, namely as $\{N_1', Q\}_{eK_B}$. This will succeed as long as there is an n with $\langle n, Q\rangle = N_2$, that is, it will succeed with non-negligible probability $1/2^8$.
5. B then generates a new nonce, N_2', and sends $\{n, N_2', B\}_{eK_Q}$ to Q.
6. The adversary Q decrypts $\{n, N_2', B\}_{eK_Q}$, reads n, and computes $N_2 = \langle n, Q\rangle$. The secrecy of N_2 is hence broken.

This attack is shown graphically in Figure 3.

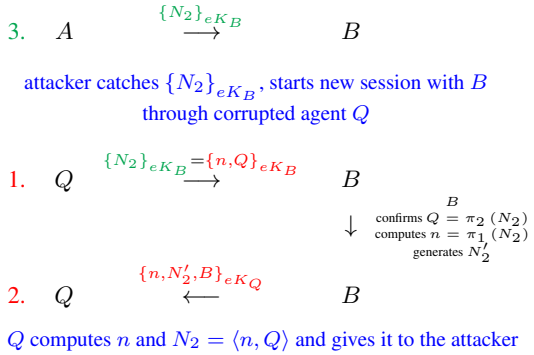

Figure 3: An Attack on the NSL Protocol

So, we can conclude that if $\langle n, Q\rangle = N$ is possible computationally with non-negligible probability, then the protocol fails. In such case, trace-lifting soundness proofs fail as a bit string can be understood both as $\langle n, Q\rangle$ and as N.

Clearly, if the implementation of the protocol is such that B always checks the length of n, then this attack is not possible. It just has to be made sure somehow that the implementation satisfies the $\mathsf{RanGen}(N) \rightarrow \neg W(\pi_2(N))$ property.

Notice that this attack is not a usual type-flaw attack, because even if type-flaw attacks are allowed, honestly generated nonces are normally considered atomic. For example, the reader may suggest that this attack is in fact very similar to the one shown in [27] (as we both wrote it as $N = \langle n, Q\rangle$). However, there is a fundamental difference. The attack in [27] is based on the fact that an honest agent sends a pair with a nonce and an agent name, and the receiving honest agent understands this as a single nonce. In other words, in [27] the honest receiver reads the pair of a nonce and a name into an input variable meant for a nonce. There, n is the honest nonce and N is the input variable. In our attack, it is an actual nonce that is understood by the receiver as a pair of a nonce and a name. In our case, an actual nonce is read into the pair of two input variables: one for a nonce and another for a name. Here N is the honest nonce and n corresponds to the input variable. This is a fundamental difference as in our case there are no atomic objects at all. Even an honest nonce is allowed to be split. To our knowledge, this is the first such attack on the NSL protocol.

13. CORRECTNESS PROOF OF NSL

In this long version of our paper we also recite the verification result of the NSL protocol published in [6]. The reason is that now with the derivation with oracles the syntax and the axioms are somewhat different, but we want to emphasize that the proof is basically the same with the new set of axioms as well.

The correctness of the NSL protocol was done for any (bounded) number of sessions. It was shown that violation of secrecy or authentication is inconsistent with the axioms. In this proof, it was assumed that agent A only executed the initiator role, and agent B only executed the responder role (proof in the case when we allow them to run both roles is also doable, but it is much longer). But both A and B were allowed to have other sessions running with possibly corrupted agents. First it was shown that nonces that were generated by honest initiator A and sent to honest responder B, or vice-versa, remained secret. This was done by picking any step m of the execution tree, and listing all possible messages sent by A and B, and then showing that $\phi_m \not\triangleright^{\mathrm{aic2}} N$ together with the axioms and agent checks imply $\phi_{m+1} \not\triangleright^{\mathrm{aic2}} N$ for each possible sent message. Hence, $\phi_m \not\triangleright^{\mathrm{aic2}} N$, the axioms and the agent checks, and $\phi_{m+1} \triangleright^{\mathrm{aic2}} N$ are inconsistent. Since $\phi_0 \not\triangleright^{\mathrm{aic2}} N$ initially holds by no-telepathy, by induction we have $\phi_m \not\triangleright^{\mathrm{aic2}} N$ (and hence $\phi_m \not\vdash N$) after any finite number of steps m. The reader can see below that the induction hypothesis is a little more complex, but essentially this is what was done.

Once secrecy is proven, authentication and agreement are shown. We pick the point on the execution tree when the responder finished his task, and using that nonces remain secret, together with non-malleability, it is shown that the initiator also finished his task and the corresponding values seen by the two parties match. In other words, B finished, A not finished or values don't match, and the axioms and the agent checks are inconsistent.

13.1 Secrecy

The aim of the secrecy proof is to show that nonces N sent between A and B remain secret. The fact that N is a nonce sent by A to B in the NSL protocol can be expressed as

$$\exists R\big(\{N, A\}^R_{eK_B} \sqsubseteq \hat{\phi}\big).$$

If B sent it to A, that means

$$\exists h R\big(\{\pi_1(dec(h, dK_B)), N, B\}^R_{eK_A} \sqsubseteq \hat{\phi}\big).$$

So, such nonces can be characterized by the condition

$$C[N] \equiv \mathsf{RanGen}(N) \wedge \big(\exists R.\{N, A\}^R_{eK_B} \sqsubseteq \hat{\phi} \\ \vee \exists hR.\{\pi_1\,(dec(h, dK_B))\,, N, B\}^R_{eK_A} \sqsubseteq \hat{\phi}\big).$$

Then the secrecy property we want to show is that

$$\forall N\big(C[N] \longrightarrow \hat{\phi} \not\triangleright^{\mathrm{aic2}} N\big),$$

meaning that such nonces cannot be computed by the adversary. It is equivalent to show that its negation, $\exists N\big(C[N] \wedge \hat{\phi} \triangleright^{\mathrm{aic2}} N\big)$, is inconsistent with the axioms and the agent checks on every possible symbolic trace.

Suppose the total length of the symbolic trace in question is n. At the end of the trace the frame ϕ contains n terms. Let us denote the frames at each node of this trace by ϕ_0, ϕ_1, ϕ_2, etc. Each frame contains one more term than the previous one. Satisfaction of $C[N]$ by this trace means that one of the terms $\{N, A\}^R_{eK_B}$ or $\{\pi_1\,(dec(h, dK_B))\,, N, B\}^R_{eK_A}$ appears in frame ϕ_n for some h, R. Let us fix such N. If $\vec{x}$ is a list of a finite number of nonces $\vec{x} \equiv N_1, ..., N_l$ that were all generated by either A or B (possibly intended to each other, possibly intended for other possibly malicious agents), and they are all different from N, then we say condition $C'[\vec{x}, N]$ is satisfied:

$$C'[N_1, ..., N_l, N] \equiv \\ \bigwedge_{i=1}^{l} \left(\begin{array}{l} \mathsf{RanGen}(N_i) \ \wedge \ N \neq N_i \ \wedge \ \big(\exists QR.\{N_i, A\}^R_{eK_Q} \sqsubseteq \hat{\phi} \vee \\ \exists QhR.\{\pi_1\,(dec(h, dK_B))\,, N_i, B\}^R_{eK_Q} \sqsubseteq \hat{\phi}\big) \end{array} \right)$$

Then an inductive proof is carried out on the length of ϕ_n. As it turns out, in order to avoid loops in the proof, instead of $\exists N\big(C[N] \wedge \hat{\phi} \triangleright^{\mathrm{aic2}} N\big)$, it is better to prove that

$$\exists N \exists \vec{x}\big(C[N] \wedge C'[\vec{x}, N] \wedge \hat{\phi}, \vec{x} \triangleright^{\mathrm{aic2}} N\big) \qquad (12)$$

is inconsistent with the axioms and agent checks. On the symbolic trace, this means that for all n,

$$\exists N \exists \vec{x}\big(C[N] \wedge C'[\vec{x}, N] \wedge \phi_n, \vec{x} \triangleright^{\mathrm{aic2}} N\big)$$

is inconsistent with the axioms and agent checks. We do this by fixing an arbitrary N satisfying $C[N]$, and by showing that if for some $m < n$, $\exists \vec{x}\big(C'[\vec{x}, N] \wedge \phi_m, \vec{x} \triangleright^{\mathrm{aic2}} N\big)$ is inconsistent with the axioms and agent checks, then $\exists \vec{x}\big(C'[\vec{x}, N] \wedge \phi_{m+1}, \vec{x} \triangleright^{\mathrm{aic2}} N\big)$ is also inconsistent with the axioms and agent checks. As at $m = 0$ the statement follows from no telepathy, we are done. This is what the following theorem says. Again, note that within C and C', $\hat{\phi}$ is always ϕ_n and not ϕ_m.

PROPOSITION 13.1. *In the above execution of NSL protocol, let N be such that $C[N]$ is satisfied, and let $m < n$. If for all $\vec{x}$ such that $C'[\vec{x}, N]$ holds, the axioms and agent checks imply (by FOL deduction rules) that $\phi_m, \vec{x} \not\triangleright^{\mathrm{aic2}} N$, then for all $\vec{x}$ such that $C'[\vec{x}, N]$ holds, the axioms and agent checks imply (by FOL deduction rules) that $\phi_{m+1}, \vec{x} \not\triangleright^{\mathrm{aic2}} N$ holds.*

Once this is shown, we still have to prove that the property initially holds, that is, $\exists N \exists \vec{x}\big(C[N] \wedge C'[\vec{x}, N] \wedge \phi_0, \vec{x} \triangleright^{\mathrm{aic2}} N\big)$ is inconsistent with the axioms. Let $C[N]$ and $C'[\vec{x}, N]$ hold for N and $\vec{x} \equiv N_1, ..., N_l$. At step 0, N, $N_1, ..., N_l$ are still fresh (remember, we assumed for simplicity that everything was generated upfront, and clearly, these nonces have not been sent), so by the no telepathy axiom, $\phi_0 \not\triangleright^{\mathrm{aic2}} N$, and then by the independence of fresh items, $\phi_0, N_1 \not\triangleright^{\mathrm{aic2}} N$. Then again by the independence of fresh items, $\phi_0, N_1, N_2 \not\triangleright^{\mathrm{aic2}} N$, etc. So $\phi_0, N_1, ..., N_l \not\triangleright^{\mathrm{aic2}} N$ holds, meaning that $\exists N \exists \vec{x}\big(C[N] \wedge C'[\vec{x}, N] \wedge \phi_0, \vec{x} \triangleright^{\mathrm{aic2}} N\big)$ is indeed inconsistent with the axioms. Therefore, together with the induction step of Proposition 13.1, we have:

THEOREM 13.2 (SECRECY). *Consider a symbolic execution of the NSL protocol, with an arbitrary number of possible dishonest participants and two honest participants A, B that follow the initiator and responder roles correspondingly, and that only execute these roles in each of their bounded number of sessions. Further, consider the convention $\langle x, y, z\rangle \equiv \langle x, \langle y, z\rangle\rangle$.*

Our axioms together with the agent checks and $\mathsf{RanGen}(N) \rightarrow \neg W(\pi_2\,(N))$ imply that for any $n \in \mathbb{N}$ and for any nonce N that was either generated by A and sent to B, or vice versa, $\phi_n \not\triangleright^{\mathrm{aic2}} N$.

The above Theorem states that secrecy of nonces satisfying $C[N]$ is never broken. That is, nonces that were generated by A or B and intended to be sent between each other, remain secret. In particular, requiring $\vec{x}$ to be the empty list, the formula $\exists N\big(C[N] \wedge \hat{\phi} \triangleright^{\mathrm{aic2}} N\big)$, together with the axioms and the agent checks, and $\mathsf{RanGen}(N) \rightarrow \neg W(\pi_2\,(N))$ are inconsistent on any symbolic trace.

13.2 Agreement and Authentication

The agreement from the responder's viewpoint is the following:

$$Resp^B_{NSL}[B, i', N_2, h_2, h_4, R_2] \\ \text{AND} \\ \pi_2\,(dec(h_2, dK_B)) = A$$

$$\Downarrow$$

$$\text{EXIST } i, N_1, h_1, h_3, R_1, R_3 \text{ SUCH THAT} \\ Init^A_{NSL}[A, i, B, N_1, h_1, h_3, R_1, R_3] \\ \text{AND} \\ dec(h_2, dK_B) = \langle N_1, A\rangle \\ \text{AND} \\ dec(h_3, dK_A) = \langle N_1, N_2, B\rangle \\ \text{AND} \\ dec(h_4, dK_B) = N_2$$

where by the implication sign we mean that the agent checks and the axioms imply this. We can also write this within our syntax:

$$c_r(A, B, N_1, N_2) \sqsubseteq \hat{\phi} \\ \wedge \ A = \pi_2\,(dec(h_2, dK_B)) \ \wedge \ N_1 = \pi_1\,(dec(h_2, dK_B)) \\ \longrightarrow \exists h_3. \left(\begin{array}{l} c_i(A, B, N_1, N_2) \sqsubseteq \hat{\phi} \wedge \\ N_2 = \pi_1\,(\pi_2\,(dec(h_3, dK_A))) \end{array} \right)$$

What we have to prove is that the negation of this is inconsistent with the axioms and agent checks. But for that it is sufficient to show that the agent checks and axioms, and the premise of the formula imply the conclusion of this formula, as the following theorem states with the proof available in [5] without oracles, but it is straightforward to rewrite it with oracles.

THEOREM 13.3 (AGREEMENT AND AUTHENTICATION). *We consider a symbolic execution of the NSL protocol, with an arbitrary number of possible dishonest participants and two honest participants A, B that follow the initiator and responder roles correspondingly, and that only execute these roles in each of their bounded number of sessions. Furthermore, consider the convention $\langle x, y, z\rangle \equiv \langle x, \langle y, z\rangle\rangle$.*

Our axioms together with the agent checks and $\mathsf{RanGen}(N) \rightarrow \neg W(\pi_2\,(N))$ are inconsistent with the negation of the formula

$$c_r(\pi_2\,(dec(h_2, dK_B))\,, B, \pi_1\,(dec(h_2, dK_B))\,, N_2) \sqsubseteq \hat{\phi} \\ \wedge A = \pi_2\,(dec(h_2, dK_B)) \\ \longrightarrow \ \exists N_1 h_3. \left(\begin{array}{l} c_i(A, B, N_1, \pi_1\,(\pi_2\,(dec(h_3, dK_A)))) \sqsubseteq \hat{\phi} \\ \wedge N_2 = \pi_1\,(\pi_2\,(dec(h_3, dK_A))) \\ \wedge N_1 = \pi_1\,(dec(h_2, dK_B)) \end{array} \right)$$

14. THE SYMMETRIC NEEDHAM-SCHROEDER PROTOCOL

With the axioms that we presented, we have proven the amended symmetric Needham-Schroeder protocol:

1. $A \to B : A$
2. $B \to A : \{A, N_1\}_{K_{BT}}$
3. $A \to T : \langle A, B, N_2, \{A, N_1\}_{K_{BT}} \rangle$
4. $T \to A : \{N_2, B, K, \{K, N_1, A\}_{K_{BT}}\}_{K_{AT}}$
5. $A \to B : \{K, N_1, A\}_{K_{BT}}$
6. $B \to A : \{N_3\}_K$
7. $A \to B : \{N_3 - 1\}_K$

This protocol first has a key distribution part, and then the distributed key is used to securely encrypt a nonce. We showed that no symbolic (hence computational) attacker succeeds the following way (motivated by [28]). Using IND-CCA2 and INT-CTXT axioms, we first showed by an inductive technique (similar to the NSL proof) that the key K from the trusted party meant for honest A and B are never compromised (compromise is inconsistent with the axioms and agent checks). Then, again with an inductive technique we showed that N_3 is never leaked. Finally, agreement and authentication were shown. Besides the presented axioms, we also needed that adding 1 and subtracting 1 are inverses of each other, and $x - 1 \neq x$. We needed an additional property, namely, that applying the first projection of a pairing on an honestly generated nonce cannot result the nonce itself with more than negligible probability. Triples, quadruples were constructed out of pairs. The detailed proof is available online at the first author's homepage. We assumed that A is running the initiator role in all his sessions, and B is running the responder's role. There is only one trusted party. They all are allowed to run any number of multiple parallel sessions with honest and corrupted agents.

On a note about dynamic corruption, the proof works even if the protocol allows the release of the key K at a later time. Secrecy can still be proven until that point, authentication that was carried out earlier can still be verified.

15. CONCLUSIONS

In this paper we further expanded the framework proposed by Bana and Comon-Lundh [8] for computationally complete symbolic adversary. We have shown how key exchange can be handled. Proofs with this technique are computationally sound without the need of any further assumptions such as no bad keys, etc. that are assumed in other literature. We presented a modular set of axioms that are computationally sound for implementations using IND-CCA2, KDM-CCA2 and INT-CTXT secure encryptions respectively. We illustrated their power via simple examples and the verification entire protocols.

We are investigating extensions of the general soundness theorem in order to account for unbounded number of sessions and also to be able to handle indistinguishability properties. More importantly, we are also researching automation.

16. REFERENCES

[1] P. Adão, G. Bana, J. Herzog, and A. Scedrov. Soundness and completeness of formal encryption: the cases of key-cycles and partial information leakage. *Journal of Computer Security*, 17(5):737–797, 2009.

[2] M. Backes, A. Malik, and D. Unruh. Computational soundness without protocol restrictions. In *CCS'12*, pages 699–711. ACM, 2012.

[3] M. Backes, B. Pfitzmann, and M. Waidner. A composable cryptographic library with nested operations. In *CCS'03*, pages 220–230. ACM, 2003.

[4] M. Backes, B. Pfitzmann, and M. Waidner. The reactive simulatability (rsim) framework for asynchronous systems. *Information and Computation*, 205(12):1685–1720, 2007.

[5] G. Bana, P. Adão, and H. Sakurada. Computationally complete symbolic attacker in action—Long version. Available at IACR ePrint Archive, Report 2012/316.

[6] G. Bana, P. Adão, and H. Sakurada. Computationally Comlete Symbolic Attacker in Action. In *FSTTCS'12*, LIPIcs, pages 546–560. Schloss Dagstuhl, 2012.

[7] G. Bana and H. Comon-Lundh. Towards unconditional soundness: Computationally complete symbolic attacker. Available at IACR ePrint Archive, Report 2012/019.

[8] G. Bana and H. Comon-Lundh. Towards unconditional soundness: Computationally complete symbolic attacker. In *POST'12*, LNCS, pages 189–208. Springer, 2012.

[9] G. Bana, K. Hasebe, and M. Okada. Computational semantics for first-order logical analysis of cryptographic protocols. In *Formal to Practical Security*, volume 5458 of *LNCS*, pages 33–58. Springer, 2009.

[10] G. Barthe, B. Grégoire, and S. Zanella Béguelin. Formal certification of code-based cryptographic proofs. In *POPL'09*, pages 90–101. ACM, 2009.

[11] G. Barthe, B. Grégoire, and S. Zanella Béguelin. Formal certification of code-based cryptographic proofs. In *POPL*, pages 90–101. ACM, 2009.

[12] M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting. In *EUROCRYPT'00*, pages 258–274. Springer, 2000.

[13] M. Bellare, A. Desai, D. Pointcheval, and Ph. Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO'98*, LNCS. Springer, 1998.

[14] M. Bellare and Ch. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *J. Cryptology*, 21(4):469–491, 2008.

[15] B. Blanchet. A computationally sound mechanized prover for security protocols. *IEEE Transactions on Dependable and Secure Computing*, 5(4):193–207, 2008.

[16] J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *EUROCRYPT'09*, LNCS, pages 351–368. Springer, 2009.

[17] H. Comon, C. Marché, and R. Treinen, editors. *Constraints in Computational Logics: Theory and Applications, International Summer School, CCS'99*, LNCS, 2001.

[18] H. Comon-Lundh and V. Cortier. Computational soundness of observational equivalence. In *CCS'08*, pages 109–118. ACM, 2008.

[19] H. Comon-Lundh and V. Cortier. How to prove security of communication protocols? A discussion on the soundness of formal models w.r.t. computational ones. In *STACS'11*, LIPIcs, pages 29–44. Schloss Dagstuhl, March 2011.

[20] H. Comon-Lundh, V. Cortier, and G. Scerri. Tractable inference systems: an extension with a deducibility predicate. In *CADE'13*, LNAI. Springer, 2013.

[21] Hubert Comon-Lundh, Masami Hagiya, Yusuke Kawamoto, and Hideki Sakurada. Computational soundness of indistinguishability properties without computable parsing. In *ISPEC'12*, pages 63–79. Springer, 2012.

[22] V. Cortier and B. Warinschi. Computationally sound, automated proofs for security protocols. In *ESOP'05*, LNCS, pages 157–171, 2005.

[23] A. Datta, A. Derek, J. C. Mitchell, V. Shmatikov, and M. Turuani. Probabilistic polynomial-time semantics for a protocol security logic. In *ICALP'05*, LNCS, pages 16–29. Springer, 2005.

[24] A. Datta, A. Derek, J. C. Mitchell, and B. Warinschi. Computationally sound compositional logic for key exchange protocols. In *CSFW '06*, pages 321–334. IEEE, 2006.

[25] Melvin Fitting. An embedding of classical logic in s4. *The Journal of Symbolic Logic*, 35(4):529–534, 1970.

[26] R. Küsters and M. Tuengerthal. Computational soundness for key exchange protocols with symmetric encryption. In *CCS'09*, pages 91–100. ACM, 2009.

[27] Jonathan K. Millen and Vitaly Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In Michael K. Reiter and Pierangela Samarati, editors, *ACM Conference on Computer and Communications Security*, pages 166–175. ACM, 2001.

[28] F. J. Thayer, J. C. Herzog, and J. D. Guttman. Strand spaces: Proving security protocols correct. *Journal of Computer Security*, 7(1):191–230, 1999.