Cutting Out the Middle Man: A Game-Theoretic Analysis in Computability Logic of the Needham-Schroeder Protocol

Cosimo Perini Brogi*,†, Stella Spadoni†

IMT School for Advanced Studies Lucca, Italy

Abstract

Cryptographic protocols constitute the cornerstone of secure communication in open distributed systems. The systematic formal verification of such protocols gained prominence following Gavin Lowe's 1995 discovery of a structural flaw in the classical Needham-Schroeder Public Key protocol from 1978. This paper presents a novel formal analysis of such protocol through Computability Logic (CoL), a game-theoretic semantics and reasoning system that models interaction between an honest agent and a hostile environment. By formalising the protocol's execution as a game specified in the CoL fragment CL4, we demonstrate that the original vulnerability allows the environment to employ a successful Copycat Strategy isomorphic to the standard Man-in-the-Middle attack (MitM). Conversely, we prove that the revised protocol including Lowe's fix effectively breaks this adversarial advantage, guaranteeing security against the MitM. We propose this case study as a promising starting point for the development of a new methodology in protocol verification.

Keywords

Formal Methods, Needham-Schroeder Protocol, Computability Logic, Protocol Verification, Mutual Authentication, CySec Case Studies, Game-Theoretic Semantics, Lowe's Man-in-the-Middle

1. Introduction

As the reliance upon digital infrastructure deepens, the assurance that cryptographic protocols function correctly – specifically, regarding authentication and key exchange – becomes paramount [1, 2, 3]. Yet, the design of such protocols for secure communication remains notoriously error-prone [4]. The complexity of this task arises not necessarily from the cryptographic primitives themselves but from the very protocol dynamics – i.e. the subtle *logical* interaction between agents (and potential adversaries) over a communication network [5, 6]. Experience has repeatedly demonstrated that intuition is a poor guide in this domain: history is replete with protocols that, despite intuitive correctness, have harboured latent vulnerabilities discovered only years after their inception.

The Needham-Schroeder Public Key protocol (NS-PK) serves as a paradigmatic example of this phenomenon. For nearly two decades following its design in [7], the protocol was widely regarded as a secure method for mutual authentication. Lowe's finding of a canonical Man-in-the-Middle attack in [8, 9] underscored a critical realisation in the field of formal methods: informal reasoning and standard debugging techniques are insufficient to guarantee the security properties of interaction-heavy protocols [10]. Consequently, there has been a sustained drive towards rigorous mathematical frameworks capable of modelling the adversarial nature of "cryptographic dialogues".

While various formalisms – such as strand spaces [11], process algebras [12], and BAN logic [13] – have been employed to analyse protocols, this paper posits that an approach based on *logical verification* via Computability Logic (CoL) offers a distinct and powerful semantic advantage by virtue of its game-theoretic roots. CoL reinterprets logic as a theory of computation and interaction. Unlike classical

Joint National Conference on Cybersecurity (ITASEC & SERICS 2026), February 09-13, 2026, Cagliari, IT

*Corresponding author.

†These authors contributed equally.

cosimo.perinibrogi@imtlucca.it (C. Perini Brogi); stella.spadoni@imtlucca.it (S. Spadoni)

https://logicosimo.gitlab.io/ (C. Perini Brogi); https://shpadocs.github.io/ (S. Spadoni)

0000-0001-7883-5727 (C. Perini Brogi); 0009-0007-3169-4745 (S. Spadoni)

logic – which deals with static truth values – CoL views formal statements as computational tasks, i.e. games played between a proponent (Machine) and an opponent (Environment). This framework enables the systematic development of a uniform treatment of security properties as assertions of game winnability: in the case of NS-PK, the honest participants strive to win a 'mutual authentication' game, while the intruder attempts to thwart this goal, achieving victory by extracting confidential information or successfully impersonating a sender.

Beyond this general verification capability, a unique strength of CoL's game-theoretic approach is the identification of a structural correspondence between logical validity and parallel session attacks. In CoL, the tautology $A \to A$ is proven via the *Copycat Strategy*, where moves are mirrored between the antecedent and consequent of the logical implication. Cryptographically, this strategy is isomorphic to a Man-in-the-Middle (MitM) relay attack. Therefore, proving the security of a protocol in CoL entails demonstrating that the logical structure of the game prevents the Copycat Strategy – typically by binding messages to unique session identifiers or identities, thereby breaking the symmetry required for $A \to A$ to hold in the adversarial context.

Contemporary threat landscapes substantiate this theoretical isomorphism, offering tangible instances of the Copycat where the required cryptographic binding is absent or compromised. For instance, the recent 'Terrapin' attack against the OpenSSH protocol exploits a manipulation of the handshake sequence [14]; by truncating the negotiation before the integrity channel is established, the attacker effectively preserves the symmetry required to inject malicious data. This mirrors the theoretical structural weakness of a game lacking proper move-binding, a vulnerability ideally specifiable and analysable within CoL.[1]

1.1. Our Contribution

In the present paper, we explore the capabilities of CoL for this 'MitM scenario' starting from its most relevant historical instance: we propose a complete formalisation in the CoL fragment CL4 [18] of the Needham-Schroeder Public Key protocol, as a case study for Computability Logic applied to cybersecurity. Specifically, our technical contributions are the following:

1. We identify and formalise the structural correspondence between the logical *Copycat Strategy* (validating $A \to A$) and the cryptographic *Man-in-the-Middle* attack;
2. We provide a *game-theoretic representation* in CoL of the original NS-PK protocol, proving its insecurity by demonstrating that Environment can successfully deploy the Copycat Strategy *to break authentication*; and
3. We formalise the Needham-Schroeder-Lowe protocol (NSL), verifying in CoL that *the inclusion of identity tags breaks the logical symmetry required for the attack*, thereby guaranteeing a winning strategy for the honest participants.

Collectively, these contributions aim to establish CoL not merely as a theoretical curiosity, but as a robust semantic framework for detecting and correcting interaction-based security flaws in real-world protocols. By grounding the analysis in game semantics, we move beyond static verification to a dynamic model where security is synonymous with the existence of a *winning strategy* for the honest participants. This offers a distinct methodological advantage: in CoL, the protocol's logical validity implies the constructive existence of an algorithmic strategy that provably defeats any computational adversary in the defined interaction.

Paper Contents. The remainder of the paper is organised as follows.
Section 2 provides a necessary overview of Computability Logic and its CL4 fragment, introducing the fundamental syntax required to formalise the cryptographic exchanges. Section 3 establishes the

[1] This abstract mirroring we advocate for is further exemplified by recent cybersecurity incidents, such as the Lapsus$ group's subversion of MFA [15], the Salt Typhoon Telecom Breach [16], and the Tesla phishing-based MitM attack [17].

groundwork for our analysis by defining the game-theoretic primitives and CoL predicates used in our model of Needham-Schroeder. Here, we extend the base logic CL4 to a system CL4^{Crypt} capable of modelling secure primitives and formally specify the operational flow of the protocol as a CoL game. Section 4 presents our main technical contribution, detailing two key CoL *proofs*: the first one demonstrates the logical vulnerability of the original NS-PK protocol, while the second one verifies the security of the NSL protocol against the MitM strategy. Finally, Section 5 concludes the paper with a discussion of these results and their possible future developments for protocol verification.

2. Background on Computability Logic

Computability Logic (CoL) provides a semantics for formal reasoning where formulas represent interactive computational tasks, referred to as *games*. In this framework, logical validity corresponds to the existence of an effective – possibly, not necessarily algorithmic in the sense of the Church-Turing paradigm [19] – solution to the task.

Each computational problem thus represents a game played out between two agents:

- **Machine** $\mathcal{M}$ – also addressed as $\top$ – representing the system, protocol, or prover;
- **Environment** $\mathcal{E}$ – known as $\bot$ – representing the world, the user, or the adversary.

The former has to act according to an effective (possibly algorithmic) and fully determined behaviour, while the latter is free to act arbitrarily.

As a result, we say that a logical formula F is *computationally true* (or valid) if and only if there exists a computable winning strategy for $\mathcal{M}$ against any possible strategy played by $\mathcal{E}$. Computability is thus interpreted as winnability for $\mathcal{M}$ of a certain game.[2]

In the context of cybersecurity and the adversarial settings discussed here, the *winning strategy* of the Machine ($\mathcal{M}$) corresponds to a *secure protocol implementation*. Conversely, such a game-semantic foundation directly translates the concept of protocol security into the *logical impossibility* for Environment ($\mathcal{E}$) to win the verification game. Therefore, a protocol vulnerability corresponds to the existence of a *winning strategy* for $\mathcal{E}$ – effectively, a *computable attack algorithm* that guarantees the violation of the security assertion.[3]

2.1. The System CL4

Being a conservative extension of classical first-order logic, CL4 is a sound and complete fragment of CoL which is able to balance both expressiveness and good-behaviour out [23]. Specifically, it is a generalisation of fragment CL3 [24], obtained by incorporating the following two sorts of atoms to the language.

Definition 2.1 (Elementary and General Atoms)**.** *We assume a fixed universe of discourse (domain) V for variables, where $Vr \subset V$ is the set of the variables which a game depends upon.*

- ***Elementary Atoms**: These represent non-interactive problems that are either won by $\mathcal{M}$ ($\top$) or $\mathcal{E}$ ($\bot$) and are equivalent to atoms in classical logic. We use the lowercase $p, q, r \ldots$ to denote them.*
- ***General Atoms**: These represent any interactive problem (even elementary ones) and we use the uppercase $P, Q, R \ldots$ to denote them.*

[2] The recent overview [20] provides additional information and technical details about this emerging research area at the intersection of theoretical computer science and applied logic.

[3] This dichotomy holds because the specific CoL games employed for protocol verification belong to the class of *determined games* – specifically, games of finite depth or closed Safety games. For such games, the absence of a winning strategy for one player logically implies the existence of a winning strategy for the opponent [21, 22], guaranteeing no "middle ground" exists between security and vulnerability.

2.1.1. CL4 Formulas

CL4 formulas are built by applying the logical operators of the following signature[4]

$$\{\neg, \wedge, \vee, \sqcap, \sqcup, \bigsqcap, \bigsqcup, \forall, \exists\}$$

to two different types of games: elementary and general ones.

As a result, a CL4 formula is said to be **elementary** if and only if it does not contain **choice operators** (that is: choice conjunction $\sqcap$, choice disjunction $\sqcup$, universal choice quantifier $\bigsqcap$, and existential choice quantifiers $\bigsqcup$) and general atoms. According to such categorisation, formulas of classical logic are nothing but elementary formulas of CL4.

A **surface occurrence** is an occurrence of a formula which is not in the scope of any choice operator. We will write $F[E_1, \ldots, E_n]$ to mean a formula F together with some n fixed surface occurrences of subformulas $E_1, \ldots, E_n$. Writing F in this form sets a context, in which $F[G_1, \ldots, G_n]$ will mean the result of replacing in $F[E_1, \ldots, E_n]$ those occurrences by $E_1, \ldots, E_n$ by $G_1, \ldots, G_n$ respectively.

The **elementarisation** [22] of a formula F is the result of replacing in F all surface occurrences of $\sqcap$ and $\bigsqcap$ subformulas by $\top$, all surface occurrences of $\sqcup$ and $\bigsqcup$ subformulas by $\bot$ and all surface occurrences of general atoms by $\bot$.

Moreover, a formula is said to be **stable** if and only if its elementarisation is classically valid (i.e. provable in some standard axiomatisation of classical first-order logic); otherwise, we deem it as unstable.

We hereby consider **strict games** only, meaning games where at most one of the players is allowed to move at each step. As proven in [22], every strict game is also a **static** one, i.e. a speed-independent interaction which remains well defined even without any particular assumption on speed or time. In these types of games, a winning strategy still remains a winning one no matter how fast the adversary is in making moves. Conversely, a strategy remains a losing one no matter how slow the adversary is in making moves. As a result, a player is allowed to delay certain moves to a later time, without hurting the final outcome of the strategy itself.[5]

The restriction to strict games pairs well with the type of protocol we want to model: the NS-PK protocol is a terminating process with a strictly bounded number of interactions – the full expressive power of infinite-depth dynamic games is unnecessary. Modelling it as a strict game structure is sufficient to capture every possible execution trace without introducing the complexity of infinite duration games.

Games generalise to gameframes analogously to the way propositional atoms generalise to predicates in classical logic.

Definition 2.2 (Types of games in CL4, from [20])**.** *Let (Vr, G) be an n-ary **gameframe** where G is a mapping that assigns a game $G(e)$ ("instance of G") to each variable valuation e.*

- ***Elementary Gameframes (Predicates):** We say that a gameframe is elementary if and only if all of its instances are so, i.e. moveless games of zero depth where $Vr = \{\varnothing\}$.*
- ***General Gameframes:** We say that a gameframe is general if and only if all of its instances are so.*

As a result, an atomic formula $p(x_1, \ldots, x_n)$ is an elementary gameframe: we say that it is automatically won by $\mathcal{M}$ ($\top$) if the predicate is true in the interpretation, and lost otherwise ($\bot$).

Indeed, there are exactly two elementary games: $\top$ and $\bot$. On the other hand, $P(x_1, \ldots, x_n)$ is a general n-ary gameframe which depends on n distinct variables: this means that there are potentially infinite legal game runs (i.e. allowed lists of moves made by both $\mathcal{M}$ and $\mathcal{E}$ in a game play).

We now move on to define CL4 predicates and their validity conditions. Given *Lr* as the set of all **legal runs** of a game and $\langle\rangle$ as the empty game run, let $Wn : Lr \to \{\top, \bot\}$ be the content of a gameframe – i.e. the **winning function** that prints the winner out for each legal run of the gamestructure.

[4] We provide the informal semantics of CL4 logical operators in Section 2.1.2 below.

[5] On the other hand, **dynamic games** are those in which time and speed are of the essence: the player who is quick enough to make the first move becomes the winner. In such case, having a winning strategy becomes meaningless [22].

Definition 2.3 (CL4 Predicates)**.** *Given a content function Wn, we say that a* ***predicate*** (Vr, p) *is the elementary gameframe* (Vr, G) *such that, for any valuation* e*,* $Wn_e^G\langle\rangle = \top \iff e(p) = \top$*. Conversely, we say that every elementary gameframe* (Vr, G) *is the predicate* (Vr, p) *such that, for any valuation* e*,* $e(p) = \top \iff Wn_e^G\langle\rangle = \top$*.*

2.1.2. CL4 Game Operators

CL4 game operators are defined by the moves they authorise and the ownership of choices [25]. We are going to explain each of them in intuitive terms – which is sufficient for the purpose of this paper. Let A and B be elementary games, and $A(x)$ be a game depending on a variable x.

- **Negation (¬):** Intuitively, the game $\neg A$ is the negative image of A, meaning it is played exactly as A with a role-swap of $\mathcal{M}$ and $\mathcal{E}$. A win for $\mathcal{M}$ in A is a loss in $\neg A$. In CL4, negation can only be applied to elementary moveless games (thus turning $\bot$ to $\top$ and viceversa).
- **Parallel Conjunction** ($\wedge$): $\mathcal{M}$ wins $A \wedge B$ if it simultaneously and independently wins both games. This is essential for modelling the concurrent use of multiple resources or the need to satisfy multiple goals.
- **Parallel Disjunction** ($\vee$): In order to win $A \vee B$, $\mathcal{M}$ needs to win at least one of the components A, B which are being played simultaneously. In the simple case of $\neg A \vee A$, this can usually be done by adopting a Copycat Strategy – i.e. applying symmetric strategies to both games.
- **Parallel implication (→):** Classical implication is here defined through the parallel operators $\vee$ and $\wedge$ as $A \to B := \neg A \vee B$. Intuitively, this is a game where $\mathcal{M}$ plays B against $\mathcal{E}$, while simultaneously $\mathcal{E}$ plays A against $\mathcal{M}$ (swapped roles). This represents *security statement* and *resource conversion*: $\mathcal{M}$ can use information or resources gained from the antecedent A to solve the consequent B.
- **Universal Choice Quantifier (⨅):** This operator is the constructive version of the classical $\forall$. The game $\sqcap x A(x)$, being a long series of **choice conjunctions** $\sqcap$, begins with a move by $\mathcal{E}$, who selects a constant c from the domain. The game then proceeds as $A(c)$. This models adversary action – i.e. receiving an input or a challenge.
- **Existential Choice Quantifier (⨆):** This operator is the constructive version of the classical $\exists$. The game $\sqcup x A(x)$, i.e. a long series of **choice disjunctions** $\sqcup$, begins with a move by $\mathcal{M}$, who selects a constant c from the domain. The game then proceeds as $A(c)$. This models honest party action – i.e. computing an output or a response.
- **Blind Universal Quantifier (∀)**: This operator translates the classical $\forall$ into CL4 by defining it as a blind quantifier: playing $\forall x A(x)$ means playing $A(x)$ "blindly", i.e. without knowing the actual value of x. $\mathcal{M}$ wins $\forall x A(x)$ if and only if it can win $A(x)$ for every possible value of x from the domain.
- **Blind Existential Quantifier (∃)**: This operator translates the classical $\exists$ into CL4 by defining it as a blind quantifier: being the dual of $\forall$, $\mathcal{M}$ wins $\exists x A(x)$ if and only if it succeeds for at least one value of x.

2.2. Validity, Soundness, and their Cryptographic Significance

In the context of protocol verification, the logical concepts of *validity* and *soundness* are not merely abstract properties; they provide the bridge between the formal model and executable security guarantees, as we discuss in Section 3.1.

Definition 2.4 (Uniform validity, from [23])**.** *We say that a CL4-formula* F *is* ***uniformly valid*** *if and only if there is an algorithmic strategy* σ *that is independent of the specific interpretation* $*$ *of the non-logical symbols and according to which* $\mathcal{M}$ *solves* F^**. Such a* σ *is said to be a* ***logical solution*** *of* F*.*

Uniform validity is the semantic counterpart to *security against an arbitrary attacker*: it guarantees that a winning strategy exists in principle. *Soundness* guarantees that we can find it through formal reasoning.

Theorem 2.1 (Constructive soundness, from [23, 18])**.** *If a CL4-formula F is provable in CL4, then F is uniformly valid.*

The definition of provability in CL4 is the standard one: a formula F is provable in CL4 if it can be derived as the conclusion of a finite number of applications of CL4-inference rules.

In this constructive framework, such derivation constitutes the actual construction of the winning strategy: proving F guarantees the existence of a computable winning strategy $\mathcal{M}$ such that, for every possible opponent strategy $\mathcal{E}$, $\mathcal{M}$ wins the game defined by F.

3. Modelling the Needham–Schroeder Public Key Protocol

We formalise the protocol interactions as a game played over a domain of agents. We first establish the CoL framework for the modelling. Then we present the original Needham-Schroeder Public Key protocol (NS-PK). For the latter, we formally demonstrate a vulnerability by defining a winning strategy for the Environment ($\mathcal{E}$), representing a Man-in-the-Middle (MitM) attack that jeopardises the protocol's security. Subsequently, we model the Needham-Schroeder-Lowe protocol (NSL) as to prove that Lowe's fix makes the Copycat Strategy behind the MitM illicit. This allows us to define a winning strategy for the Machine ($\mathcal{M}$), guaranteeing security against any arbitrary Lowe-style adversarial environment ($\mathcal{E}$).

3.1. Security as Game Validity

In this framework, cryptographic properties are modelled as conditional games of the form:

$$\mathsf{Resources} \to \mathsf{SecurityGoal}$$

This implication asserts that if $\mathcal{M}$ is provided with the necessary Resources (e.g., valid keys, initial nonces), it can successfully satisfy the SecurityGoal (e.g., Authentication or Agreement) against any Environment.

Elementary games, such as $\mathsf{HasKey}(k_{Sx}, k_{Px})$ from Def. 3.3 below, have no moves; they represent static information or initial conditions provided at the start of the game. Consequently, the formal proof of security reduces to the mathematical verification that the protocol's algorithm constitutes a uniform solution for the game. More precisely, abstract protocol security is interpreted as *uniform validity* in the sense of Def. 2.4: the logical validity of the implicative game corresponds directly to the existence of a robust defence strategy for $\mathcal{M}$ that defeats any computational adversary.

The requirement that the strategy σ must succeed regardless of the interpretation of atoms means that the security of the protocol relies on its logical structure rather than on specific data values. In game-theoretic terms, if a protocol specification P is uniformly valid, it implies that $\mathcal{M}$ possesses a defence strategy that defeats any adversarial Environment $\mathcal{E}$.

Theorem 2.1 establishes the *constructive* nature of our framework. Unlike classical model checking, which often only verifies a static property of a pre-existing model, the soundness of CoL implies that a formal proof of a protocol's security is, in itself, a blueprint for secure implementation. If we can prove the security formula in CL4, the logic guarantees the existence of an algorithm (the protocol implementation) that correctly executes such security guarantee.

3.2. Cryptographic Primitives

Given $\mathcal{K}_P \subseteq \{0,1\}^n$ the set of public keys and $\mathcal{K}_S \subseteq \{0,1\}^n$ the set of secret keys, we define the key space $\mathcal{K} \subseteq \{0,1\}^n$ as the disjoint sum $\mathcal{K} := \mathcal{K}_P \uplus \mathcal{K}_S$. Then, let $\mathcal{C}$ be the space of valid ciphertexts $\mathcal{C} \subseteq \{0,1\}^*$ and $\mathcal{N}$ the nonce space $\mathcal{N} \subseteq \{0,1\}^n$.

In our model, $\mathcal{M}$ acts as the honest protocol (representing both an Initiator A and a Responder B), while $\mathcal{E}$ is the arbitrary opponent (the MitM intruder I) that tries to break the game.

Since we explicitly specify this task as the Resources $\rightarrow$ SecurityGoal game, the protocol model relies on the resources we formalise in the following subsections.

3.3. Atomic Predicates

The game logic is built upon the following atomic formulas (i.e. predicates). Being elementary games, these do not involve any moves; they evaluate to $\top$ or $\bot$ statically based on the values of their arguments.

Definition 3.1 (Protocol Predicates)**.** *We assume that the language we are working with is comprised of the following predicates:*

- $\mathsf{HasKeys}(k_{Sx}, k_{Px})$*: Given* $k_{Px} \in \mathcal{K}_P$, $k_{Sx} \in \mathcal{K}_S$ *and* $x \in \{A, B, I\}$*, it represents the fact that each player is equipped with its own secret/public key pairs* (k_{Sx}, k_{Px}) *and also has access to the others' public keys.*

- $\mathsf{Nonce}(N_x)$*: With* $N_x \in \mathbb{N}$ *and* $x \in \{A, B\}$*, it verifies that a given nonce is fresh. It becomes* $\top$ *if and only if* N_x *has never been used before. It is typically employed by* $\mathcal{M}$ *to check incoming messages.*

- $\mathsf{HasKeys}_N(A, B)$*: We define this predicate as the configuration of the game for* A, B *and a generic intruder* I*. It is valid if and only if all of them possess their respective key pairs (knowing everyone's public keys) and the honest agents can check each other's nonces. Formally:*

$$\mathsf{HasKeys}_N(A, B) \;:=\; \bigwedge_{x \in \{A,B,I\}} \mathsf{HasKeys}(k_{Sx}, k_{Px}) \wedge \mathsf{Nonce}(N_A) \wedge \mathsf{Nonce}(N_B).$$

  .

- $\mathsf{Enc}(N_x, P_x, k_{Px})$*: Used by both* $\mathcal{M}$ *and* $\mathcal{E}$ *to encrypt messages through the public key of the receiver. Once encrypted, the message becomes a ciphertext* $C_n \in \mathcal{C}$*, with* $n \in \mathbb{N}$*. A ciphertext is valid in our computation if and only if its plaintext* M_n *contains at least one nonce and potentially zero identifiers.*

- $\mathsf{Dec}(C_n, k_{Sx})$*: Used by the players to decrypt the incoming ciphertexts through their own personal secret key. The Unbreakable axiom from Def. 3.4 below avoids it from failing (i.e.* $\mathcal{E}$ *cannot forge or break any key): if that were to happen,* $\mathcal{M}$ *would win and the game would stop.*

- $\mathsf{ValidPair}(k_{Sx}, k_{Px})$*: This predicate is* $\top$ *if and only if* k_{Sx} *can uniquely reverse* k_{Px}*'s effect – i.e. they can seamlessly encrypt and decrypt the same ciphertexts.*

- Auth_A*: If* $\top$*, it means that* A *has completed all checks and has accepted* B*'s nonce.*

- Auth_B*: If* $\top$*, it means that* B *has completed all checks and has accepted* A*'s nonce.*

- $\mathsf{Secure}(A, B)$*: We define this predicate as the implication* $\mathsf{Auth}_B \rightarrow \mathsf{Auth}_A$*: indeed, if Responder has finished authentication, it means that Initiator must have honestly finished too.*

3.4. The Logic CL4$^{\mathsf{Crypt}}$ for Perfect Cryptography

We further extend the system CL4 to a CoL fragment – we named CL4^{Crypt} – dealing with minimal perfect cryptography assumptions, encoded by the following axioms.

Definition 3.2 (Axiom of Perfect Decryption)**.**

(*Dec*)
$$\sqcap k_{Px} \sqcap k_{Sx} \sqcap C_n \Big(\mathsf{ValidPair}(k_{Px}, k_{Sx}) \wedge \mathsf{Dec}(C_n, k_{Sx}) \rightarrow \sqcap k'_{Sx} (k_{S'x} \neq k_{Sx} \rightarrow \neg \mathsf{Dec}(C_n, k_{S'x})) \Big)$$

This axiom guarantees that C_n is a secure digital envelope. Indeed, it asserts that for any valid key pair (k_{Px}, k_{Sx}) and a ciphertext C_n encrypted with k_{Px}, there cannot be another $k_{S'x}$ different from k_{Sx} which can successfully decrypt the same ciphertext.

Definition 3.3 (Axiom of Perfect Encryption)**.**

$$\textit{(Enc)} \qquad \sqcap k_{Px} \sqcap C_n \Big(C_n = \mathsf{Enc}(M_n, k_{Px}) \rightarrow \neg \big(M_n = \mathsf{Dec}(C_n, k_{Sx}) \big) \Big)$$

This axiom guarantees secure authentication: the intruder cannot create a message that passes the decryption checks for a different party's key. Indeed, given a certain ciphertext C_n and its public encryption key k_{Px}, it asserts that a new plaintext cannot be forged to look as the authentic decryption of C_n without the proper k_{Sx} needed to decrypt it.

Together, these two axioms define the following one:

Definition 3.4 (Axiom of Perfect Cryptography Assumption)**.**

$$\textit{Unbreakable} \;:=\; (\textit{Dec} \wedge \textit{Enc})$$

We adopt the *Unbreakable* axiom as the logical placeholder for the computational security of the cryptographic primitives, isolating the protocol's vulnerability to be purely one of logic and interaction dynamics.

3.4.1. Mathematical Interpretation

The axiom *Dec* formalises the concept of *confidentiality* – i.e. one cannot read the contents of a message without having the specific private key. Indeed, it guarantees that the decryption process fails for any key other than the designated private counterpart, enforcing key exclusivity.

On the other hand, *Enc* formalises the concept of *integrity* – i.e. one cannot forge a valid ciphertext as if it were sent by another player by simply encrypting a counterfeit plaintext with the player's public key without knowing the same player's secret key needed to decrypt it. This axiom sets a constructibility constraint: a valid ciphertext cannot simply be guessed or generated by the adversary, which must explicitly possess the correct inputs to produce a matching output.

As a result, the axiom *Unbreakable* from Def. 3.4 formalises a power constraint on $\mathcal{E}$ to match real world computational hardness in perfect cryptography scenarios: it is a non-logical postulate that replaces the probabilistic asymptotic assumptions of cryptographic complexity theory. If we did not include it as a sort of game-theoretic shield, the security proof would instantly fail.

In our modelling of such a cryptographic assumption in CoL, we had to implement some precise formalisation options:

1. **Quantification**: We explicitly quantify over keys and ciphertexts using the *choice quantifiers* in order to express not only the existence but the fact that specific C_n and k_{Px}s are chosen (thus *known*) by our players. The universal scope of $\sqcap$ forces the security guarantee to hold uniformly over the entire key and ciphertext spaces. $\mathcal{E}$'s freedom to choose any key $k_{S'x}$ for decryption is constrained by *Dec*.

2. **Game Semantics**: The *Unbreakable* axiom becomes a game rule that the honest players can appeal to, guaranteeing a win against specific moves of $\mathcal{E}$. The first component *Dec* of the *parallel conjunction* ensures that any attempt by Environment of choosing a different key to recover a plaintext M_n automatically fails. The second one *Enc* forbids $\mathcal{E}$ from winning the forgery game: if it tries to generate a new ciphertext, Machine will automatically win.

3. **Symbolic Abstraction**: *Unbreakable* provides a clean symbolic interface. The protocol security proof we propose focuses on logical correctness and not on probabilistic reduction proofs; furthermore, predicates like $\mathsf{Dec}(C_n, k_{Sx})$ abstract complex security features into a single Boolean

test: the logical proof can proceed without modelling all the actual ways a cipher might fail if tampered with.[6]

Remark 3.5. *For our verification, the Unbreakable axiom operates as a logical proxy for the computational security of the cryptographic primitives, thereby isolating the protocol's vulnerabilities strictly to matters of logic and interaction dynamics.*

Remark 3.6. *Throughout our modelling and analysis, choice quantifiers application can have two different meanings: on one hand, Machine $\mathcal{M}$ is called upon to choose constants at each step of the game that contains $\sqcup$; on the other hand, Environment $\mathcal{E}$, given its ability to act arbitrarily, is also free to choose its own actions at each step of the game that contains $\sqcap$ (e.g. relaying a received message to the other party as is, without changing anything). Such a design aligns with common models in theoretical security analyses, where attackers are assumed to be (computationally) unbounded [26].*

3.5. The Game Flow

For the sake of completeness, before formalising the game in Computability Logic, we briefly review the original Needham-Schroeder Public Key protocol (NS-PK). The protocol aims to establish mutual authentication between an Initiator (A) and a Responder (B) using public key cryptography. We use the notation $\{M_n\}_{k_{Px}}$ to denote a plaintext M_n encrypted with agent x's public key. Appendix A provides an Alice-and-Bob rendering of protocol execution by Figure 1.

The correct execution of the protocol consists of three steps (assuming that the keys have already been shared with all players):

1. **Challenge**: A generates a fresh nonce N_A and sends it to B, along with the identifier, encrypted with B's public key.
$$A \to B : \{N_A, A\}_{k_{PB}}$$
2. **Response**: B decrypts the message to retrieve N_A and generates nonce N_B. Both nonces are then sent back to A, encrypted with k_{PA}: this proves to A that B successfully decrypted the first message.
$$B \to A : \{N_A, N_B\}_{k_{PA}}$$
3. **Completion**: A decrypts the message, verifies N_A and retrieves N_B; then returns it to B encrypted with k_{PB}. This proves to B that A successfully decrypted the second message.
$$A \to B : \{N_B\}_{k_{PB}}$$

At the end of this exchange, both parties possess the shared nonces and are mutually authenticated.

The vulnerability of the NS-PK protocol pointed out by Lowe arises when the honest participant A initiates a session with a dishonest intruder I – which intends to impersonate A to the second honest participant B. Indeed, I acts as a middle man trying to jeopardise the protocol's security. Appendix A provides an Alice-and-Bob rendering of the attack by Figure 2.

We briefly analyse this MitM attack in three logical steps, directly mapped to the CL4^{Crypt} setup:[7]

1. **Setup (Resources):** The secret and public key contexts are established for all players. Furthermore, $\mathcal{M}$ is able to verify the freshness of incoming nonces.

[6] We do not rule out the possibility to provide a more refined formalisation of such probabilistic constraints by considering a richer fragment of CoL including arithmetical reasoning, as we mention in Section 5.

[7] A formal, game-theoretic proof of the vulnerability is provided by our Theorem 4.1 below.

2. **Challenge and Response:** As usual, $\mathcal{E}$ acts with the aim of getting into $\mathcal{M}$'s way towards SecurityGoal (i.e. $\mathsf{Auth}_B \to \mathsf{Auth}_A$) by controlling the network channel. The game starts with A sending a ciphertext to intruder I.

$$A \to I : \{N_A, A\}_{k_{PI}}$$

Upon receiving and decrypting the message, $\mathcal{E}$ uses it to start a session with B keeping up the appearance of being A.

$$I \to B : \{N_A, A\}_{k_{PB}}$$

Indeed, I can choose to resend the ciphertext as is or change the encryption key to its desire. This way, $\mathcal{E}$ tries to insert itself as the middle man.

At this point, B, believing to be interacting with A, responds with the nonce pair encrypted with k_{PA}.

$$B \to I : \{N_A, N_B\}_{k_{PA}}$$

Consequently, I relays the ciphertext to A, unable to decrypt it.

$$I \to A : \{N_A, N_B\}_{k_{PA}}$$

A deciphers the message and believes it to be a valid response from I. Hence, A sends the received nonce back to I, which is now able to decrypt it.

$$A \to I : \{N_B\}_{k_{PI}}$$

3. **Completion:** The game comes to an end when the safety of the protocol is formally compromised, i.e. I successfully completes the attack.

$$I \to B : \{N_B\}_{k_{PB}}$$

Ultimately, B completes the protocol believing he has established a secure communication with A, which, on the other hand, is sure of interacting with I. This means that the security of the protocol has been breached, i.e. $\mathsf{Secure}(A, B) = \bot$. Indeed, Auth_B is $\top$, while Auth_A is $\bot$, making the implication $\mathsf{Auth}_B \to \mathsf{Auth}_A$ false.

Lowe's fix to prevent mis-authentication through identity tags will be modelled by introducing a new predicate for specifying the NSL protocol and show its uniform validity – i.e. safety in all possible game runs.

4. Game-Based Verification

We are now ready to show that in CL4^{Crypt} we can always find a strategy σ for $\mathcal{E}$ such that the original NS-PK protocol's security is compromised – i.e. $\mathcal{E}$ wins.

Theorem 4.1 (NS-PK Vulnerability)**.** *In CL4^{Crypt}, the following game is winnable by $\mathcal{E}$:*

$$\neg(\mathsf{HasKeys}_N(A, B) \to \mathsf{Secure}(A, B))$$

Proof. We explicitly construct a winning strategy σ for $\mathcal{E}$. Recall that in CL4, the logical implication $A \to B$ is defined as $\neg A \vee B$. Indeed, our proof shows that $\mathcal{E}$ can win the classically equivalent formula $\neg(\neg A) \wedge \neg B$, i.e. $\mathsf{HasKeys}_N(A, B) \wedge \neg\mathsf{Secure}(A, B)$ – meaning that the keys are honestly distributed and the protocol is, however, not safe.

Our goal is to prove that $\mathcal{E}$ can force A to accept a session with I.

1. **Assumption Phase (Antecedent)**: The game begins with the resource $\mathsf{HasKeys}_N(A, B)$ which, by definition, is equivalent to $\bigwedge_{x\in\{A,B,I\}} \mathsf{HasKeys}(k_{Sx}, k_{Px}) \wedge \mathsf{Nonce}(N_A) \wedge \mathsf{Nonce}(N_B)$. Since we are working with CL4^{Crypt}, we also have the *Unbreakable* axiom as one of our resources. Considering that if $\mathsf{HasKeys}_N(A, B)$ is false the game is won by $\mathcal{M}$, we assume that it is true (i.e. everyone knows their own secret/public keys and also each other's public ones).

2. **Start of the Protocol**:
   - A begins a session with I by choosing a valid nonce and encrypting a message with it:
   $$\sqcup N_A\ \mathsf{Nonce}(N_A) \wedge \sqcup C_0\ (C_0 = \mathsf{Enc}(N_A, P_A, k_{PI}))$$
   - I uses the ciphertext to initiate a session with B – all the while impersonating A. I chooses a new ciphertext C_1 to forward to the other party: the logical choice is to forward exactly the one received encrypted with B's public key in order to keep up the appearance of being A:
   $$\sqcap C_1(C_1 = \mathsf{Enc}(N_A, P_A, k_{PB}))$$
   - B responds to who believes to be A:
   $$\mathsf{Dec}(C_1, k_{SB}) \rightarrow \sqcup N_B\ \mathsf{Nonce}(N_B) \wedge \sqcup C_2\ (C_2 = \mathsf{Enc}(N_A, N_B, k_{PA}))$$
   Obviously, if decryption fails, $\mathcal{M}$ instantly wins.
   - I forwards the message to A, this time without changing the encryption key (since *Unbreakable* prevents $\mathcal{E}$ from decrypting C_2 and from forging a new ciphertext C_3):
   $$\sqcap C_3\ (C_3 = C_2)$$

3. **MitM Attack:**
   - A receives and responds to I (if the decryption fails, $\mathcal{M}$ instantly wins, since $\mathcal{E}$ breaks *Dec*):
   $$\mathsf{Dec}(C_3, k_{SA}) \rightarrow \sqcup C_4\ (C_4 = \mathsf{Enc}(N_B, K_{PI}))$$
   - I learns the session key N_B and uses it to complete the session with B:
   $$\sqcap C_5(C_5 = \mathsf{Enc}(N_B, k_{PB}))$$

4. **Final Resolution**: The attack successfully forces a final state where Auth_B is $\top$ (B accepts I as A) and Auth_A is $\bot$ (A fails to authenticate honest B). As a result, we have that $\mathsf{Auth}_B \rightarrow \mathsf{Auth}_A$ is false; since the antecedent $\mathsf{HasKeys}_N(A, B)$ is assumed true, $\mathcal{E}$ has successfully forced the condition $\mathsf{HasKeys}_N(A, B) \wedge \neg\mathsf{Secure}(A, B)$ to be true. This proves the vulnerability captured by the formal statement
   $$\neg(\mathsf{HasKeys}_N(A, B) \rightarrow \mathsf{Secure}(A, B)).$$

□

4.1. Verification of the Fix

We now move on to show how to verify the security fix implemented by the NSL protocol, i.e. the NS-PK protocol strengthened to overcome Lowe's attack. Appendix A provides an Alice-and-Bob rendering of the fixed protocol execution by Figure 3.

The MitM vulnerability is hereby tackled by the inclusion of the sender's identity in the exchange of the session key. Hence, we now introduce a new CL4 predicate which represents this solution.

Definition 4.1 (Identity check predicate)**.** $\mathsf{Check}(N_x, P_x)$ *is used by* $\mathcal{M}$ *to verify the freshness of the received nonce and the identity of the other protocol party. It takes into account nonce* N_x *and identifier* P_x*: if the nonce is not fresh, i.e. it has been already used in the past, the game fails and* $\mathcal{M}$ *wins. Same thing happens if the received* P_x *is different from the expected one – since Machine was able to maintain the safety of the protocol.*

We now show that there exists a strategy σ for $\mathcal{M}$ by which Machine always wins against any and every possible strategy of $\mathcal{E}$.

Theorem 4.2 (NSL Secure Authentication)**.** *In CL4*^{Crypt}*, the following game is winnable by* $\mathcal{M}$*:*

$$\mathsf{HasKeys}_N(A, B) \to \mathsf{Secure}(A, B).$$

Proof. We explicitly construct a winning strategy σ for $\mathcal{M}$. Recall that in CL4, the logical implication $A \to B$ is defined as $\neg A \vee B$: $\mathcal{M}$ then wins if it can win either the consequent B or the negation of the antecedent A.

1. **Assumption Phase (Antecedent):** The game begins with the resource $\mathsf{HasKeys}_N(A, B)$ which, by definition, is equivalent to $\bigwedge_{x\in\{A,B,I\}} \mathsf{HasKeys}(k_{Sx}, k_{Px}) \wedge \mathsf{Nonce}(N_A) \wedge \mathsf{Nonce}(N_B)$. Since we are working with CL4^{Crypt}, we also have the *Unbreakable* axiom as one of our resources. Considering that if $\mathsf{HasKeys}_N(A, B)$ is false the game is won by $\mathcal{M}$, we assume that it is true (i.e. everyone knows their own secret/public keys and also each other's public ones).

2. **Start of the Protocol**:
   - A begins a session with I by choosing a valid nonce and encrypting a message with it:
   $$\sqcup N_A\ \mathsf{Nonce}(N_A) \wedge \sqcup C_0\ (C_0 = \mathsf{Enc}(N_A, P_A, k_{PI}))$$
   - I uses the ciphertext to initiate a session with B – all the while impersonating A. I chooses a new ciphertext C_1 to forward to the other party: the logical choice is to forward exactly the one received encrypted with B's public key in order to keep up the appearance of being A:
   $$\sqcap C_1 (C_1 = \mathsf{Enc}(N_A, P_A, k_{PB}))$$
   - B responds to "A":
   $$\mathsf{Dec}(C_1, k_{SB}) \to \sqcup N_B\ \mathsf{Nonce}(N_B) \wedge \sqcup C_2\ (C_2 = \mathsf{Enc}(N_A, N_B, P_B, k_{PA}))$$
   Obviously, if decryption fails, $\mathcal{M}$ instantly wins for *Dec* axiom. As one can see, the addition of P_B in this step constitutes Lowe's fix, i.e. a protective shield that maintains the safety of the protocol.
   - I forwards the message to A, not knowing what its contents are (thanks to the *Unbreakable* axiom):
   $$\sqcap C_3\ (C_3 = C_2)$$

3. **Blocking the MitM Attack:**
   - A receives the ciphertext and decrypts it (if the decryption fails, $\mathcal{M}$ instantly wins for *Dec* axiom). At this point A checks for the freshness of N_B and the identity of the sender:
   $$\mathsf{Dec}(C_3, k_{SA}) \to \mathsf{Check}(N_B, P_B)$$
   Seeing that the received identity P_B is different from the expected P_I, and given that *Enc* guarantees authenticity, A instantly rejects the message and terminates the session.

4. **Final Resolution**: Since A has aborted, we have that Auth_A is $\bot$. B is waiting for I's final message to complete the session; however, since the protocol was terminated, I cannot complete the session with B because A will never send the final nonce back to B. Given that $(\bot \to \bot) = \top$, we have that $\mathsf{Auth}_B \to \mathsf{Auth}_A$ is true. This means that, despite the attack, the security property was not violated: $\mathcal{M}$ successfully prevents the intruder from creating a state where B finishes but A does not. Since the security predicate holds true, $\mathcal{M}$ wins the game, proving the theorem:

$$\mathsf{HasKeys}_N(A, B) \to \mathsf{Secure}(A, B).$$

□

5. Conclusion

We have shown that Computability Logic (CoL) offers a natural and rigorous framework for the formal verification of cryptographic protocols. We did so through a self-contained case study. By modelling the Needham-Schroeder Public Key protocol (NS-PK) as a game played against an adversarial Environment $\mathcal{E}$, we established a precise correspondence between the logical Copycat Strategy and the cryptographic Man-in-the-Middle (MitM) attack identified by Lowe.

Specifically, working with the sound and complete CL4 granted us the sufficient level of expressiveness and good-behaviour our analysis needed. Indeed, by distinguishing between elementary and general atoms, we were able to seamlessly integrate static cryptographic assumptions with dynamic protocol behaviours within a single logical derivation in an extended system we named CL4^{Crypt}. Furthermore, CL4 operators allowed us to express the constructive choice situation – i.e. the ability of not only knowing that a certain variable exists, but of being able to also *know exactly* which constant we are dealing with. In addition, establishing the uniform validity of the safety statement in CL4^{Crypt} provided a stronger guarantee than traditional finite-state model checking: it certified that the protocol's defence strategy is effective against any arbitrary adversary $\mathcal{E}$, provided they adhere to the fundamental cryptographic axioms.

Our analysis revealed that the vulnerability of the original NS-PK protocol arises as a structural symmetry in the model that allows $\mathcal{E}$ to mirror moves between parallel sessions – effectively validating the identification without possessing the requisite authentication credentials. This confirms a malicious application of the Copycat Strategy, which is a standard validity in CoL game semantics.

Conversely, our formalisation of the Needham-Schroeder-Lowe protocol (NSL) proves that including identity in the encrypted payload breaks this symmetry. In game-theoretic terms, this forces $\mathcal{E}$ to generate information it does not possess, rendering the Copycat Strategy behind the MitM illicit, and guaranteeing the existence of a winning strategy for the honest participants.

Crucially, by doing so, we shifted the verification perspective from the checking static trace properties to proving the existence of a dynamic defence. The validity of the security assertion in CoL implies that an algorithmic strategy exists which can defeat any Lowe-style adversary.

Future work. We would expand this methodology in some main directions. First, we aim to leverage the constructive nature of CoL to automate *strategy extraction*, enabling the synthesis of correct-by-construction executable code directly from verification proofs. This objective parallels the proofs-as-programs paradigm established in constructive type theories and computerised mathematics [27, 28, 29]. Second, we plan to extend our protocol analysis to *Zero-Knowledge Proofs*, utilizing CoL's game semantics to model information-hiding properties; this approach is intended to provide a computational complement to the standard logical analysis of knowledge dynamics [30, 31, 32]. Finally, we intend to investigate *resource-bound analyses*, exploring how CoL's inherent resource consciousness can be employed to detect and prevent algorithmic Denial-of-Service (DoS) attacks. This aligns with research on cost-based frameworks for protocol analysis [33] and the application of linear logic to resource-aware security policies and interactions [34]. Further general extensions of our CoL paradigm

would integrate probabilistic aspects by considering CoL-based arithmetical theories [35] and, possibly, dynamic games [22] to model more refined scenarios and protocol attacks.

Acknowledgments

We thank the two anonymous reviewers for their constructive feedback on the original manuscript. We also extend our gratitude to the attendees of our talk in Cagliari for their engaging discussion, with special thanks to Gabriele Costa and Marino Miculan for their stimulating questions and insightful comments.

This work was partially supported by the project SERICS – Security and Rights in the CyberSpace PE0000014, financed within PNRR, M4C2 I.1.3, funded by the European Union - NextGenerationEU (MUR Code: 2022CY2J5S). The International Research Network "Logic and Interaction" is also acknowledged.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] J. Blessing, D. Hugenroth, R. J. Anderson, A. R. Beresford, SoK: Web Authentication and Recovery in the Age of End-to-End Encryption, Proc. Priv. Enhancing Technol. 2025 (2025) 560–589. URL: https://doi.org/10.56553/popets-2025-0113. doi:10.56553/POPETS-2025-0113.

[2] OWASP Foundation, A02:2021 – Cryptographic Failures, 2021. https://owasp.org/Top10/A02_2021-Cryptographic_Failures/, Accessed: 2026-02-23.

[3] OWASP Foundation, A04:2025 - Cryptographic Failures, 2025. URL: https://owasp.org/Top10/2025/A04_2025-Cryptographic_Failures/, https://owasp.org/Top10/2025/A04_2025-Cryptographic_Failures/, Accessed: 2026-02-23.

[4] R. J. Anderson, Security engineering - a guide to building dependable distributed systems (3. ed.), Wiley, 2020.

[5] M. Abadi, R. M. Needham, Prudent engineering practice for cryptographic protocols, in: 1994 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, USA, May 16-18, 1994, IEEE Computer Society, 1994, pp. 122–136. URL: https://doi.org/10.1109/RISP.1994.296587. doi:10.1109/RISP.1994.296587.

[6] M. Abadi, R. M. Needham, Prudent Engineering Practice for Cryptographic Protocols, IEEE Trans. Software Eng. 22 (1996) 6–15. URL: https://doi.org/10.1109/32.481513. doi:10.1109/32.481513.

[7] R. M. Needham, M. D. Schroeder, Using encryption for authentication in large networks of computers, Commun. ACM 21 (1978) 993–999. URL: https://doi.org/10.1145/359657.359659. doi:10.1145/359657.359659.

[8] G. Lowe, Breaking and fixing the needham-schroeder public-key protocol using FDR, in: T. Margaria, B. Steffen (Eds.), Tools and Algorithms for Construction and Analysis of Systems, Second International Workshop, TACAS '96, Passau, Germany, March 27-29, 1996, Proceedings, volume 1055 of *Lecture Notes in Computer Science*, Springer, 1996, pp. 147–166. URL: https://doi.org/10.1007/3-540-61042-1_43. doi:10.1007/3-540-61042-1_43.

[9] G. Lowe, Breaking and fixing the needham-schroeder public-key protocol using FDR, Softw. Concepts Tools 17 (1996) 93–102.

[10] D. Dolev, A. C. Yao, On the security of public key protocols, IEEE Trans. Inf. Theory 29 (1983) 198–207. URL: https://doi.org/10.1109/TIT.1983.1056650. doi:10.1109/TIT.1983.1056650.

[11] F. J. Thayer, J. C. Herzog, J. D. Guttman, Strand Spaces: Why is a Security Protocol Correct?, in: Security and Privacy - 1998 IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 3-6, 1998, Proceedings, IEEE Computer Society, 1998, pp. 160–171. URL: https://doi.org/10.1109/SECPRI.1998.674832. doi:10.1109/SECPRI.1998.674832.

[12] M. Abadi, A. D. Gordon, A Calculus for Cryptographic Protocols: The spi Calculus, Inf. Comput. 148 (1999) 1–70. URL: https://doi.org/10.1006/inco.1998.2740. doi:10.1006/INCO.1998.2740.
[13] M. Burrows, M. Abadi, R. M. Needham, A Logic of Authentication, ACM Trans. Comput. Syst. 8 (1990) 18–36. URL: https://doi.org/10.1145/77648.77649. doi:10.1145/77648.77649.
[14] F. Bäumer, M. Brinkmann, J. Schwenk, Terrapin Attack: Breaking SSH Channel Integrity By Sequence Number Manipulation, in: D. Balzarotti, W. Xu (Eds.), 33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024, USENIX Association, 2024. URL: https://www.usenix.org/conference/usenixsecurity24/presentation/b%C3%A4umer.
[15] Microsoft Threat Intelligence Center, DEV-0537 Criminal Actor Targeting Organizations for Data Exfiltration and Destruction, Microsoft Security Blog, 2022. https://www.microsoft.com/en-us/security/blog/2022/03/22/dev-0537-criminal-actor-targeting-organizations-for-data-exfiltration-and-destruction/, Accessed: 2026-02-23.
[16] Cybersecurity and Infrastructure Security Agency, Countering Chinese State-Sponsored Actors Compromise of Networks Worldwide to Feed Global Espionage System, Official Release, 2025. https://www.cisa.gov/news-events/cybersecurity-advisories/aa25-239a, Accessed: 2026-02-23.
[17] T. H. Bakry, T. Mysk, Can a Tesla Stop Phishing and Social Engineering Attacks?, Mysk Inc. Security Research, 2024. https://www.mysk.blog/2024/03/10/tesla-phone-key/, Accessed: 2026-02-23.
[18] G. Japaridze, In the beginning was game semantics?, in: O. Majer, A. Pietarinen, T. Tulenheimo (Eds.), Games: Unifying Logic, Language, and Philosophy, volume 15 of *Logic, Epistemology, and the Unity of Science*, Springer, 2009, pp. 249–350. URL: https://doi.org/10.1007/978-1-4020-9374-6_11. doi:10.1007/978-1-4020-9374-6_11.
[19] G. Japaridze, Thoughts on sub-Turing interactive computability, 2024. arXiv:2411.01393, arXiV preprint 2411.01393, https://arxiv.org/abs/2411.01393.
[20] G. Japaridze, Fundamentals of Computability Logic, FLAP 7 (2020) 1115–1176. URL: http://collegepublications.co.uk/ifcolog/?00042.
[21] D. Gale, F. M. Stewart, Infinite games with perfect information, Contributions to the Theory of Games 2 (1953) 245–266.
[22] G. Japaridze, Introduction to computability logic, Ann. Pure Appl. Log. 123 (2003) 1–99. URL: https://doi.org/10.1016/S0168-0072(03)00023-X. doi:10.1016/S0168-0072(03)00023-X.
[23] G. Japaridze, From truth to computability II, Theoretical Computer Science 379 (2007) 20–52. URL: https://www.sciencedirect.com/science/article/pii/S0304397507000199. doi:https://doi.org/10.1016/j.tcs.2007.01.004.
[24] G. Japaridze, From truth to computability I, Theoretical Computer Science 357 (2006) 100–135. URL: https://www.sciencedirect.com/science/article/pii/S0304397506002660. doi:https://doi.org/10.1016/j.tcs.2006.03.014, clifford Lectures and the Mathematical Foundations of Programming Semantics.
[25] G. Japaridze, The logic of tasks, Annals of Pure and Applied Logic 117 (2002) 261–293. URL: https://www.sciencedirect.com/science/article/pii/S0168007201001233. doi:https://doi.org/10.1016/S0168-0072(01)00123-3.
[26] N. Ferguson, B. Schneier, Practical Cryptography, 1 ed., John Wiley & Sons, Inc., USA, 2003.
[27] A. Chlipala, Certified Programming with Dependent Types - A Pragmatic Introduction to the Coq Proof Assistant, MIT Press, 2013. URL: http://mitpress.mit.edu/books/certified-programming-dependent-types.
[28] M. H. Sørensen, P. Urzyczyn, Lectures on the Curry-Howard isomorphism, volume 149, Elsevier, 2006.
[29] M. Busi, R. Focardi, F. L. Luccio, Strands Rocq: Why is a Security Protocol Correct, Mechanically?, in: 38th IEEE Computer Security Foundations Symposium, CSF 2025, Santa Cruz, CA, USA, June 16-20, 2025, IEEE, 2025, pp. 33–48. URL: https://doi.org/10.1109/CSF64896.2025.00022. doi:10.1109/CSF64896.2025.00022.
[30] A. Baltag, L. S. Moss, Logics for Epistemic Programs, Synth. 139 (2004) 165–224. URL: https://doi.org/10.1023/B:SYNT.0000024912.56773.5e. doi:10.1023/B:SYNT.0000024912.56773.5E.

[31] G. Costa, C. Perini Brogi, Toward dynamic epistemic verification of zero-knowledge protocols, in: G. D'Angelo, F. L. Luccio, F. Palmieri (Eds.), Proceedings of the 8th Italian Conference on Cyber Security (ITASEC 2024), Salerno, Italy, April 8-12, 2024, volume 3731 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024. URL: https://ceur-ws.org/Vol-3731/paper25.pdf.

[32] A. Aldini, D. Fazio, P. Graziani, R. Mascella, M. Tagliaferri, A logical perspective on intending to keep a true secret, J. Log. Comput. 35 (2025). URL: https://doi.org/10.1093/logcom/exaf028. doi:10.1093/LOGCOM/EXAF028.

[33] C. Meadows, A Cost-Based Framework for Analysis of Denial of Service Networks, J. Comput. Secur. 9 (2001) 143–164. URL: https://doi.org/10.3233/jcs-2001-91-206. doi:10.3233/JCS-2001-91-206.

[34] L. Ceragioli, P. Degano, L. Galletta, L. Viganò, A Logic for Policy Based Resource Exchanges in Multiagent Systems, in: U. Endriss, F. S. Melo, K. Bach, A. J. B. Diz, J. M. Alonso-Moral, S. Barro, F. Heintz (Eds.), ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024), volume 392 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2024, pp. 1405–1412. URL: https://doi.org/10.3233/FAIA240641. doi:10.3233/FAIA240641.

[35] G. Japaridze, Build your own clarithmetic I, Log. Methods Comput. Sci. 12 (2016). URL: https://doi.org/10.2168/LMCS-12(3:8)2016. doi:10.2168/LMCS-12(3:8)2016.

A. Protocol, Attack, and Fix in Alice-Bob Notation

In the present section, we briefly recall the NS-PK standard interaction, together with the MitM attack, the NSL protocol and the failure of the attack on the latter. They are summarised according to the Alice-and-Bob notation by, respectively, Figure 1, Figure 2, Figure 3 and Figure 4 below.

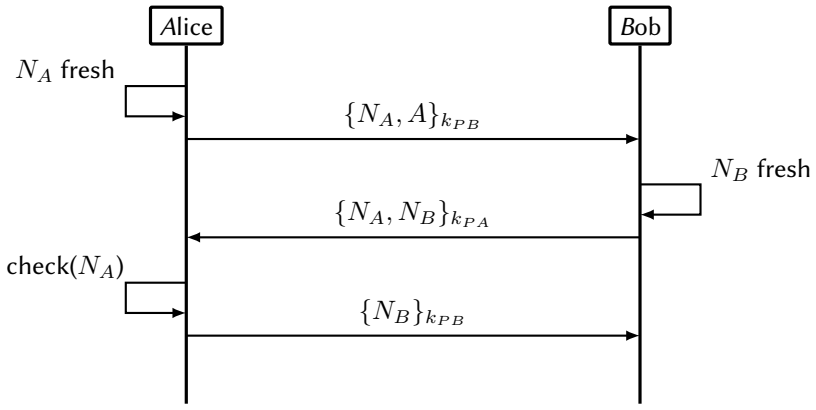

Figure 1: Interaction flow in the original Needham-Schroeder Public Key Protocol – Alice-Bob Notation.

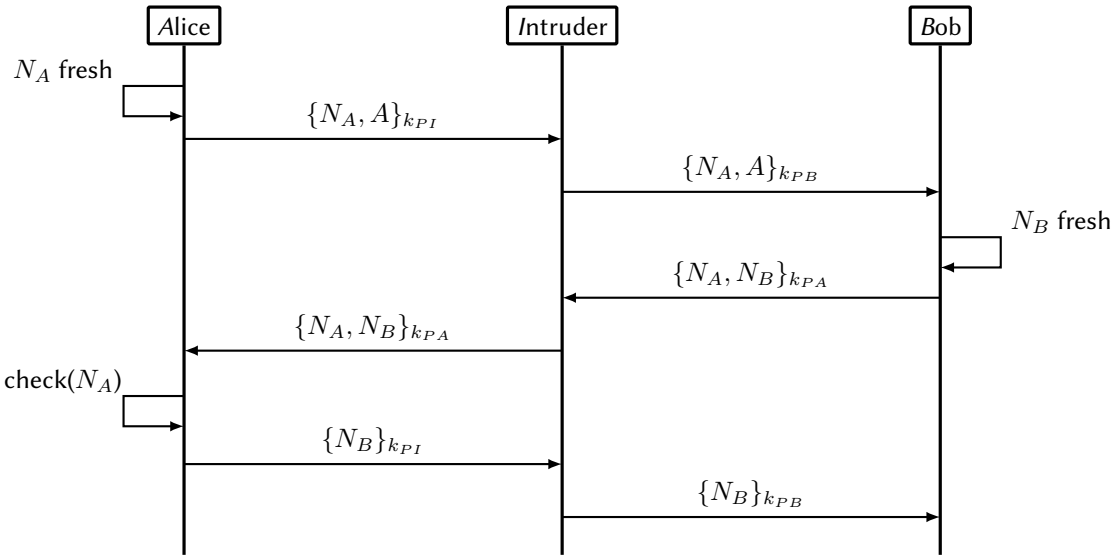

Figure 2: Lowe's Man-in-the-Middle attack on the original Needham-Schroeder Public Key Protocol – Alice-Bob Notation.

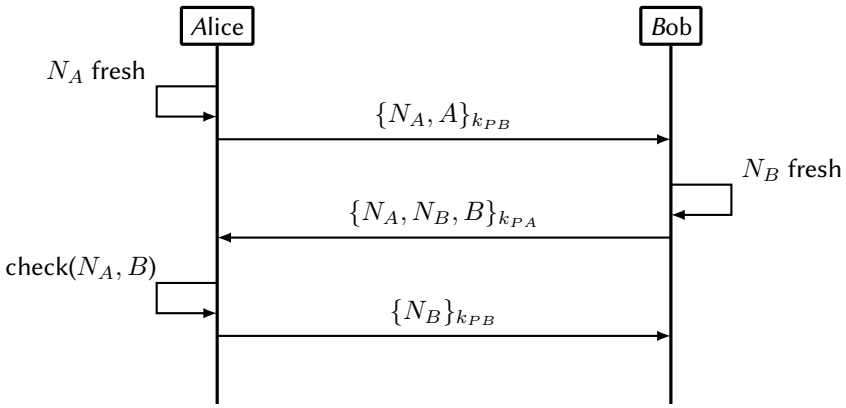

Figure 3: Interaction flow in the fixed Needham-Schroeder-Lowe protocol (NSL) –– Alice-Bob Notation.

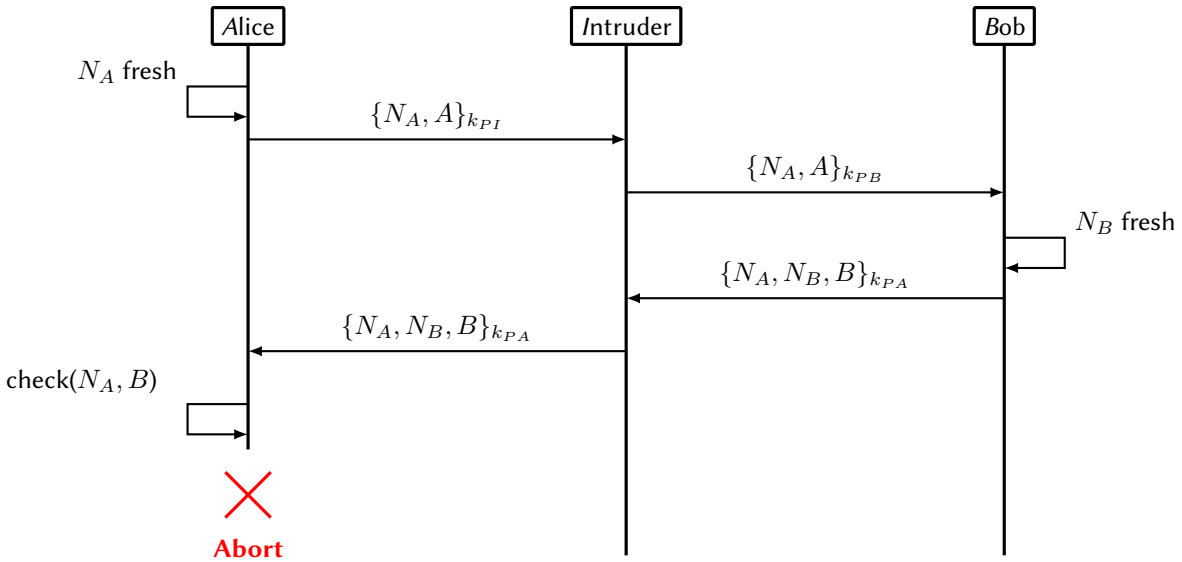

Figure 4: Lowe's Man-in-the-Middle attack on the fixed NSL protocol – Alice-Bob Notation.