A Complete Characterization of Secure Human-server Communication

A Complete Characterization of Secure Human-Server Communication

David Basin, Saša Radomirović, and Michael Schläpfer

Institute of Information Security,
Department of Computer Science,
ETH Zürich {david.basin, sasa.radomirovic, michael.schlaepfer}@inf.ethz.ch
http://www.infsec.ethz.ch

Abstract. Establishing a secure communication channel between two parties is a nontrivial problem, especially when one or both are humans. Unlike computers, humans cannot perform strong cryptographic operations without supporting technology, yet this technology may itself be compromised. We introduce a general communication topology model to facilitate the analysis of security protocols in this setting. We use it to completely characterize all topologies that allow secure communication between a human and a remote server via a compromised computer. These topologies are relevant for a variety of applications, including online banking and Internet voting. Our characterization can serve to guide the design of novel solutions for applications and it allows one to quickly exclude proposals that cannot possibly offer secure communication.

1 Introduction

Security-critical applications, such as online banking and Internet voting, rely on a secure communication channel between a human and a remote communication partner. These channels are constructed using security protocols that protect the messages exchanged between the human's personal computer and the remote system. However, unless the personal computer's hardware and software are trustworthy, information appearing on its screen may not faithfully represent the messages communicated with the remote system. Moreover, the personal computer may leak information to unauthorized third parties [10,20]. Securing the last few inches of the communication channel, namely between the network cable and the human, is difficult: in contrast to computing devices, most people's computing and memorizing abilities are insufficient to perform cryptographic computations. This problem is addressed by supporting technologies, ranging from simple code sheets [7] to smart cards and hand-held readers with integrated keypads and displays, commonly used for online banking [15].

How do we formally model a system where humans, computers, and supporting technologies interact? Most existing work focuses on particular scenarios, for instance on browser-based security protocols [12,13], login procedures [14], solutions for online banking [26], or Internet voting [22]. A general approach to modeling and reasoning about such systems are security ceremonies [9]. These

extend communication protocols to include human actors and communication means that are not considered in conventional security protocol models. Security ceremonies have not been formally defined, but they have inspired a variety of formal models with different focal points, which we discuss later in Section 5.

In this paper, we consider the setting of a distributed algorithm running on nodes communicating over links. We use traditional terminology and call such a distributed algorithm a protocol rather than a ceremony. We capture the abstraction of nodes communicating over links with a simple, intuitive, graph-theoretic model that we call a *communication topology*. We model the protocol execution for a given topology as a multiset term rewriting system. Our approach differs from existing approaches in that it largely ignores the interpretation of what nodes and links are and it focuses instead on their capabilities and security properties. The result is a simple and useful model with applications both to protocol verification and to establishing impossibility results.

Contributions. We introduce a communication topology model on top of an operational semantics for security protocols. Our topology model formalizes the environment in which protocols are executed and allows one to reason about communication systems at different levels of abstraction. We use the model to completely characterize necessary and sufficient conditions for the existence of security protocols that provide secure channels between a human and a remote server using an insecure network and a dishonest platform. Necessary conditions are established by impossibility results and sufficient conditions are proved constructively by providing protocols.

Our characterization is relevant for practical applications such as online banking and Internet voting. It allows one to quickly assess whether a particular protocol design and supporting technology can plausibly offer secure communication. The characterization can be used to guide the design of novel solutions for establishing secure channels between humans and a remote server and we provide examples that illustrate this.

Organization. We introduce our communication topology model in Section 2 and the underlying security protocol model in Section 3. We characterize secure human-server communication in Section 4. We discuss related work in Section 5, and draw conclusions in Section 6. In the Appendices we give full details on our model and proofs.

2 Communication Topology Model

We first define a general communication topology model that formalizes assumptions relative to which a communication protocol's security properties are analyzed. Every node in the topology corresponds to a unique role in the protocol which specifies the node's behavior. The topology specifies the node's capabilities, initial knowledge, honesty, and available communication channels. Afterwards we restrict our focus to a particular class of topologies that is relevant for protocols where a human securely communicates with a remote server using a potentially compromised computer.

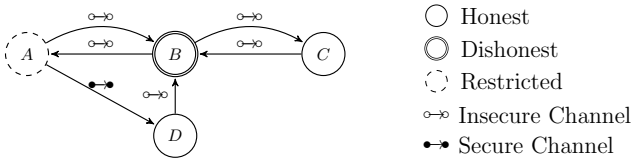


Fig. 1: Communication topology example.

2.1 General Communication Topology Model

A *communication topology* (relative to a signature Σ) is an edge- and vertex-labeled directed graph (V, E, η, μ), where V is the set of vertices, $E \subseteq V \times V$, and η and μ are functions assigning labels to vertices and edges respectively. We call a sequence of vertices $[v_1, \ldots, v_{k+1}] \in V^*$, such that $(v_i, v_{i+1}) \in E$ for $1 \leq i \leq k$, a *path from* v_1 *to* v_{k+1} *of length* k or simply a *path*. The path is *acyclic* if $v_i \neq v_j$ for all $1 \leq i < j \leq k+1$. We denote the transitive closure of E by E^+, i.e., we write $(v_i, v_j) \in E^+$ if there is a path from v_i to v_j.

The set of vertices V represents a protocol's roles. For $A, B \in V$, an edge $(A, B) \in E$ denotes the existence of a link from a node representing role A to the node representing role B. The vertex labeling function $\eta : V \to \mathit{NodeProp}$ assigns capability and trust assumptions to role names. It indicates, for instance, whether a role is assumed to be executed by a human and whether the executing agent is assumed to be honest. The edge labeling function $\mu : E \to \mathit{LinkProp}$ assigns channel assumptions to links, for example, whether channels are insecure, authentic, or confidential. The sets *NodeProp* and *LinkProp* are specified in Section 3, where our formal protocol model is defined.

Graphical representation. We graphically represent a communication topology (V, E, η, μ) as follows. Vertices $A \in V$ are drawn as simple, concentric, or dashed circles depending on the labeling η. To express that a role $A \in V$ is assumed to be executed by a dishonest agent, we draw concentric circles. A dashed circle indicates that an honest agent executing the role A has restricted capabilities. Note that our vertex representation does not distinguish between different types of restricted capabilities and knowledge. This limitation suffices for the present paper, since humans are the only agents with restricted capabilities. Edges $e \in E$ are drawn as arrows connecting the circles and are labeled according to μ. The edge labels are written next to the arrows representing the corresponding edges.

Figure 1 shows a communication topology (V, E, η, μ), with $V = \{A, B, C, D\}$. In this example, the role A is assumed to be executed by an honest restricted agent, and role B is assumed to be executed by a dishonest agent. The remaining roles are assumed to be executed by honest, unrestricted agents. The set of edges E and their labeling can be read off of Figure 1. For example, $(A, B) \in E$, $(A, C) \notin E$, and $(A, C) \in E^+$. The link from A to D is secure ($\bullet\!\!\rightarrow\!\!\bullet$) and all other links are insecure ($\circ\!\!\rightarrow\!\!\circ$).

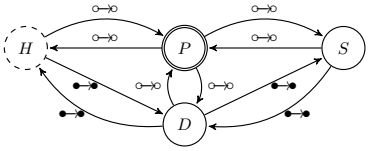


Fig. 2: The supergraph of all HISP topologies.

2.2 Human-Interaction Security Protocols

We now introduce the class of security protocols where humans intend to securely communicate with a remote server. We make the following assumptions regarding humans' capabilities.

Assumption 1 *Humans may send, receive, compare, concatenate (pair) and select (project) terms. They may generate random (fresh) values. No restriction are imposed on human memory.*[1]

Thus, humans are assumed to be able to remember all terms received on any channel and to output any term constructible from their knowledge using pairing and projection on any other channel. However, they cannot perform cryptographic operations without supporting technology.

To motivate the communication topology for human-interaction security protocols, consider protocols that provide a secure communication channel between a human and a server. We can model such protocols' communication topology by defining two nodes, a human H and server S connected by a secure channel. However, this is too abstract to reason about the requirements a protocol must satisfy to provide a secure channel from the human to the server. A natural step in making this model more concrete is to assume that the human cannot directly communicate with the remote server and must instead use a computing platform P that communicates with the server over an insecure network. The resulting refined topology consists of a channel between H and P instead of H and S and an insecure channel between P and S. If we assume that the computing platform P is honest, then this topology represents the well-known problem of establishing a secure communication channel between two agents over an insecure network.

Our focus is on the case where the computing platform is dishonest, i.e., compromised. Achieving secure communication generally requires that the human has access to a trusted device D and we model this by including D in the topology. Examples of such devices are a list of one-time passwords, a code sheet, or a smart card with a corresponding card reader. Protocols that establish secure communication between the human and a remote server under these circumstances are highly relevant in practice, for example in online banking and

[1] Our possibility results show that the human never needs to remember more than three terms plus the names of communication partners. However, not posing limits on human memory merely strengthens our impossibility results.

Internet voting. We call such a protocol a *Human-Interaction Security Protocol*, or *HISP* for short, and the corresponding communication topology a *HISP topology*.

A HISP topology (formally defined in Section 3.2) consists of a human H, a server S, and a device D, which are assumed to be honest, and a computing platform P, which is assumed to be dishonest. There are no restrictions on the capabilities or initial knowledge of S, D, and P. H is restricted as stated in Assumption 1. Figure 2 shows the supergraph of all HISP topologies (V, E, η, μ) and indicates the edge labels. Since the edge labels are constant, we omit them in graphical representations of HISP topologies in the remainder of this paper. Examples of such representations are shown in Theorem 1.

3 Security Protocol Model

In this section we describe our security protocol model which constitutes the formal underpinning of our communication topology model. Our model is based on Tamarin's [23] security protocol model, which we call the *Tamarin model* throughout this paper. We summarize its main features and several extensions that we made to support HISPs, such as the notion of communicating knowledge. We provide full details in Appendix A. Note that although our extensions are substantial, the Tamarin tool, which performs deduction based on term rewriting, can still be directly applied to analyze our protocol models.[2]

3.1 Background

Notation. We denote the set of sequences of elements from a set S by S^*. For the sequence s, $|s|$ denotes the length of s and we write s_i to refer to the i-th element of s. A sequence s with $|s| = k$ is denoted by $[s_1, \ldots, s_k]$ and the empty sequence by $[\,]$. We denote the concatenation of two sequences s and s' by $s \cdot s'$. $\mathcal{P}(S)$ denotes the powerset of S.

We use the term algebra of the Tamarin model. The term algebra is denoted by $\mathcal{T}$, its underlying signature by Σ, and the set of ground terms by $\mathcal{M}$. The signature Σ contains functions $\langle \_, \_ \rangle$ for pairing, $\mathrm{senc}(\_, \_)$ and $\mathrm{sdec}(\_, \_)$ for symmetric encryption and decryption, $\mathrm{aenc}(\_, \_)$ and $\mathrm{adec}(\_, \_)$ for asymmetric encryption and decryption, $\mathrm{sign}(\_, \_)$ and $\mathrm{verify}(\_, \_, \_)$ for signing messages and verifying signatures, $\pi_1(\_)$ and $\pi_2(\_)$ for the first and second projection of a pair of terms, $\mathrm{h}(\_)$ for hashing terms, and $\mathrm{pk}(\_)$ to represent the public key corresponding to a given secret key. The function $\mathrm{pk}(\_)$ can be applied to any term t to yield the term $\mathrm{pk}(t)$, but t cannot be inferred from $\mathrm{pk}(t)$. Σ contains the two countably infinite, disjoint sets of fresh and public constants, denoted by $\mathcal{C}_{\mathrm{fresh}}$ and $\mathcal{C}_{\mathsf{pub}}$, respectively. Fresh constants model the generation of nonces, while public terms represent agent names and other publicly known values.

[2] In particular, our results are proven for unbounded numbers of interleaved protocol sessions and remain true for all equational theories supported by Tamarin that include the standard theory used here.

Multiset term rewriting system. We use a labeled multiset term rewriting system to represent all possible protocol behaviors. The system states are represented as finite multisets of *facts*. Facts are functions over $\mathcal{T}$ whose symbols appear in a signature Σ_{Fact} (disjoint from Σ), which is partitioned into *linear* and *persistent* fact symbols. The set of facts is denoted by $\mathcal{F}$ and the set of all ground facts, i.e., facts $\mathsf{F}(t_1, \ldots, t_k)$ such that $\mathsf{F} \in \Sigma_{Fact}$ and $t_i \in \mathcal{M}$ for all $1 \leq i \leq k$, is denoted by $\mathcal{G}$. Linear facts model resources that can only be consumed once. Persistent facts, prefixed by "!", model inexhaustible resources.

State transitions are effected by labeled multiset rewriting rules. Each such rule is denoted by $l -[\, a \,]\!\!\rightarrow r$ with $l, a, r \in \mathcal{F}^*$. The elements in l, a, r are called the rule's premises, actions, and conclusions, respectively. The transition rewrites the current state by replacing the linear facts in l with the facts in r and is labeled with the facts in a. The initial system state is the empty multiset.

A trace tr is a finite sequence of sets of actions $tr_i \in \mathcal{P}(\mathcal{G})$, for $1 \leq i \leq |tr|$. The action sets in the trace label the system's state transitions that correspond to applying a ground instance of a rule in a set $\mathcal{R}$. We write $a \in tr$ if $a \in tr_i$ for some $1 \leq i \leq |tr|$, that is, when the action a occurs in a set of ground actions in the trace tr. We denote the set of all traces for the set of rules $\mathcal{R}$ by $TR(\mathcal{R})$ and refer to $\mathcal{R}$ as a *protocol*.

We distinguish between model rules and protocol specification rules in $\mathcal{R}$, denoted by $\mathcal{R}_{Model}$ and $\mathcal{R}_{Spec}$ respectively, where $\mathcal{R}_{Model} \cap \mathcal{R}_{Spec} = \emptyset$ and $\mathcal{R} = \mathcal{R}_{Model} \cup \mathcal{R}_{Spec}$. The former are a fixed set of message deduction rules modeling a standard Dolev-Yao adversary [8] and our model extensions described in Section 3.2 below. The latter rules model a given protocol specification and are described in Section 3.3 and Appendix A.3.

The Tamarin rules modeling a Dolev-Yao adversary are implemented with three facts. The adversary learns all terms in Out facts and injects messages from his knowledge using In facts. Terms learned by the adversary are stored as persistent $\mathsf{!K}$ facts which represent the adversary's knowledge.

3.2 Model Extensions

To connect the communication topology to the underlying security protocols model, we need to define the node and link properties, i.e. the sets *NodeProp* and *LinkProp* introduced in Section 2.1, in the Tamarin model.

Node Properties. Every node in a communication topology (V, E, η, μ) is assigned capability and trust assumptions by the vertex labeling function $\eta : V \rightarrow \mathit{NodeProp}$. We let $\mathit{NodeProp} = \mathcal{P}(\Sigma) \times \mathcal{P}(\mathcal{T}) \times \{\mathsf{honest}, \mathsf{dishonest}\}$, where $\mathcal{P}(S)$ denotes the powerset of the set S. An agent's capabilities are defined by its computational abilities and initial knowledge. The computational capability assumption is specified by a subset of Σ consisting of the function symbols available to the agent executing the role that is represented by the node. The initial knowledge assumption is specified as a subset of $\mathcal{T}$. It indicates the *maximal* initial knowledge an agent is allowed to have. An empty set formalizes that the

$$\mathcal{DA} := \{\ [\mathsf{AgentState}(A, c, n)] -[\ \mathsf{Dishonest}(A)\ ]\rightarrow [\mathsf{Out}(\langle A, c, n\rangle)]\,, \quad (1)$$

$$[\mathsf{In}(\langle c', n'\rangle)] -[\ \mathsf{Dishonest}(A)\ ]\rightarrow [\mathsf{AgentState}(A, c', n')]\,, \quad (2)$$

$$[\mathsf{In}(x')] -[\ \mathsf{Dishonest}(A)\ ]\rightarrow [\mathsf{Fresh}(A, x')]\ \} \quad (3)$$

Fig. 3: Dishonest agent rules.

agent has no initial knowledge, while $\mathcal{T}$ states that no restrictions are placed on the agent's initial knowledge other than that it is a finite set.[3] The elements in $\{\mathsf{honest}, \mathsf{dishonest}\}$ indicate the trust assumptions associated with a role. Agents marked $\mathsf{dishonest}$ are assumed to be controlled by the adversary whereas those marked honest are assumed to faithfully execute the security protocol.

We model agents explicitly with $\mathsf{AgentState}(A, c, n)$ facts, where A is a public term representing an agent's name, c refers to the role step the agent is in, and n is the agent's knowledge at that step. The set of agents appearing in a trace tr, denoted by $\mathit{Agents}(tr)$, is the set of all public constants A such that $\mathsf{Agent}(A)$, $\mathsf{Honest}(A)$, or $\mathsf{Dishonest}(A)$ appears in tr. The subset of honest agents, denoted by $\mathit{Honest}(tr)$, is the set of all agents A such that $\mathsf{Dishonest}(A)$ does not appear in tr. We model dishonest agents with the $\mathcal{DA}$ rules shown in Figure 3. These agents are marked with a $\mathsf{Dishonest}$ action. By Rule (1) a dishonest agent may leak all information in its state to the adversary. Rule (2) models the adversary's capability to arbitrarily modify a dishonest agent's internal state and Rule (3) models that a dishonest agent's fresh constants may be chosen by the adversary.

Link Properties. Every link in a communication topology (V, E, η, μ) is assigned a channel property, representing an assumption on the link's behavior, by the edge labeling function $\mu : E \rightarrow \mathit{LinkProp}$. We define four channel properties and set $\mathit{LinkProp} = \{\circ\!\!\rightarrow\!\!\circ, \bullet\!\!\rightarrow\!\!\circ, \circ\!\!\rightarrow\!\!\bullet, \bullet\!\!\rightarrow\!\!\bullet\}$, where the four symbols denote the properties for insecure, authentic, confidential, and secure channels, respectively. This notation is adapted from Maurer and Schmid's channel calculus [16].

The insecure channel $\circ\!\!\rightarrow\!\!\circ$ is the standard communication channel between protocol agents in a Dolev-Yao model. We extend the Dolev-Yao message deduction rules of the Tamarin model that pertain to insecure channels with a set of channel rules, $\mathcal{CH}$, for confidential, authentic, and secure channels. The set $\mathcal{CH}$ models how protocol agents access insecure, authentic, confidential, and secure (i.e., authentic and confidential) channels and is shown in Figure 4. Rules (4) and (5) represent insecure channels. The sending of messages over an insecure channel is labeled with the $\mathsf{Snd}_{\mathsf{I}}$ action and produces an Out fact, which represents the adversary's capability to learn messages by eavesdropping. Rule (5) is annotated with the $\mathsf{Rcv}_{\mathsf{I}}$ action and represents the adversary's capability to in-

[3] This finite initial knowledge requirement is without loss of generality, because the initial knowledge set is not required to be closed under term inference. This is a simple way to prevent that an agent's initial knowledge contains all fresh constants.

$$\mathcal{CH} := \{\ [\mathsf{Snd_I}(A,B,m)] -\![\ \mathsf{Snd_I}(A,B,m)\ ]\!\rightarrow [\mathsf{Out}(\langle A,B,m\rangle)]\,, \tag{4}$$
$$[\mathsf{In}(\langle A,B,m\rangle)] -\![\ \mathsf{Rcv_I}(A,B,m)\ ]\!\rightarrow [\mathsf{Rcv_I}(A,B,m)]\,, \tag{5}$$
$$[\mathsf{Snd_A}(A,B,m)] -\![\ \mathsf{Snd_A}(A,B,m)\ ]\!\rightarrow [!\mathsf{Auth}(A,m), \mathsf{Out}(\langle A,B,m\rangle)]\,, \tag{6}$$
$$[!\mathsf{Auth}(A,m), \mathsf{In}(B)] -\![\ \mathsf{Rcv_A}(A,B,m)\ ]\!\rightarrow [\mathsf{Rcv_A}(A,B,m)]\,, \tag{7}$$
$$[\mathsf{Snd_C}(A,B,m)] -\![\ \mathsf{Snd_C}(A,B,m)\ ]\!\rightarrow [!\mathsf{Conf}(B,m)]\,, \tag{8}$$
$$[!\mathsf{Conf}(B,m), \mathsf{In}(A)] -\![\ \mathsf{Rcv_C}(A,B,m)\ ]\!\rightarrow [\mathsf{Rcv_C}(A,B,m)]\,, \tag{9}$$
$$[\mathsf{In}(\langle A,B,m\rangle)] -\![\ \mathsf{Rcv_C}(A,B,m)\ ]\!\rightarrow [\mathsf{Rcv_C}(A,B,m)]\,, \tag{10}$$
$$[\mathsf{Snd_S}(A,B,m)] -\![\ \mathsf{Snd_S}(A,B,m)\ ]\!\rightarrow [!\mathsf{Sec}(A,B,m)]\,, \tag{11}$$
$$[!\mathsf{Sec}(A,B,m)] -\![\ \mathsf{Rcv_S}(A,B,m)\ ]\!\rightarrow [\mathsf{Rcv_S}(A,B,m)]\ \} \tag{12}$$

Fig. 4: Channel rules.

sert arbitrary messages into insecure channels whenever a protocol agent intends to receive a message from an insecure channel (In).

The authentic channel $\bullet\!\!\rightarrow\!\!\circ$ allows the adversary to learn messages sent on the channel, but prevents the adversary from modifying the message or its sender. The adversary may, however, replay transmitted messages on this channel. Rules (6) and (7) model authentic channels. In Rule (6), the adversary learns the message (Out). The auxiliary !Auth fact ensures that in Rule (7) the adversary can neither alter the message nor its sender. The !Auth fact is persistent, which reflects the adversary's capability to replay authentically transmitted messages. The rules are annotated with the corresponding $\mathsf{Snd_A}$ and $\mathsf{Rcv_A}$ actions.

The confidential channel $\circ\!\!\rightarrow\!\!\bullet$ does not allow the adversary to learn the message sent on the channel, but allows the adversary to modify the sender and to repeatedly deliver (replay) the message on the confidential channel. The adversary can also deliver an arbitrary message from his knowledge (faking an arbitrary sender) on the confidential channel. Confidential channels are modeled using Rules (8)–(10). Rule (8) creates an auxiliary !Conf fact and the adversary does not learn the message. Rule (9) represents the case where the adversary passes the (unknown) confidential message m to the intended recipient, possibly pretending that it stems from another sender (In). The !Conf fact is persistent, which reflects the adversary's capability to replay confidentially transmitted messages. Rule (10) represents the adversary's capability to access the confidential channel to deliver any message from his knowledge.

Finally, for the secure channel $\bullet\!\!\rightarrow\!\!\bullet$, the adversary neither learns the message sent on it, nor can he change the sender, receiver, or transmitted message, but he may repeatedly deliver it. Rules (11) and (12) model secure channels. In Rule (11), the adversary learns nothing and an auxiliary !Sec fact is generated, which models that the adversary can neither alter the message nor its sender. Rule (12) models receiving a message from a secure channel. The !Sec fact is persistent, allowing the adversary to replay securely transmitted messages.

The protocol rules for the above channels are labeled with send and receive actions that indicate the type of channel used, the sender, receiver, and message.

This means that in a protocol execution, the application of a rule that sends a message on, e.g., the authentic channel $\bullet\!\!\rightarrow\!\!\circ$ is labeled with a $\mathsf{Snd}_{\mathsf{A}}(A, B, m)$ action, where A is the agent sending the message m and B is the intended recipient. The reception of a message on the confidential channel $\circ\!\!\rightarrow\!\!\bullet$ is labeled with a $\mathsf{Rcv}_{\mathsf{C}}(A, B, m)$ action, where A is the apparent sender of the message m and B the recipient. The send and receive actions for the insecure and secure channels are $\mathsf{Snd}_{\mathsf{I}}$, $\mathsf{Rcv}_{\mathsf{I}}$ and $\mathsf{Snd}_{\mathsf{S}}$, $\mathsf{Rcv}_{\mathsf{S}}$, respectively. Thus every message sent or received by an agent is logged with a corresponding action in the trace.

HISP topology. We can now formally define the HISP topology.

Definition 1. *A* HISP topology *is a communication topology* (V, E, η, μ)*, where the set of nodes is* $V = \{H, D, S, P\}$ *and the set of links is* $E \subseteq \{(a, b) \in V \times V \mid a \neq b \wedge (a, b) \neq (H, S) \wedge (a, b) \neq (S, H)\}$*. The vertex labels are defined by* $\eta(H) = (\Sigma_H, \mathcal{T}, \mathsf{honest})$*,* $\eta(D) = (\Sigma, \mathcal{T}, \mathsf{honest})$*,* $\eta(S) = (\Sigma, \mathcal{T}, \mathsf{honest})$*, and* $\eta(P) = (\Sigma, \mathcal{T}, \mathsf{dishonest})$*, where* $\Sigma_H = \{\langle \_, \_ \rangle, \pi_1(\_), \pi_2(\_)\} \cup \mathcal{C}_{\mathsf{pub}} \cup \mathcal{C}_{\mathrm{fresh}}$*. The edge labels are* $\mu(e) = \circ\!\!\rightarrow\!\!\circ$*, for* $e \in E_P$*, and* $\mu(e) = \bullet\!\!\rightarrow\!\!\bullet$*, for* $e \in E \setminus E_P$*, where* $E_P = \{(a, b) \in E \mid a = P \vee b = P\}$*.*

3.3 Channels as Goals

In the preceding section, we defined communication channels as a means for agents to communicate. Here we define the notion of a communication channel as a protocol goal. This provides us with a formal meaning for statements asserting the existence or non-existence of protocols providing secure channels in HISP topologies. The alignment of the semantics of our HISP model with the semantics of Tamarin is particularly significant here because it allows us to give manual proofs of impossibility results and use Tamarin to obtain automatic proofs of possibility results in the same protocol model.

Our use of channels as goals has three aspects we highlight here. First, we consider the *communication of knowledge* rather than the transmission of messages over a network. We formally define this concept in Definition 2 and illustrate its application thereafter. Second, to avoid protocols that trivially satisfy security properties by never communicating a useful message, we require that there exists a trace in which security-relevant knowledge is communicated from one honest agent to another. We therefore define the notion of *providing a communication channel*. Finally, we consider as a special case protocols in which a fresh constant generated by the sender can be communicated. We use this as a coarse, but for our purposes sufficient, way to differentiate between protocols that allow for the communication of an arbitrary message and protocols that impose limits on the communicated message, such as that it be a yes/no vote.

We define what it means for knowledge to be communicated as follows. We say that an agent S communicates a message m in a trace, if the action $\mathsf{Comm}(S, m)$ appears in the trace. This merely implies that S knows m, but there is no guarantee that m is sent on the network. We say that an agent R

learns a message m in a trace, if $\mathsf{Learn}(R, m)$ appears in the trace. This too implies that R knows m, but there is no guarantee that R did not know m earlier in the trace. To say that m is communicated from S to R in a trace means that $\mathsf{Comm}(S, m)$ occurs before $\mathsf{Learn}(R, m)$ in the trace. In other words, the agents S and R know m and S performs a protocol step labeled $\mathsf{Comm}(S, m)$ before R performs a protocol step labeled $\mathsf{Learn}(R, m)$.

Definition 2. *A message $m \in \mathcal{M}$ is said to be* communicated *from an agent S to an agent R in a trace tr, denoted* communicate(tr, S, R, m)*, if*

$$\exists\, tr', tr'' \in \mathcal{P}(\mathcal{G})^* : tr = tr' \cdot tr'' \wedge \mathsf{Comm}(S, m) \in tr' \wedge \mathsf{Learn}(R, m) \in tr''.$$

Communicating a message from an agent S to an agent R is more general than transmitting a message from S to R. If R receives a message m from S, then S has communicated m to R. However, a message can be communicated without being sent, as the next example shows.

Example 1. Consider a code sheet shared between S and R that consists of pairs of distinct fresh constants. If (x, y) is such a pair, then S can communicate x to R by sending y.

A protocol where the sender communicates a message by sending its code limits the sender's communication channel to messages on the code sheet. This is useful for applications like code voting [7], but cumbersome for an email application where senders communicate arbitrary messages. For email, the shared code sheet would be better used to establish a shared cryptographic key for securing subsequent email communication. This, however, is a different protocol and is not an option for humans who cannot perform encryption without supporting technology. For this reason we distinguish between protocols that allow for the communication of a fresh constant generated by the sender and those that do not. Thus we use the generation and subsequent communication of a fresh constant as a symbolic representative for the ability to communicate an arbitrary message without requiring an encoding or other computational tasks.

Definition 3. *We say that a message m* originates *with an agent A in a trace tr, if m is a fresh term that A generates, that is, if $\mathsf{Fresh}(A, m) \in tr$.*

We now define what it means for a protocol to provide a particular type of channel. A channel property is a pair of predicates (p, q) each of which has domain $\mathcal{P}(\mathcal{G})^* \times \mathcal{C}_{\mathsf{pub}} \times \mathcal{C}_{\mathsf{pub}} \times \mathcal{M}$. A protocol provides a channel with a property defined by (p, q) if there exists a trace, two honest agents, and a message, such that p is satisfied and if for all traces, agents, and messages, q is satisfied. The existential requirement p ensures that the protocol provides some given functionality, such as communicating messages. The universal requirement q specifies a safety property, such as confidentiality. In order to reason about the (im-)possibility of secure communication, we need both of these requirements.

Definition 4. *Protocol $\mathcal{R}$* provides a channel with the property (p, q) *if*

$$\exists\, tr \in TR(\mathcal{R}), S, R \in \mathit{Honest}(tr), m \in \mathcal{M} \colon p(tr, S, R, m) \wedge$$
$$\forall\, tr \in TR(\mathcal{R}), S, R \in \mathit{Honest}(tr), m \in \mathcal{M} \colon q(tr, S, R, m).$$

We now define several channel properties, starting with the properties related to Definitions 2 and 3 above and concluding with security properties. A *communication channel* is defined by the property $(p_{\text{comm}}, q_{\text{comm}})$, where

$$p_{\text{comm}}(tr, S, R, m) := \text{communicate}(tr, S, R, m), \qquad q_{\text{comm}}(tr, S, R, m) := \top.$$

The predicate $\top$ (true) places no additional requirement on the set of traces. We say that a protocol provides a communication channel if the protocol satisfies the communication channel property. Intuitively, this states that the protocol is indeed a functioning communication protocol: it allows an honest agent to communicate a message to another honest agent. We will use analogous terminology for the channel properties to be defined in the remainder of this section.

An *originating* channel is defined by the property $(p_{\text{orig}}, q_{\text{orig}})$, where

$$p_{\text{orig}}(tr, S, R, m) := \mathsf{Fresh}(S, m) \in tr, \qquad q_{\text{orig}}(tr, S, R, m) := \top.$$

This says that a protocol with this property allows agents to generate fresh constants. We use this property together with the communication channel property to model an agent's ability to communicate an arbitrary message.

Example 2. A messaging protocol, such as an email protocol, must provide an originating channel. Items on code sheets or a username/password pair in a login protocol can be communicated over a non-originating channel.

We say that a protocol *combines* channel properties (p_1, q_1) and (p_2, q_2) if it satisfies the property $(p_1 \wedge p_2, q_1 \wedge q_2)$. In this case, we combine the adjectives used to describe the channel properties. For instance, we say that a protocol provides an originating communication channel if it combines an originating channel with a communication channel.

The two channel properties defined above concern protocols' functionality. We now define confidentiality and authenticity of messages, which are safety properties. A channel has the confidentiality property if the adversary does not learn a specified message. To identify the messages m that should remain confidential in a protocol, we annotate a protocol rule with a $\mathsf{Secret}(S, R, m)$ action.

Definition 5. *The* confidentiality property *is defined by* (p_{conf}, q_{conf})*, where*

$$p_{conf}(tr, S, R, m) := \mathsf{Secret}(S, R, m) \in tr$$
$$q_{conf}(tr, S, R, m) := \mathsf{Secret}(S, R, m) \in tr \rightarrow \mathsf{!K}(m) \notin tr.$$

A channel has the authenticity property for the agents S and R, if whenever R learns m, then m was previously communicated by S. To specify that a message m should be authentically communicated in a protocol, we annotate the protocol rule in which the message is learned with an $\mathsf{Authentic}(S, R, m)$ action.

Definition 6. *The* authenticity property *is defined by* (p_{auth}, q_{auth})*, where*

$$p_{auth}(tr, S, R, m) := \mathsf{Authentic}(S, R, m) \in tr$$
$$q_{auth}(tr, S, R, m) := \mathsf{Authentic}(S, R, m) \in tr \rightarrow \text{communicate}(tr, S, R, m).$$

Additional Channel Properties. One contribution of our work is to characterize the settings in which secure communication channels exist, even when some intended communication partners are dishonest. We therefore need to explicitly state which roles of a protocol are assumed to be executed by honest agents. This is done by annotating a protocol rule with the action $\mathsf{Trust}(A)$, where A is an agent.

We distinguish between the trust assumptions for confidentiality and authenticity and therefore define two properties.

Definition 7. *The* trust assumption for confidentiality *is defined by the property* (p_{ctrust}, q_{ctrust})*, where*

$$\begin{aligned} p_{ctrust}(tr, S, R, m) := \exists\, T \in \mathcal{C}_{\mathsf{pub}}, i \in \{1, \ldots, |tr|\} : \\ \mathsf{Trust}(T) \in tr_i \wedge \mathsf{Secret}(S, R, m) \in tr_i \\ \wedge\, T \in \mathit{Honest}(tr) \\ q_{ctrust}(tr, S, R, m) := \forall\, T \in \mathcal{C}_{\mathsf{pub}}, i \in \{1, \ldots, |tr|\} : \\ \mathsf{Trust}(T) \in tr_i \wedge \mathsf{Secret}(S, R, m) \in tr_i \\ \rightarrow T \in \mathit{Honest}(tr). \end{aligned}$$

The trust assumption for authenticity *is defined by the property* (p_{atrust}, q_{atrust})*, where*

$$\begin{aligned} p_{atrust}(tr, S, R, m) := \exists\, T \in \mathcal{C}_{\mathsf{pub}}, i \in \{1, \ldots, |tr|\} : \\ \mathsf{Trust}(T) \in tr_i \wedge \mathsf{Authentic}(S, R, m) \in tr_i \\ \wedge\, T \in \mathit{Honest}(tr) \\ q_{atrust}(tr, S, R, m) := \forall\, T \in \mathcal{C}_{\mathsf{pub}}, i \in \{1, \ldots, |tr|\} : \\ \mathsf{Trust}(T) \in tr_i \wedge \mathsf{Authentic}(S, R, m) \in tr_i \\ \rightarrow T \in \mathit{Honest}(tr). \end{aligned}$$

The two properties state that if a confidentiality or authenticity action occurs with a $\mathsf{Trust}(T)$ action, then agent T is honest. We can use these properties to express that whenever a confidentiality or authenticity claim is made, the specified intended communication partners are assumed to be honest. We achieve this expression with a relativization.

We say that a protocol provides the channel property (p_1, q_1) relative to the channel property (p_2, q_2) if it satisfies the property $(p_1 \wedge p_2, q_1 \vee \neg q_2)$. That is, both existential predicates need to be satisfied, and the universal predicate q_2 implies q_1. For instance, the property $(p_{\text{conf}} \wedge p_{\text{ctrust}}, q_{\text{conf}} \vee \neg q_{\text{ctrust}})$ specifies that a protocol provides a confidential channel *if* the sender's trusted communication partners are honest.

We also use relative channel properties to specify that a communication channel should be provided from a specific protocol role to another.

Definition 8. *Let* $\text{RoleMaps}(\mathcal{R}, tr)$ *be the set of all functions that assign to each role of* $\mathcal{R}$ *an agent executing that role in trace* tr*. The* channel from a role A to a role B *is defined by the property* (p_{role}, q_{role})*, where*

$$\begin{aligned} p_{role}(tr, S, R, m) &:= \exists\, \phi \in \text{RoleMaps}(\mathcal{R}, tr) : \phi(A) = S \wedge \phi(B) = R \\ q_{role}(tr, S, R, m) &:= \forall\, \phi \in \text{RoleMaps}(\mathcal{R}, tr) : \phi(A) = S \wedge \phi(B) = R \end{aligned}$$

Remark 1. We call the combination of a confidential channel and an authentic channel a *secure* channel. We will henceforth only consider protocols that provide a communication channel from one protocol role to another, relative to the sender's and recipient's trust assumptions combined with other channel properties. We will therefore omit the word "communication" for the channels provided by the protocols.

Remark 2. The communication topologies defined in Section 2 specify which roles are assumed to be executed by honest agents and which may be executed by dishonest agents. We will therefore drop the explicit designation of the set of roles that must be executed by honest agents when discussing protocols in the context of communication topologies.

4 Complete Classification of HISPs

The objective of a HISP is to provide a secure channel from the human H to the server S or vice versa. In the following sections we provide a complete classification of which HISP topologies allow such protocols. We first prove two general impossibility results concerning the establishment of confidential and authentic channels between agents. Then we classify the HISP topologies for which protocols exist that provide an originating secure channel. Such protocols permit the communication partners to securely exchange arbitrary messages. Afterwards, we consider the general case of HISPs that provide secure channels.

4.1 General impossibility results

The following two lemmas are impossibility results for secret establishment when confidential or authentic channels are available. They can be considered folklore, although, to the best of our knowledge, there are no published proofs for their statements. Impossibility results for secret establishment over insecure channels have been proven by Schmidt et al. [24].

The first lemma states the topological conditions under which no confidential channel from an honest agent S to an honest agent R can be created: If one of the agents has no initial knowledge, then there is no protocol that provides a confidential channel from S to R, even if S may send messages via authentic channels to R and R may send messages via confidential channels to S.

Lemma 1. *Let $\tau = (V, E, \eta, \mu)$ be a communication topology where $S, R \in V$ are distinct roles such that $\eta(S) = (\Sigma_S, K_S, \mathsf{honest})$, $\eta(R) = (\Sigma_R, K_R, \mathsf{honest})$ and $K_S = \emptyset$ or $K_R = \emptyset$. If the following two conditions are satisfied, then there exists no protocol for τ that provides a confidential channel from S to R.*

1. $\forall (a, b) \in E : (a = S \vee b = R) \rightarrow \mu(a, b) \in \{\circ\!\!\rightarrow\!\!\circ, \bullet\!\!\rightarrow\!\!\circ\}$
2. $\forall (a, b) \in E : (a = R \vee b = S) \rightarrow \mu(a, b) \in \{\circ\!\!\rightarrow\!\!\circ, \circ\!\!\rightarrow\!\!\bullet\}$

To prove Lemma 1, we map every trace where a message is confidentially communicated from S to R to a trace where S performs the same protocol steps, yet the adversary learns the message by impersonating R to S. This is possible because the messages from R to S are not authenticated. Thus, S cannot distinguish between information that R sends to S and information that the adversary sends. The technical details are given in Appendix B.1.

The following lemma states the dual of the preceding one: If an honest agent S has no access to an authentic (or secure) channel and another honest agent R has no access to a confidential (or secure) channel, then there is no protocol that provides an authentic channel from S to R.

Lemma 2. *Let $\tau = (V, E, \eta, \mu)$ be a communication topology where $S, R \in V$ are distinct roles such that $\eta(S) = (\Sigma_S, K_S, \mathsf{honest})$, $\eta(R) = (\Sigma_R, K_R, \mathsf{honest})$ and $K_S = \emptyset$ or $K_R = \emptyset$. If the following two conditions are satisfied, then there exists no protocol for τ that provides an authentic channel from S to R.*

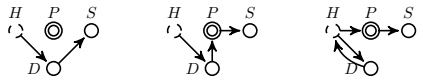
1. $\forall (a, b) \in E : (a = S \vee b = R) \rightarrow \mu(a, b) \in \{\circ\!\!\rightarrow\!\!\circ, \circ\!\!\rightarrow\!\!\bullet\}$
2. $\forall (a, b) \in E : (a = R \vee b = S) \rightarrow \mu(a, b) \in \{\circ\!\!\rightarrow\!\!\circ, \bullet\!\!\rightarrow\!\!\circ\}$

Note that we can strengthen Lemmas 1 and 2 by relaxing the empty initial knowledge condition on the agents. Instead of requiring that one of the two agents S and R has an empty initial knowledge, it suffices to make a restriction on terms that contain fresh constants. More precisely, for one of the two agents, say R, any fresh constant n occurring as a subterm of a term in the initial knowledge of R is either known to the adversary or no agent other than R has a term in his initial knowledge that contains n as a subterm.

4.2 Originating Secure Channels

For a human to send an arbitrary message securely to a remote server, we expect that the message must be input into a trusted device. To prove this, we separate the secure channel into its confidential and authentic components. There are no surprises for the confidential channel: A human can send a confidential message to a server if and only if the human can input the message into a trusted device and there is a communication path from the trusted device to the server.

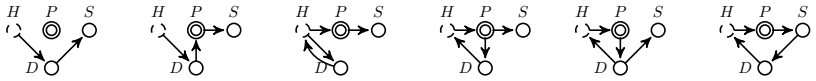
Theorem 1. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology. There exists a protocol for τ that provides an originating confidential channel from H to S if and only if $(H, D) \in E$ and $(D, S) \in E^+$. The following are all minimal graphs satisfying these conditions.*



(a) (b) (c)

Perhaps surprisingly, the possibilities for originating authentic channels are less restrictive than for originating confidential channels. As we now show, there are originating authentic channels from a human to a server, where the human *receives* a message from the trusted device instead of inputting one into it.

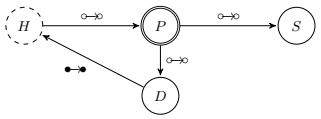
Theorem 2. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology. Then there exists a protocol for τ that provides an originating authentic channel from H to S if and only if $(H, S) \in E^+$, there exists an edge between H and D, and there exists an edge incoming to D as well as an edge outgoing from D. The following are all minimal graphs satisfying these conditions.*



(a) (b) (c) (d) (e) (f)

The difference between the two theorems reflects the human's limitations. The human's ability to generate fresh messages and compare previously sent messages with received messages suffices to guarantee originating authenticity for certain HISP topologies, but it is insufficient for originating confidentiality. The following example illustrates this difference.

Example 3. Let $\tau = (V, E, \eta, \mu)$ be the HISP topology shown in Figure 5a for the following scenario. A human user has a device with a small display. This is represented by $(D, H) \in E$ in τ. The device is connected to and receives input from the user's computer, so $(P, D) \in E$. The user sends messages to the server through the computer, therefore $(H, P) \in E$ and $(P, S) \in E$.



(a) HISP topology.

$$
\begin{aligned}
D &: \text{knows}(\langle H, S, k\rangle) \\
S &: \text{knows}(\langle H, D, k\rangle) \\
H &\multimap P : \text{fresh}(m).m \\
P &\multimap D : m \\
D &\bullet\!\!\multimap H : \langle m, h(\langle k, m\rangle)\rangle \;/\; \langle m, mac\rangle \\
H &\multimap P : mac \\
P &\multimap S : \langle m, mac\rangle \;/\; \langle m, h(\langle k, m\rangle)\rangle
\end{aligned}
$$

(b) Protocol.

Fig. 5: A protocol that provides an originating authentic channel from H to S.

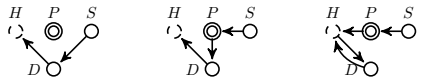
Figure 5b presents a protocol[4] for this HISP topology that provides an originating authentic, but not confidential, channel from the human user to the remote server. Namely, the user inputs his message m into the computer, which forwards it to the device. The device displays m along with a message authentication code (represented as a keyed hash) to the user. The message authentication code is computed by the device by using a symmetric key k that the device shares with the remote server. The user inputs the code mac into the computer, which then sends the message along with the code to the remote server. The correctness of the originating authenticity claim is verified by Tamarin. Hence

[4] For ease of reading, the protocol is represented in Alice & Bob notation. We provide a detailed description of our protocol specification rules in Appendix A.3.

the protocol provides an originating authentic channel as our model is faithfully represented by multiset rewriting rules within Tamarin. Since the graph shown in Figure 5a is not a supergraph of any of the graphs shown in Theorems 1 and 3, there is no protocol for this topology that provides a confidential channel in either direction.

A similar situation arises in the reverse direction. An originating confidential channel from the server to the human requires that the human receives the server's message from the trusted device.

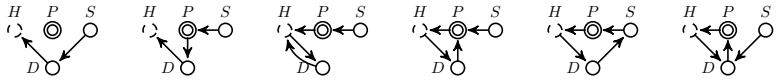
Theorem 3. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology. There exists a protocol for τ that provides an originating confidential channel from S to H if and only if $(D, H) \in E$ and $(S, D) \in E^+$. The following are all minimal graphs satisfying these conditions.*



(a) (b) (c)

Analogous to Theorem 2, the conditions for a human to receive an originating authentic message from a server are weaker than the conditions for originating confidential messages.

Theorem 4. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology. Then there exists a protocol for τ that provides an originating authentic channel from S to H if and only if $(S, H) \in E^+$, there exists an edge between H and D, and there exists an edge incoming to D as well as an edge outgoing from D. The following are all minimal graphs satisfying these conditions.*



(a) (b) (c) (d) (e) (f)

The proofs of these theorems are in Appendix B.3. Combining Theorems 1 and 2 shows that the topology of any HISP providing an originating secure channel from H to S is a supergraph of one of the graphs shown in Theorem 1.

Corollary 1. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology. There exists a protocol for τ that provides an originating secure channel from H to S if and only if $(H, D) \in E$ and $(D, S) \in E^+$.*

Theorems 3 and 4 imply Corollary 2. It states that the topology of any HISP that provides an originating secure channel from S to H is a supergraph of one of three graphs shown in Theorem 3.

Corollary 2. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology. There exists a protocol for τ that provides an originating secure channel from S to H if and only if $(D, H) \in E$ and $(S, D) \in E^+$.*

Note that a closer inspection of the proofs of the results in this section shows that all four theorems hold even if no initial knowledge is given to the human H. This can be seen by inspecting the protocols used to prove possibility results.

4.3 Secure Channels

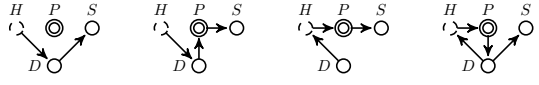
In this section we classify all HISP topologies for which there exist protocols that provide secure channels. As opposed to HISPs that provide originating secure channels, these protools may restrict the communication partners to a pre-defined set of messages that can be securely exchanged, such as codewords for candidates in an Internet voting system. Due to the weaker requirements regarding the origin of the exchanged messages, the set of HISP topologies for which protocols exist providing a secure channel is a superset of the former set of topologies. The following example illustrates a HISP sketch that provides a secure channel but not an originating secure channel.

Example 4. Suppose the human H needs to receive the result of a medical test from a testing facility S. As this information is sensitive, the human's computing platform P must not learn or modify this information. There are only few possible test outcomes and the result can therefore be communicated to H over a non-originating channel. To this end, H generates for each possible outcome a random code word. Then H uses a trusted device D to securely transmit the outcome/code word pairs to S. Once the test result is available, S sends to H via P the code word corresponding to the test result. Thus P receives one code word, but does not learn the corresponding test result. Since P does not know the other code words, it cannot change the result. The channel is non-originating, since S cannot communicate an arbitrary message to H, but only the code words selected by H. This initial sketch of a HISP can now be specified in detail and verified with Tamarin.

Our topology model and characterization can systematically guide us to this and several other HISPs. We discuss this design process after in Example 5 after presenting the characterization of the available HISP topologies .

We now classify all HISPs with respect to protocols providing secure channels from H to S and vice versa. We first consider the more interesting case where H has no initial knowledge and then discuss shared knowledge. Our main results are stated in the following two theorems. Theorem 5 shows the four minimal HISP topologies for which a protocol exists that provides a secure channel from a human H to a server S.

Theorem 5. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology where $K_H = \emptyset$. Then there is a protocol for τ that provides a secure channel H to S if and only if τ either contains an edge from D to H and a path from H to S or contains an edge from H to D and a path from D to S. All minimal graphs satisfying these conditions are shown below.*



(a) (b) (c) (d)

	$(D,H) \notin E$ $\wedge (H,D) \notin E$	$(D,H) \notin E$ $\wedge (H,D) \in E$ $\wedge (D,S) \notin E^+$	$(D,H) \notin E$ $\wedge (H,D) \in E$ $\wedge (D,S) \in E^+$	$(D,H) \in E$
A, C, S	no (Lemma 3)	no (Lemma 4)	yes (Lemma 6)	yes (Lemma 8)

(a) Existence of indicated channels from H to S when $(H,S) \in E^+$.

	$(D,H) \notin E$ $\wedge (H,D) \notin E$	$(D,H) \notin E$ $\wedge (H,D) \in E$ $\wedge (D,H) \notin E^+$	$(D,H) \notin E$ $\wedge (H,D) \in E$ $\wedge (D,H) \in E^+$	$(D,H) \in E$
A	no (Lemma 3)	no (Lemma 5)	yes (Lemma 9)	yes (Lemma 7)
C, S	no (Lemma 3)	no (Lemma 5)	iff $(H,S) \in E^+$ (Lemma 10)	yes (Lemma 7)

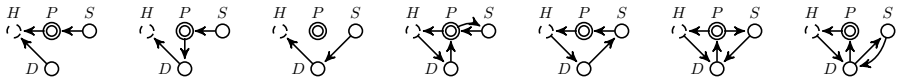
(b) Existence of indicated channels from S to H when $(S,H) \in E^+$.

Table 1: Classification of all HISP topologies. The cells state whether there exist protocols providing authentic (**A**), confidential (**C**), or secure (**S**) channels under the conditions shown on top.

Proof. We prove the theorem by case distinction. Table 1a classifies all HISP topologies that contain a path from H to S. For all other topologies, no protocol that provides an authentic, confidential, or secure channel from H to S can exist, because no information can be communicated from H to S. The statements concerning the four cases shown in Table 1a are proven by the lemmas referenced in the table, whose statements and proofs are given in Appendix B.

Theorem 6 shows the seven minimal HISP topologies for which a protocol exists that provides a secure channel from a server S to a human H. Its proof is analogous to the proof of Theorem 5 and follows from Table 1b and the lemmas referenced therein.

Theorem 6. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology where $K_H = \emptyset$. Then there is a protocol for τ that provides a secure channel S to H if and only if τ either contains an edge from D to H and a path from S to H or τ contains an edge from H to D and a path from S to itself that includes D and H. All minimal graphs satisfying these conditions are shown below.*



(a) (b) (c) (d) (e) (f) (g)

Note that Tables 1a and 1b also characterize HISP topologies with respect to authentic and confidential channels between H and S. The impossibility and possibility lemmas referred to in the tables are proven in Appendices B.1 and B.2.

In the following example we show how the minimal topologies of Theorem 6 can guide the design of a protocol that provides a secure channel from S to H.

Example 5. We return to the scenario of Example 4 where medical test results should be securely communicated from S to H. We are interested in a protocol

where H can suggest code words to be used for the test results. It follows from our characterization that this requires a path from H to S and excludes protocols based on the topologies (a)–(c).

Topology (d) suggests a protocol where H enters outcome/code word pairs into a device D that is connected to P. The code words are signed and encrypted by D and sent to S via P. The code word corresponding to the test result is sent from S to P which displays it to H.

Topology (e) has the simplest protocol flow. If we assume that postal mail is secure and that the medical test is a mail-in test, then the topology suggests that D could be a paper form provided with the test kit. The human fills in the form with code words next to the possible test outcomes and sends it with the kit to the testing facility. The resulting code word is communicated back to the human as above.

Topologies (f) and (g) apply in a scenario where the testing facility provides electronic data, but does not operate a download server. The protocol starts identically to the one outlined for topology (d). The results are sent back from S to D via an out-of-band channel and are then displayed on P.

Note that Theorems 5 and 6 assume that the human H has no initial knowledge. This may appear rather strong as, in reality, humans know many things including PINs and passwords. The following theorem states the simple topological condition for which HISPs providing secure channels exist under the assumption that there are secret terms in the initial knowledge of H and S. The only condition is that there exists a communication path.

Theorem 7. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology. If H and S share secret fresh nonces, i.e., $\exists x, y \in (\mathcal{C}_{\text{fresh}} \cap K_H \cap K_S) \setminus K_P : x \neq y$ and there is a path from H to S (from S to H) then there exists a protocol providing a secure channel from H to S (from S to H).*

To see why this theorem is true, suppose H and S have the term $\langle x, y \rangle$ in their initial knowledge, where x and y are fresh constants, not known to the dishonest agent P. Then H sends x to securely communicate y to S. Such protocols are of marginal interest in practice. In particular, x is a term that can be used only once and that a human would typically read off of a code sheet. But code sheets are modeled in HISPs as a supporting technology D and reading the code sheet is represented by the edge (D, H).

5 Related Work

Security ceremonies were informally introduced by Ellison [9,25] as a generalization of security protocols. They have given rise to several formal models that we discuss below. Our model is both more abstract and more precise than Ellison's description of security ceremonies.

Bella and Coles-Kemp extend security ceremonies with socio-technical elements such as a human agent's belief system and cultural values [1,2]. They

propose modeling security ceremonies using five layers: (1) the security of the protocol executed by the computers of the communicating partners; (2) the interprocess communication of the operating system; (3) human-computer interaction; (4) the user's state of mind and (5) the influence of society on individuals. In [2], they formalize layer (3) and give a case study verifying a user's confidence in the privacy assurance offered by a service provider in an example ceremony. In contrast to Bella and Coles-Kemp's work, we prove general results about secure communication scenarios that involve a human and his compromised computer.

Meadows and Pavlovic propose a logic of networks involving humans, devices, and computers. They analyze various authentication protocols [19] with respect to claimed security guarantees, but they do not provide a formal attacker model. Their formalism is comprehensive, but complex. In subsequent work, they extend their logic to a "logic of moves" and use it to analyze physical airport security procedures [17]. Similarly to Meadows and Pavlovic, we provide a graphical model for the communication topologies of security ceremonies. However, our abstraction is simpler while supporting the modeling of the communication topologies of security ceremonies in arbitrary detail. The level of abstraction we use is both intuitive to understand and straightforward to verify with existing protocol verification tools. Moreover, we provide a comprehensive formal attacker model for the verification of security properties of protocols involving humans, devices, and computers.

Carlos et al. sketch a method to formalize human knowledge distribution in security ceremonies [5]. In subsequent work [6], they consider an adversary that is weaker than the standard Dolev-Yao adversary in order to verify a Bluetooth pairing ceremony under realistic conditions. Their results are, however, specific to Bluetooth pairing ceremonies.

Further related research areas concern the *secure platform problem* [21], *problem of untrusted terminals* [3], and *trusted paths* [11,27]. The two former deal with the problem of ensuring that the user's computing platform faithfully executes a security protocol and does not leak confidential information to any unintended third party. The latter is the problem of providing secure channels from an input device to a trusted application and onward to an output device and focuses on implementation details at the system level.

Regarding our formalization of insecure, authentic, and confidential channels, Mödersheim and Viganò provide a security protocol model [18] based on abstract channels as assumptions and goals. Their *ideal channel model* is closely related to our channel rules in that it provides an abstract notation for sending messages via authentic and confidential channels. Whereas Mödersheim and Viganò implement their abstract channels using asymmetric cryptography, our channel rules directly specify the adversary's interaction with the abstract channels.

6 Conclusions

We have introduced a formal model for security protocols running in an environment with humans, computers, and devices as actors. The salient feature of our

model is the communication topology, which is a labeled graph whose vertices and edges represent the protocol's actors and their communication means. The vertex labeling represents the assumptions made about the actors' initial knowledge, computational capabilities, and honesty. The edge labeling assigns channel assumptions (such as confidential, authentic, insecure) to communication links. These assumptions determine whether secure communication is possible between two nodes in the topology. We have demonstrated the usefulness of our model by completely characterizing the necessary and sufficient conditions for the existence of HISPs. This is the class of security protocols where a human securely communicates with a remote server while using a compromised computer platform. Our model is supported by Tamarin [23], a security protocol verification tool and our examples show a concrete application of our modeling approach and its tool support.

References

1. G. Bella and L. Coles-Kemp. Seeing the full picture: the case for extending security ceremony analysis. In *Proceedings of 9th Australian Information Security Management Conference*, pages 49–55, 2011.
2. G. Bella and L. Coles-Kemp. Layered analysis of security ceremonies. In *Information Security and Privacy Research*, volume 376 of *IFIP Advances in Information and Communication Technology*, pages 273–286. Springer, 2012.
3. I. Berta. *Mitigating the attacks of malicious terminals.* PhD thesis, Budapest University of Technology and Economics, 2005.
4. C. Caleiro, L. Viganò, and D. A. Basin. On the semantics of Alice & Bob specifications of security protocols. *Theor. Comput. Sci.*, 367(1-2):88–122, 2006.
5. M. C. Carlos, J. E. Martina, G. Price, and R. F. Custódio. A proposed framework for analysing security ceremonies. In *SECRYPT*, pages 440–445, 2012.
6. M. C. Carlos, J. E. Martina, G. Price, and R. F. Custódio. An updated threat model for security ceremonies. In *28th Symposium on Applied Computing*, pages 1836–1843. ACM, 2013.
7. D. Chaum. SureVote: Technical overview. In *WOTE (2001)*, 2001.
8. D. Dolev and A. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198–208, 1983.
9. C. M. Ellison. Ceremony design and analysis. *IACR Cryptology ePrint Archive*, 2007:399, 2007.
10. A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner. A survey of mobile malware in the wild. In *SPSM'11, Proceedings of the 1st ACM Workshop Security and Privacy in Smartphones and Mobile Devices*, pages 3–14. ACM, 2011.
11. A. Filyanov, J. M. McCuney, A.-R. Sadeghiz, and M. Winandy. Uni-directional trusted path: Transaction confirmation on just one device. In *IEEE/IFIP 41st Intl. Conf. on Dependable Systems & Networks (DSN)*, pages 1–12. IEEE, 2011.
12. S. Gajek. A universally composable framework for the analysis of browser-based security protocols. In *Provable Security*, volume 5324 of *LNCS*, pages 283–297. Springer, 2008.
13. T. Groß, B. Pfitzmann, and A.-R. Sadeghi. Browser model for security analysis of browser-based protocols. In *Computer Security – ESORICS 2005*, volume 3679 of *LNCS*, pages 489–508. Springer, 2005.

14. A. Herzberg and R. Margulies. Forcing Johnny to login safely. In *Computer Security – ESORICS 2011*, volume 6879 of *LNCS*, pages 452–471. Springer, 2011.
15. A. Hiltgen, T. Kramp, and T. Weigold. Secure internet banking authentication. *Security & Privacy, IEEE*, 4(2):21–29, 2006.
16. U. Maurer and P. Schmid. A calculus for secure channel establishment in open networks. *Computer Security – ESORICS 94*, pages 173–192, 1994.
17. C. Meadows and D. Pavlovic. Formalizing physical security procedures. In *Security and Trust Management*, volume 7783 of *LNCS*, pages 193–208. Springer, 2013.
18. S. Mödersheim and L. Viganò. Secure pseudonymous channels. In *Computer Security – ESORICS 2009*, volume 5789 of *LNCS*, pages 337–354. Springer, 2009.
19. D. Pavlovic and C. Meadows. Actor-network procedures. In *Distributed Computing and Internet Technology*, volume 7154 of *LNCS*, pages 7–26. Springer, 2012.
20. M. Polychronakis, P. Mavrommatis, and N. Provos. Ghost turns zombie: Exploring the life cycle of web-based malware. LEET'08. USENIX Association, 2008.
21. R. Rivest. *Perspective on Electronic voting*, volume 2339 of *LNCS*, chapter "The Business of Electronic Voting (Panel)", pages 243–268. Springer, 2001.
22. M. Schläpfer and M. Volkamer. The secure platform problem: Taxonomy and analysis of existing proposals to address this problem. In *6th International Conference on Theory and Practice of Electronic Governance*, pages 410–418. ACM, 2012.
23. B. Schmidt, S. Meier, C. Cremers, and D. Basin. Automated analysis of Diffie–Hellman protocols and advanced security properties. In *Computer Security Foundations Symposium (CSF), 2012 IEEE 25th*, pages 78–94. IEEE, 2012.
24. B. Schmidt, P. Schaller, and D. Basin. Impossibility results for secret establishment. In *23rd Computer Security Foundations Symposium (CSF)*, pages 261–273. IEEE, 2010.
25. UPnP Security Working Group. UPnP$^{\text{TM}}$ security ceremonies, October 2003.
26. T. Weigold, T. Kramp, R. Hermann, F. Höring, P. Buhler, and M. Baentsch. The Zurich Trusted Information Channel–an efficient defence against man-in-the-middle and malicious software attacks. In *TRUST 2008*, volume 4968 of *LNCS*, pages 75–91. Springer, 2008.
27. Z. Zhou, V. D. Gligor, J. Newsome, and J. M. McCune. Building verifiable trusted path on commodity x86 computers. In *Security and Privacy (S&P), 2012 IEEE Symposium on*, pages 616–630. IEEE, 2012.

A Security Protocol Model

We first give full details of the Tamarin model [23], then full details of our extensions.

A.1 Tamarin Model Details

In this appendix we give full details of the Tamarin model. For ease of reading, we repeat a few definitions made in the main text.

Notation. The superscript $\_^b$ (bag) is used to denote operations on multisets such as $\cup^b$ for multiset-union. S^b denotes the set of finite multisets with elements from S. For a sequence s, $mset(s)$ denotes the multiset of its elements and $set(s)$ the corresponding set. A set S is also a multiset and for a multiset M, $set(M)$ denotes the corresponding set.

Term algebra. The term algebra is order-sorted with the sort *msg* and two incomparable subsorts *fresh* and *pub*. There are two countably infinite sets $\mathcal{C}_{\text{fresh}}$ and $\mathcal{C}_{\mathsf{pub}}$ of fresh and public constants, respectively, and we denote their union by $\mathcal{C}$. Let $S := \{fresh, pub, msg\}$. For each sort $s \in S$, there is a countably infinite set $\mathcal{V}_s$ of variables. We write $x : s$ to denote that $x \in \mathcal{V}_s$ and we let $\mathcal{V} := \bigcup_{s \in S} \mathcal{V}_s$.

A signature Σ is a set of function symbols, where each function symbol is associated with an arity. The subset of n-ary function symbols is denoted by Σ^n and we set $\Sigma^0 = \mathcal{C}_{\text{fresh}} \cup \mathcal{C}_{\mathsf{pub}}$. Messages are elements of the term algebra $\mathcal{T} = T(\Sigma, \mathcal{V})$, and ground terms are elements of $\mathcal{M} = T(\Sigma, \emptyset)$.

In this paper we assume that $\Sigma = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3$, where

$$\begin{aligned}
\Sigma^1 &= \{\pi_1(_), \pi_2(_), \mathrm{h}(_), \mathrm{pk}(_)\} \\
\Sigma^2 &= \{\langle _, _ \rangle, \mathrm{senc}(_, _), \mathrm{sdec}(_, _), \mathrm{aenc}(_, _), \mathrm{adec}(_, _), \mathrm{sign}(_, _)\}, \\
\Sigma^3 &= \{\mathrm{verify}(_, _, _)\}.
\end{aligned}$$

For $i > 0$, all functions in Σ^i are of sort $msg \times \cdots \times msg \to msg$. The function $\langle \_, \_ \rangle$ represents the pairing of terms, and π_1 and π_2 are the projections first and second, respectively. The functions $\mathrm{senc}(\_, \_)$ and $\mathrm{aenc}(\_, \_)$ represent symmetric and asymmetric encryption and $\mathrm{sdec}(\_, \_)$ and $\mathrm{adec}(\_, \_)$ represent symmetric and asymmetric decryption, respectively. The functions $\mathrm{sign}(\_, \_)$ and $\mathrm{verify}(\_, \_, \_)$ represent signing and verification of signatures. $\mathrm{h}(\_)$ represents a hash function and $\mathrm{pk}(\_)$ corresponds to the public key for a given secret key. For $a, b \in \mathcal{T}$, $true \in \mathcal{C}_{\mathsf{pub}}$, we let $\mathcal{E}$ be the following set of equations over Σ:

$$\begin{aligned}
\{\ & \pi_1(\langle a, b \rangle) = a, \pi_2(\langle a, b \rangle) = b, \\
& \mathrm{sdec}(\mathrm{senc}(a, b), b) = a, \mathrm{adec}(\mathrm{aenc}(a, \mathrm{pk}(b)), b) = a, \\
& \mathrm{verify}(\mathrm{sign}(a, b), a, \mathrm{pk}(b)) = true\ \}.
\end{aligned}$$

The equational theory $Eq(\Sigma, \mathcal{E})$ is the smallest congruence containing all instances of the equations of $\mathcal{E}$ over Σ.

A position p is a (possibly empty) sequence of natural numbers. The subterm $t_{|p}$ of t at position p is inductively defined by t if p is empty and by $(t_i)_{|p'}$ if $p = i, p'$ and $t = f(t_1, \ldots, t_k)$ for $f \in \Sigma^k$ and $1 \leq i \leq k$. The set of all subterms of t is denoted by $St(t)$. The set of variables of t is denoted by $vars(t) := St(t) \cap \mathcal{V}$.

Multiset term rewriting system The Tamarin model uses a multiset term rewriting system to represent all possible protocol behaviors. The system states are represented as finite multisets of *facts*. Facts are functions over $\mathcal{T}$ whose symbols appear in the signature Σ_{Fact} (disjoint from Σ) defined below. The set $\mathcal{F}$ consists of all facts $\mathsf{F}(t_1, \ldots, t_k)$ such that $t_i \in \mathcal{T}$ and $\mathsf{F} \in \Sigma^k_{Fact}$. The set of all ground facts, i.e., facts $\mathsf{F}(t_1, \ldots, t_k)$ such that $t_i \in \mathcal{M}$, is denoted by $\mathcal{G}$. Facts can be *linear* or *persistent*. Linear facts model resources that can only be consumed once, whereas persistent facts, prefixed by "!", model inexhaustible resources that can be consumed arbitrarily often.

State transitions are effected by labeled multiset rewriting rules. Each such rule is denoted by $l -[\, a \,]\!\!\rightarrow r$ with $l, a, r \in \mathcal{F}^*$. The elements in l, a, r are called the rule's premises, actions, and conclusions, respectively.

The labeled transition relation $\rightarrow_{\mathcal{R}} \subseteq \mathcal{G}^b \times \mathcal{P}(\mathcal{G}) \times \mathcal{G}^b$ for a set of multiset rewriting rules $\mathcal{R}$ is defined as:

$$\frac{l -[\, a \,]\!\!\rightarrow r \ \in ginsts(\mathcal{R}) \quad lfacts(l) \subseteq^b S \quad pfacts(l) \subseteq set(S)}{S \xrightarrow{set(a)}_{\mathcal{R}} (S \setminus^b lfacts(l)) \cup^b mset(r)}, \tag{13}$$

where $lfacts(l)$ is the multiset of all linear facts in l, $pfacts(l)$ is the set of all persistent facts in l, and $ginsts(\mathcal{R})$ consists of all ground instances of rules in $\mathcal{R}$. Formally, $ginsts(\mathcal{R})$ is the set of all rules $l -[\, a \,]\!\!\rightarrow r$ for which there exists a rule $l' -[\, a' \,]\!\!\rightarrow r' \in \mathcal{R}$ with $|l'| = |l|$, $|a'| = |a|$, $|r'| = |r|$, and a substitution $\sigma : \mathcal{F} \rightarrow \mathcal{G}$ such that $\forall i \in \{1, \ldots, |l|\}, j \in \{1, \ldots, |a|\}, k \in \{1, \ldots, |r|\} : \sigma(l'_i) = l_i \wedge \sigma(a'_j) = a_j \wedge \sigma(r'_k) = r_k$. The transition rewrites the current state by replacing the facts in l with the facts in r and is labeled with the facts in a.

For a set of multiset rewriting rules $\mathcal{R}$, the system behaviors are given by the set of traces $TR(\mathcal{R})$, defined as:

$$\begin{aligned} TR(\mathcal{R}) := \quad \{\, [a_1, \ldots, a_n] \mid \exists S_1, \ldots, S_n \in \mathcal{G}^b : \emptyset^b \xrightarrow{a_1}_{\mathcal{R}} \cdots \xrightarrow{a_n}_{\mathcal{R}} S_n \\ \wedge \forall i \neq j \forall x : (S_{i+1} \setminus^b S_i) = \{\mathsf{Fr}(x)\} \Rightarrow \\ (S_{j+1} \setminus^b S_j) \neq \{\mathsf{Fr}(x)\}\}. \end{aligned} \tag{14}$$

Fr facts may only be generated by a distinguished model-specific rule (to be discussed in the next subsection). Thus, the second conjunct ensures that each instance of the rule for generating Fr facts is used at most once in a trace and therefore each consumer of a Fr fact obtains a different fresh constant. Hence, a trace $tr \in TR(\mathcal{R})$ is a finite sequence of sets of actions $tr_i \in \mathcal{P}(\mathcal{G})$, $i \in \{1, \ldots, |tr|\}$. We write $b \in tr$ if $b \in tr_i$ for some $1 \leq i \leq |tr|$, that is, when the action b occurs in a set of ground actions in the trace tr.

Adversary model The network is controlled by a Dolev-Yao adversary [8]. The adversary chooses whether to deliver each message. He eavesdrops on, injects, and modifies messages on channels. However, he can neither eavesdrop on confidential (or secure) channels nor inject or modify messages on authentic (or secure) channels. The message deduction rules in $\mathcal{MD}$ represent his capability to receive, construct, and send messages in a protocol execution:

$$\mathcal{MD} := \{\, [\mathsf{Out}(x)] -[\, !\mathsf{K}(x) \,]\!\!\rightarrow [!\mathsf{K}(x)], \tag{15}$$

$$[!\mathsf{K}(x)] -[\, !\mathsf{K}(x) \,]\!\!\rightarrow [\mathsf{In}(x)], \tag{16}$$

$$[\,] -[\, !\mathsf{K}(x : pub) \,]\!\!\rightarrow [!\mathsf{K}(x : pub)], \tag{17}$$

$$[\mathsf{Fr}(x)] -[\, !\mathsf{K}(x) \,]\!\!\rightarrow [!\mathsf{K}(x)] \,\} \tag{18}$$

$$\cup \{\, [!\mathsf{K}(x_1), \ldots, !\mathsf{K}(x_k)] -[\, !\mathsf{K}(f(x_1, \ldots, x_k)) \,]\!\!\rightarrow [!\mathsf{K}(f(x_1, \ldots, x_k))] \mid f \in \Sigma^k \wedge k > 0 \,\}. \tag{19}$$

The !K fact appearing in all rules of $\mathcal{MD}$ is used to store and observe the adversary's knowledge in a trace and plays a role in specifying secrecy properties[5]. Rule (15) allows the adversary to learn all terms that are produced with Out facts and rule (16) allows the adversary to input any term in his knowledge into an In fact. The Rules (17) and (18) represent the adversary's capabilities to learn public and freshly generated constants, respectively. The set of Rules (19) allow the adversary to apply any function in Σ^k, for $k > 0$, to known messages.

A.2 Extended Model Details

In this appendix we provide additional details to our model extensions.

We first define a fixed set of fact symbols and rules to model security protocols. The following equations summarize all facts used in the model.

$$\begin{aligned}
\Sigma_{Fact} :=& \Sigma^1_{Fact} \cup \Sigma^2_{Fact} \cup \Sigma^3_{Fact}, \text{ where} \\
\Sigma^1_{Fact} :=& \{\mathsf{Fr}, \mathsf{Out}, \mathsf{In}, \mathsf{!K}, \mathsf{Agent}, \mathsf{Honest}, \mathsf{Dishonest}, \mathsf{Trust}\}, \\
\Sigma^2_{Fact} :=& \{\mathsf{!Auth}, \mathsf{!Conf}, \mathsf{Fresh}, \mathsf{Comm}, \mathsf{Learn}\}, \\
\Sigma^3_{Fact} :=& \{\mathsf{Snd_I}, \mathsf{Rcv_I}, \mathsf{Snd_A}, \mathsf{Rcv_A}, \mathsf{Snd_C}, \mathsf{Rcv_C}, \mathsf{Snd_S}, \mathsf{Rcv_S}\} \\
& \cup \{\mathsf{!Sec}, \mathsf{Secret}, \mathsf{Authentic}, \mathsf{AgentState}\}.
\end{aligned}$$

The set of all facts $\mathcal{F}$ is therefore

$$\mathcal{F} := \left\{ f(t_1, \ldots, t_k) \mid f \in \Sigma^k_{Facts} \wedge t_1, \ldots, t_k \in \mathcal{T} \right\}.$$

We use Agent, Honest, Dishonest actions to indicate agents in a trace. Honest agents are indicated with Honest, dishonest agents with Dishonest. Once an agent is indicated to be honest it cannot become dishonest or vice-versa. This is enforced in Tamarin with an axiom. Trust is used to indicate agents which are assumed to be honest for the purpose of security properties, see Definition 7 below. These are agents whose roles are marked honest in the communication topology.

We distinguish between model and protocol specification rules, denoted by $\mathcal{R}_{Model}$ and $\mathcal{R}_{Spec}$ respectively. The former are a fixed set of rules introduced in the following and the latter specify the security protocol. There are four types of model rules: Rules for generating fresh constants $\mathcal{FR}$, message deduction rules $\mathcal{MD}$, channel rules $\mathcal{CH}$, and dishonest agent rules $\mathcal{DA}$. The message deduction rules and the fresh constant generation rule are adapted from [23], while the remaining rules are specific to our model. Thus our model rules are

$$\mathcal{R}_{Model} := \mathcal{FR} \cup \mathcal{MD} \cup \mathcal{CH} \cup \mathcal{DA}.$$

The only rule producing fresh constants and thereby creating Fr facts is Rule (20). Recall that due to Equation (14), every fresh constant is produced

[5] For efficiency reasons, Tamarin distinguishes between !KU and !KD facts. For simplicity, we refer to both of these as !K facts.

at most once in a trace. Fresh constants can be obtained (generated) by honest agents using Rule (21). The adversary can generate fresh constants for dishonest agents using Rule (18).

$$\mathcal{FR} := \{\ [\,]\!-\![\ ]\!\rightarrow [\mathsf{Fr}(x : \mathit{fresh})]\ , \tag{20}$$

$$[\mathsf{Fr}(x)]\!-\![\ \mathsf{Fresh}(A, x), \mathsf{Honest}(A)\ ]\!\rightarrow [\mathsf{Fresh}(A, x)]\ \} \tag{21}$$

A.3 Protocol Specification

This appendix provides the details on how we specify a protocol.

A protocol defines a *setup* and the behavior of a set of *roles*. The corresponding protocol specification $\mathcal{R}_{Spec}$ consists of a finite number of setup rules and protocol rules.

Setup rules are used to initialize the protocol, i.e., to generate the initial knowledge and to distribute it to the corresponding protocol agents by generating the initial $\mathsf{AgentState}$ facts for all roles. Formally, a setup rule $l\!-\![\ a\ ]\!\rightarrow r$ is a rule where:

S1 Only Fresh and Fr facts occur in l.
S2 For every $\mathsf{AgentState}(A, \_, \_)$ fact in r, there is an $\mathsf{Agent}(A)$, $\mathsf{Honest}(A)$, or $\mathsf{Dishonest}(A)$ action in a.

A role consists of a set of protocol rules, specifying the sending and receiving of messages, branching and looping conditions, and the generation of fresh constants. In what follows, we only allow protocols where after the setup phase all information is exchanged using the channels defined in our channel abstraction model above. That is, information may not flow from one agent to another in any way other than by one of the channels defined in $\mathcal{CH}$. A protocol rule $l\!-\![\ a\ ]\!\rightarrow r$ is a rule such that the following 6 conditions are satisfied.

P1 The facts in l, a, and r do not contain elements of $\mathcal{C}_{\text{fresh}}$ as subterms.
P2 Only $\mathsf{Rcv_I}, \mathsf{Rcv_A}, \mathsf{Rcv_C}, \mathsf{Rcv_S}$, Fresh, and $\mathsf{AgentState}$ facts occur in l.
P3 Only $\mathsf{Snd_I}, \mathsf{Snd_A}, \mathsf{Snd_C}, \mathsf{Snd_S}$, and $\mathsf{AgentState}$ facts occur in r.
P4 Exactly one $\mathsf{AgentState}$ fact occurs in l, zero or more $\mathsf{AgentState}$ facts occur in r.
P5 If $\mathsf{AgentState}(A, c, n)$ occurs in l, then
  (a) every $\mathsf{Rcv_I}$, $\mathsf{Rcv_A}$, $\mathsf{Rcv_C}$, $\mathsf{Rcv_S}$, Fresh fact is of the form $\mathsf{Rcv_I}(B, A, x)$, $\mathsf{Rcv_A}(B, A, x)$, $\mathsf{Rcv_C}(B, A, x)$, $\mathsf{Rcv_S}(B, A, x)$, $\mathsf{Fresh}(A, x)$ where $B, x \in \mathcal{T}$,
  (b) every Learn, Comm, Secret, $\mathsf{Authentic}$, $\mathsf{Snd_I}$, $\mathsf{Snd_A}$, $\mathsf{Snd_C}$, $\mathsf{Snd_S}$ fact is of the form $\mathsf{Learn}(A, x)$, $\mathsf{Comm}(A, x)$, $\mathsf{Secret}(A, B, x)$, $\mathsf{Authentic}(B, A, x)$, $\mathsf{Snd_I}(A, B, x)$, $\mathsf{Snd_A}(A, B, x)$, $\mathsf{Snd_C}(A, B, x)$, $\mathsf{Snd_S}(A, B, x)$, where $B \in \mathcal{C}_{\mathsf{pub}}$, $x \in \mathcal{T}$ and x is derivable from terms in $\mathcal{C}_{\mathsf{pub}}$, terms in Fresh and $\mathsf{Rcv_I}, \mathsf{Rcv_A}, \mathsf{Rcv_C}, \mathsf{Rcv_S}$ facts occurring in l, and terms in n.
  (c) every $\mathsf{AgentState}$ fact in r is $\mathsf{AgentState}(A, c', n')$, where $c' \in \mathcal{C}_{\mathsf{pub}}$ and n' is derivable from terms in $\mathcal{C}_{\mathsf{pub}}$, terms in Fresh and $\mathsf{Rcv_I}, \mathsf{Rcv_A}, \mathsf{Rcv_C}, \mathsf{Rcv_S}$ facts occurring in l, and terms in n.
P6 $vars(r) \subseteq vars(l) \cup \mathcal{V}_{\mathsf{pub}}$.

Remark. *A protocol rule that contains a receive fact in its premise and a send fact in its conclusion models the reception and sending of messages as an* atomic *protocol execution step. If the agent executing the protocol step is dishonest, then the adversary may not be able to influence the message to be sent. To model the general situation where reception and subsequent sending of messages are not atomic, two separate rules need to be specified, one for the reception of messages and a corresponding update of the receiver's state, and another one to specify the sending of messages. The adversary may then reveal and modify a dishonest agent's state after the dishonest agent receives a message and before the agent sends the subsequent message.*

To be able to reconstruct all system states from a trace, we add a unique action R_i to every rule in $\mathcal{R}$. Formally, we do this as follows. Let q be a sequence of all rules in $\mathcal{R}$ such that every rule in $\mathcal{R}$ occurs exactly once in q. The action R_i contains all variables of the rule q_i in q as an argument. To this end, we must map the elements of the set of variables in the premises and conclusions to an ordered list. We denote such a map by *list*. Thus the set of rules that allows us to reconstruct all system states from a trace for a given protocol specification $\mathcal{R}$ is given by

$$\begin{aligned}\{\; l \text{-}[\, a \,]\!\rightarrow r \;\mid \exists i \in \{1, \ldots, |q|\} :\; l \text{-}[\, a' \,]\!\rightarrow r \;= q_i \wedge \\ a = a' \cdot [\mathsf{R}_i(\mathit{list}(vars(l) \cup vars(r)))] \;\}.\end{aligned} \tag{22}$$

Protocol notation. For ease of reading, we represent protocols in an extended Alice & Bob notation from which the corresponding protocol rules can be easily obtained. The extension contains the symbols in the *LinkProp* set representing insecure $\circ\!\!\rightarrow\!\!\circ$, authentic $\bullet\!\!\rightarrow\!\!\circ$, confidential $\circ\!\!\rightarrow\!\!\bullet$, and secure $\bullet\!\!\rightarrow\!\!\bullet$ (i.e., authentic and confidential) channels. For instance, we write $A \circ\!\!\rightarrow\!\!\circ B : m$ to express that a message m is to be sent from an agent executing role A to an agent executing role B over an insecure channel. To express that the message is sent over an authentic channel, we write $A \bullet\!\!\rightarrow\!\!\circ B : m$, whereby only A can send messages using the authentic channel.

In general, an Alice & Bob specification leaves room for different interpretations [4]. When such ambiguities arise, we indicate both the message sent and the message pattern expected to be received and separate them with " / ", as in $A \circ\!\!\rightarrow\!\!\circ B : m \;/\; m'$. The variables in m' determine how the received message is parsed by an agent executing role B. To express the initial knowledge m of an agent executing role A, we write $A : \mathrm{knows}(m)$. To express that the agent generates fresh constants $m_1, \ldots, m_k$, we write $A : \mathrm{fresh}(m_1, \ldots, m_k)$ or $A \circ\!\!\rightarrow\!\!\circ B : \mathrm{fresh}(m_1, \ldots, m_k).m$ when the generation is followed by a send event.

B Proof Details

B.1 Proof Details: Impossibility results

In this appendix we provide the proof details for all the impossibility lemmas of Section 4.1 in the paper. Lemmas that first appear in this appendix are numbered with letters.

Lemma 1. *Let $\tau = (V, E, \eta, \mu)$ be a communication topology where $S, R \in V$ are distinct roles such that $\eta(S) = (\Sigma_S, K_S, \mathsf{honest})$, $\eta(R) = (\Sigma_R, K_R, \mathsf{honest})$ and $K_S = \emptyset$ or $K_R = \emptyset$. If the following two conditions are satisfied, then there exists no protocol for τ that provides a confidential channel from S to R.*

1. $\forall (A, B) \in E : (A = S \vee B = R) \rightarrow \mu(A, B) \in \{\circ\!\!\rightarrow\!\!\circ, \bullet\!\!\rightarrow\!\!\circ\}$
2. $\forall (A, B) \in E : (A = R \vee B = S) \rightarrow \mu(A, B) \in \{\circ\!\!\rightarrow\!\!\circ, \circ\!\!\rightarrow\!\!\bullet\}$

Proof. We start by simplifying the protocol when the number of roles specified for the protocols is greater than two. Suppose that S has an empty initial knowledge. Then we combine all roles other than the role of S into the role of R. If the initial knowledge of S is not empty, then by hypothesis, the initial knowledge of R must be empty. In this case we combine all roles other than the role of R into the role of S. This type of transformation preserves confidentiality: If the original protocol provides a confidential communication channel from S to R with role specifications for several other agents, then it provides it in particular for traces where the additional roles are instantiated with the honest agents S and R.

We may thus assume that the protocol contains only the two agents S and R. We may further assume without loss of generality that S transmits and R receives all messages over an authentic channel and that R transmits and S receives all messages over a confidential channel. That is, we may upgrade all insecure channels to channels with these stronger guarantees.

Let tr be a shortest trace satisfying the confidentiality condition (Definition 5) and the communication condition (Definition 2). Then $\mathsf{Secret}(S, R, m) \in tr$, $\mathsf{Comm}(S, m) \in tr$, and $\mathsf{Learn}(R, m) \in tr$ for some $m \in \mathcal{M}$. If there is no such trace, then we are done, since then the protocol does not provide a confidential communication channel. Otherwise, we have that $!\mathsf{K}(m) \notin tr$. We exhibit a trace tr' in which $\mathsf{Secret}(S, R, m) \in tr'$ and $!\mathsf{K}(m) \in tr'$. Let g be the sequence of ground instances of rules which gives rise to the trace tr. By Equation (22), we can obtain this sequence from the trace tr by using the unique facts R_i appearing in the trace.

We construct a sequence of (ground) rewriting rules g' from g that give rise to a trace tr' for which the confidentiality condition is not satisfied. To this end, we will replace rules in g which contain $\mathsf{AgentState}(R, \_, \_)$ by instantiations of rules in $\mathcal{MD}$ and $\mathcal{CH}$. In order for such a transformation to produce a valid sequence of rewriting rules, we need to satisfy the following two conditions:

- Facts consumed by a rule g'_i must have been produced by a rule g'_j, for $j < i$.
- Every rule g'_i is a ground instantiation of a protocol rule in $\mathcal{R}$.

We obtain the transformation from g to g' by describing a series of deletions and insertions performed on the sequence g. For a rule g_i in g, $l(g_i)$ refers to the premises of g_i, $a(g_i)$ to the actions, and $r(g_i)$ to the consequences. Thus, $g_i = [l(g_i)] -[a(g_i)]\!\rightarrow [r(g_i)]$.

1. For ease of reference, we keep track of the correspondence between the fresh terms in the knowledge of agent R and the adversary's fresh terms via the partial map $\phi : \mathcal{C}_{\text{fresh}} \rightarrow \mathcal{C}_{\text{fresh}}$.

2. For every setup rule g_i containing an $\mathsf{AgentState}(R, c, n)$ fact for some $c, n \in \mathcal{M}$ we make the following two insertions.
Insertion 1. For every fact $\mathsf{Fresh}(R, y) \in l(g_i)$ there are unique rules

$$g_k = [\,]\text{–}[\,]\rightarrow[\mathsf{Fr}(y)]$$

and

$$g_j = [\mathsf{Fr}(y)]\text{–}[\,]\rightarrow[\mathsf{Fresh}(R, y)]\,,$$

$k < j < i$, producing $\mathsf{Fresh}(R, y)$.
We insert an instantiation of the $\mathcal{FR}$ rule $[\,]\text{–}[\,]\rightarrow[\mathsf{Fr}(x)]$ immediately after g_k and an instantiation of the $\mathcal{MD}$ rule $[\mathsf{Fr}(x)]\text{–}[\,]\rightarrow[!\mathsf{K}(x)]$ immediately after g_j. We set $\phi(y) := x$.
Insertion 2. For every public constant $C : pub$ in R's knowledge n, we insert a rule $[\,]\text{–}[\,]\rightarrow[!\mathsf{K}(C : pub)]$ before g_i.
After these insertions, we have a correspondence between R's initial knowledge and the adversary's knowledge. The modified sequence of rules remains a valid sequence.
3. Let g_i be the first instantiation of a role specification rule in g that contains an $\mathsf{AgentState}(R, c, n)$ fact for some $c, n \in \mathcal{M}$. By **P2** through **P4** and our hypothesis, we have only $\mathsf{Fresh}(R, \_)$, $\mathsf{Rcv}_\mathsf{A}(\_, R, \_)$, and $\mathsf{AgentState}(R, \_, \_)$ facts in $l(g_i)$, $\mathsf{Snd}_\mathsf{C}(R, \_, \_)$ and $\mathsf{AgentState}(R, \_, \_)$ facts in $r(g_i)$, and $\mathsf{Learn}(R, \_)$, $\mathsf{Comm}(R, \_)$, $\mathsf{Secret}(R, \_, \_)$, and $\mathsf{Authentic}(\_, \_, \_)$ facts in $a(g_i)$. We delete the rule g_i after having made the following changes.
Change 1. For every $\mathsf{Fresh}(R, x)$ fact in $l(g_i)$, there exists a rule $g_j = [\mathsf{Fr}(x)]\text{–}[\,]\rightarrow[\mathsf{Fresh}(R, x)]$, $j < i$, producing that fact. We replace g_j by the rule $[\mathsf{Fr}(x)]\text{–}[\,]\rightarrow[!\mathsf{K}(x)]$. Thus every fresh term learned by R in g is learned by the adversary in g'.
Change 2. For every $\mathsf{Rcv}_\mathsf{A}(S, R, m)$ fact we insert before g_i the rule $[\mathsf{Out}(\langle S, R, m\rangle)]\text{–}[\,]\rightarrow[!\mathsf{K}(\langle S, R, m\rangle)]$, which is an instantiation of $\mathcal{MD}$ Rule (15), and two instantiations of $\mathcal{MD}$ Rule (19) using the projecting functions in order to arrive at the facts $!\mathsf{K}(m)$, $!\mathsf{K}(S)$, $!\mathsf{K}(R)$. Note that there exists an $\mathsf{Out}(\langle S, R, m\rangle)$ fact in $r(g_j)$ for some $j < i$ due to instantiations of $\mathcal{CH}$ Rules (6) and (7) which are the source of the $\mathsf{Rcv}_\mathsf{A}(S, R, m)$ fact.
Thus every message received by R in g is learned by the adversary in g'.
Change 3. By step 2 above (i.e. modifications of the setup rules), **P5**(c), and previous applications of the present step, all terms in $\mathsf{AgentState}(R, c, n) \in l(g_i)$ that are derivable from n, are also derivable from the adversary's knowledge up to substitution of fresh constants y in the domain of ϕ by $\phi(y)$.
Change 4. For each $\mathsf{Snd}_\mathsf{C}(R, S, m)$ fact in $r(g_i)$, we can synthesize from the adversary's knowledge a message $\tilde{m}$ that is equal to m up to substitution of fresh constants y in the domain of ϕ by their image $\phi(y)$. To this end, we insert after g_i instantiations of $\mathcal{MD}$ Rule (19) to produce the fact $!\mathsf{K}(\langle R, S, \tilde{m}\rangle)$. We delete the corresponding rule $g_j =$ $[\mathsf{Snd}_\mathsf{C}(R, S, m)]\text{–}[\,\mathsf{Snd}_\mathsf{C}(R, S, m)\,]\rightarrow[!\mathsf{Conf}(S, m)]$, $j > i$, if it exists, and replace every subsequent rule

$[!\mathsf{Conf}(S,m), \mathsf{In}(R)] -[\ \mathsf{Rcv_C}(R,S,m)\ ]\!\rightarrow [\mathsf{Rcv_C}(R,S,m)]$ with the rules

$$[!\mathsf{K}(\langle R,S,\tilde{m}\rangle)] -[\ !\mathsf{K}(\langle R,S,\tilde{m}\rangle)\ ]\!\rightarrow [\mathsf{In}(\langle R,S,m\rangle)]$$

and

$$[\mathsf{In}(\langle R,S,m\rangle)] -[\ \ ]\!\rightarrow [\mathsf{Rcv_C}(R,S,m)]\ .$$

The latter of these rules is an instantiation of $\mathcal{CH}$ Rule (10) and the former is an incorrect instantiation of $\mathcal{MD}$ Rule (16). This is due to a mismatch between the adversary's knowledge $!\mathsf{K}(\langle R,S,\tilde{m}\rangle)$ and the produced fact $\mathsf{In}(\langle R,S,m\rangle)$. This is resolved in step 4 below.

Change 5. Note that each $\mathsf{AgentState}$ fact in $r(g_i)$ is of the form $\mathsf{AgentState}(R,c,n)$, where the terms c and n are derivable from $!\mathsf{K}$ facts up to substitution of fresh constants in the domain of the ϕ function.

Change 6. For each $\mathsf{Learn}(R,x)$ fact in $a(g_i)$, we insert instantiations of $\mathcal{MD}$ Rule (19) after g_i in order to arrive at $!\mathsf{K}(x)$ (up to substitutions of fresh constants in the domain of ϕ). This is possible, since x is a term derivable from public constants, messages in $\mathsf{Rcv_A}(S,R,m)$ facts and knowledge in $\mathsf{AgentState}(R,c,n)$ facts.

Change 7. We may ignore the $\mathsf{Authentic}$ and Secret facts in $a(g_i)$: We may ignore the $\mathsf{Authentic}$ facts, since these indicate an authenticity claim that we are not considering here. We may ignore Secret facts occurring in an instance of a rule considered here, since these concern the confidentiality of messages sent by R, as opposed to those sent by S. We may ignore the Comm facts because they label the transmission of messages from R rather than those from S.

We repeat this step 3 as long as there are rules g_i containing $\mathsf{AgentState}(R,\_,\_)$ facts in $l(g_i)$.

4. We exchange the fresh values y in the initial knowledge of R acquired in the setup rules with the corresponding fresh values $\phi(y)$ in the adversary's knowledge ($!\mathsf{K}(\phi(y))$) as follows.
   For every setup rule g_i containing a $\mathsf{AgentState}(R,c,n)$ fact and a $\mathsf{Fresh}(R,y)$ fact, we replace in all terms the fresh constant y by the fresh constant $\phi(y)$. We replace the unique rule g_j, $j<i$, producing the fact $\mathsf{Fresh}(R,y)$ by the rule $[\mathsf{Fr}(\phi(y))] -[\ \ ]\!\rightarrow [\mathsf{Fresh}(R,\phi(y))]$.
   For every instantiation of a $\mathcal{MD}$ rule in g, we replace in all terms all fresh constants $\phi(y)$ by y.

After the above replacements, we obtain a sequence of rules and consequently a trace tr' in which the adversary impersonates R. R does not perform any protocol steps other than having its initial knowledge set up. We finally append the rule $[!\mathsf{K}(m)] -[\ !\mathsf{K}(m)\ ]\!\rightarrow [\mathsf{In}(m)]$ to g' in order to have $!\mathsf{K}(m) \in tr'$. Thus we have a trace where the adversary learns m, yet $\mathsf{Secret}(S,R,m) \in tr'$.

Lemma 2. *Let $\tau = (V,E,\eta,\mu)$ be a communication topology where $S,R \in V$ are distinct roles such that $\eta(S) = (\Sigma_S, K_S, \mathsf{honest})$, $\eta(R) = (\Sigma_R, K_R, \mathsf{honest})$ and $K_S = \emptyset$ or $K_R = \emptyset$. If the following two conditions are satisfied, then there exists no protocol for τ that provides an authentic channel from S to R.*

1. $\forall (A,B) \in E : (A = S \vee B = R) \rightarrow \mu(A,B) \in \{\circ\!\!\rightarrow\!\!\circ, \circ\!\!\rightarrow\!\!\bullet\}$
2. $\forall (A,B) \in E : (A = R \vee B = S) \rightarrow \mu(A,B) \in \{\circ\!\!\rightarrow\!\!\circ, \bullet\!\!\rightarrow\!\!\circ\}$

The proof idea for this lemma is the same as for the preceding one. The adversary impersonates S to R. This is possible, since messages from S to R are not authenticated. Thus, R cannot distinguish between information that S sends to R and information that the adversary sends. We omit the technical details.

Lemma A *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology where $K_H = \emptyset$ and in which there are no outgoing edges from D. Then there exists no protocol for τ that provides a confidential channel and there exists no protocol for τ that provides an authentic channel between S and H.*

Proof. Since none of H, S, P receive any messages from D, a protocol that provides a confidential or authentic channel between H and S with such a role specification for D, also provides such a channel without a role specification for D. By Lemmas 1 and 2 no such protocol exists.

Lemma 3. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology where $K_H = \emptyset$ and no edge between H and D exists. Then there exists no protocol for τ that provides a confidential channel and there exists no protocol for τ that provides an authentic channel from H to S or vice-versa.*

The idea for the proof is that every trace establishing a confidential or authentic channel that involves actions of D, can be transformed into a valid trace with the same properties but not involving D. Since the channels between H and S are insecure, by Lemmas 1 and 2 neither confidential nor authentic channels can be established between H and S.

Proof. Since there is no edge between H and D, all communication channels to and from H are insecure.

Since there are no edges between H and D and all edges between D and P are labeled insecure as are the edges between S and P, we may include the D role in the S role while maintaining the property that all channels between S and P are labeled insecure. We thus obtain a protocol where all channels to and from S are insecure.

Thus the hypotheses of Lemmas 1 and 2 are satisfied and thus there is no protocol establishing a confidential or authentic channel between H and S.

Lemma 4. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology with $K_H = \emptyset$, $(H, D) \in E$, $(D, H) \notin E$, and $(D, S) \notin E^+$. Then there exists no protocol for τ that provides a confidential channel and there exists no protocol for τ that provides an authentic channel from H to S.*

Proof. Since $(D, H) \notin E$ and $(D, S) \notin E^+$, we have $(D, S) \notin E$. We distinguish two cases, depending on whether the edge (D, P) exists.

– $(D, P) \notin E$. Then there are no outgoing edges from D and the statement follows from Lemma A.

– $(D, P) \in E$. Then there is no edge from P to S, else there would be a path from D to S. It follows that there is no communication path from H to S, thus the protocol cannot provide a confidential nor an authentic channel from H to S.

Lemma 5. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology with $K_H = \emptyset$, $(H, D) \in E$, $(S, H) \in E^+$, and $(D, H) \notin E^+$. Then there exists no protocol for τ that provides a confidential and no protocol for τ that provides an authentic channel from S to H.*

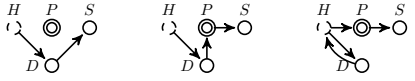
Proof. Since there is no edge from D to H in τ, there are only two possible paths from S to H, namely (S, P, H) and (S, D, P, H). The second path, however, is impossible in τ, because it contains a path from D to H. It follows that there is no outgoing edge from D in τ thus by Lemma A, there cannot be a protocol for τ that provides a confidential or an authentic channel from S to H.

B.2 Proof Details: Possibility results

The following lemmas show the existence of HISPs that provide secure channels between H and S for the topologies not covered by the impossibility results above. Our proofs embody protocols that we have verified using Tamarin. Note that in all protocols the human role H has an empty initial knowledge. Lemmas that first appear and are referenced only in this appendix are numbered with letters.

Lemma 6. *Let $\tau = (V, E, \eta, \mu)$ be any HISP topology with $(H, D) \in E$ and $(D, S) \in E^+$. Then there exists a protocol for τ that provides an originating secure channel from H to S even if $K_H = \emptyset$.*

Proof. The following graphs consist of an acyclic path from D to S and an additional edge $(H, D) \in E$.



(a) (b) (c)

We show a protocol for each of the three topologies.

(a) The following protocol communicates a message m, originating with H, authentically and confidentially from H to S using the path in case (a).

Protocol Lemma 6 (a)

$$H \bullet\!\!\rightarrow\!\!\bullet\ D : \mathrm{fresh}(m).\langle S, m\rangle$$
$$D \bullet\!\!\rightarrow\!\!\bullet\ S : \langle H, m\rangle$$

H first sends the fresh, secret message m together with the name of the intended recipient S to D using $H \bullet\!\!\rightarrow\!\!\bullet\ D$. Then, D passes the message and the sender's name H to S using $D \bullet\!\!\rightarrow\!\!\bullet\ S$.

(b) The following protocol transmits a message m, originating with H, authentically and confidentially from H to S using the path in case (b) and a secret key k shared between D and S. Recall that we specify the initial knowledge using knows(_) statements.

Protocol Lemma 6 (b)

$$\begin{aligned} D &: \text{knows}(\langle S, k\rangle) \\ S &: \text{knows}(\langle D, k\rangle) \\ H \bullet\!\!\rightarrow\!\!\bullet\ D &: \text{fresh}(m).\langle S, m\rangle \\ D \circ\!\!\rightarrow\!\!\circ\ P &: \text{senc}(\langle H, m\rangle, k)/\mathit{ciphertext} \\ P \circ\!\!\rightarrow\!\!\circ\ S &: \mathit{ciphertext}/\text{senc}(\langle H, m\rangle, k) \end{aligned}$$

The protocol executes as follows. H first sends the fresh, secret message m and the intended recipients's name S to D using $H \bullet\!\!\rightarrow\!\!\bullet\ D$. Then, D encrypts m and the sender's name H using k and sends the cipher-text to P. P sends the cipher-text to S where it is decrypted.

(c) The following protocol transmits a message m, originating with H, authentically and confidentially from H to S using the path in case (c) and a secret key k shared between D and S.

Protocol Lemma 6 (c)

$$\begin{aligned} D &: \text{knows}(\langle H, S, k\rangle) \\ S &: \text{knows}(\langle H, D, k\rangle) \\ H \bullet\!\!\rightarrow\!\!\bullet\ D &: \text{fresh}(m).\langle S, m\rangle \\ D \bullet\!\!\rightarrow\!\!\bullet\ H &: \langle m, \text{senc}(m, k)\rangle/\langle m, \mathit{ciphertext}\rangle \\ H \circ\!\!\rightarrow\!\!\circ\ P &: \mathit{ciphertext} \\ P \circ\!\!\rightarrow\!\!\circ\ S &: \mathit{ciphertext}/\text{senc}(m, k) \end{aligned}$$

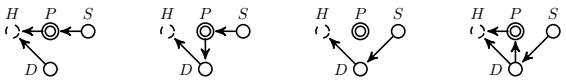
The protocol executes as follows. H first sends the fresh, secret message m and intended recipient's name S to D using $H \bullet\!\!\rightarrow\!\!\bullet\ D$. Then, D encrypts m using k and sends the cipher-text and the message back to H who inputs the cipher-text into P. P sends the cipher-text to S where it is decrypted.

We used Tamarin to prove that these protocols provide an originating secure communication channel from H to S.

Lemma 7 states that for HISP topologies containing an edge from D to H, there is a protocol providing a secure channel from S to H, if there is a path from S to H.

Lemma 7. *Let (V, E, η, μ) be a HISP topology with $(S, H) \in E^+$. If $(D, H) \in E$ then there exists a protocol that provides a secure channel from S to H even if $K_H = \emptyset$.*

Proof. The following are all acyclic paths from S to H together with an additional edge $(D, H) \in E$.



(a) (b) (c) (d)

We show a protocol for each of the first three topologies. Since case (d) is a supergraph of case (c), the protocol for case (c) also applies to case (d).

(a) The protocol is based on codebook cryptography, following [7]. The protocol below transmits a predefined message m securely from S to H.

Protocol Lemma 7 (a)

$$
\begin{aligned}
D &: \mathrm{knows}(\langle H, S, m, \mathrm{h}(m)\rangle) \\
S &: \mathrm{knows}(\langle H, D, m, \mathrm{h}(m)\rangle) \\
S &\circ\!\!\rightarrow\!\!\circ\ P : \mathrm{h}(m)/\mathit{hash} \\
P &\circ\!\!\rightarrow\!\!\circ\ H : \mathit{hash} \\
D &\bullet\!\!\rightarrow\!\!\bullet\ H : \langle S, m, \mathrm{h}(m)/\langle S, m, \mathit{hash}\rangle
\end{aligned}
$$

The hash function $\mathrm{h}(m)$ represents the mapping, shared between D and S, from a clear-text message m to the code. After the protocol's execution, H compares the code supposedly received from S with the tuple $\langle m, \mathrm{h}(m)\rangle$ received from D. This represents a lookup in the codebook.

(b) The following protocol provides an originating secure communication channel from S to H using a secret key k shared between D and S.

Protocol Lemma 7 (b)

$$
\begin{aligned}
D &: \mathrm{knows}(\langle H, S, k\rangle) \\
S &: \mathrm{knows}(\langle H, D, k\rangle) \\
S &\circ\!\!\rightarrow\!\!\circ\ P : \mathrm{fresh}(m).\mathrm{senc}(m, k)/\mathit{ciphertext} \\
P &\circ\!\!\rightarrow\!\!\circ\ D : \mathit{ciphertext}/\mathrm{senc}(m, k) \\
D &\bullet\!\!\rightarrow\!\!\bullet\ H : \langle S, m\rangle
\end{aligned}
$$

S first submits the fresh, secret message m encrypted with the key k to P using $S \circ\!\!\rightarrow\!\!\circ\ P$. P sends the cipher-text to D, who decrypts the message and sends m and its sender's name S to H using $D \bullet\!\!\rightarrow\!\!\bullet\ H$.

(c) The following protocol provides an originating secure communication channel from S to H using the secure links $S \bullet\!\!\rightarrow\!\!\bullet\ D$ and $D \bullet\!\!\rightarrow\!\!\bullet\ H$.

Protocol Lemma 7 (c)

$$
\begin{aligned}
S &\bullet\!\!\rightarrow\!\!\bullet\ D : \mathrm{fresh}(m).\langle H, m\rangle \\
D &\bullet\!\!\rightarrow\!\!\bullet\ H : \langle S, m\rangle
\end{aligned}
$$

S first submits the fresh, secret message m together with the intended recipient's name H to D using $S \bullet\!\!\rightarrow\!\!\bullet\ D$. Then, D passes m together with the sender's name S to H using $D \bullet\!\!\rightarrow\!\!\bullet\ H$.

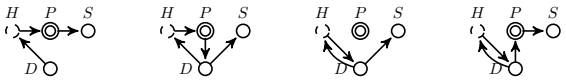
(d) The same protocol as for case (c) can be applied by omitting the additional edges $(D, P) \in E$ and $(P, H) \in E$.

We used Tamarin to prove that all three protocols above provide a secure communication channel from S to H.

Lemma 8 states that for HISP topologies containing an edge from D to H, there is a protocol providing a secure channel from H to S, if there is a path from H to S.

Lemma 8. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology with $(H, S) \in E^+$. If $(D, H) \in E$ then there exists a protocol for τ that provides a secure channel from H to S even if $K_H = \emptyset$.*

Proof. The following are all acyclic paths from H to S together with an additional edge $(D, H) \in E$.



(a) (b) (c) (d)

Cases (c) and (d) follow from Lemma 6. Protocols for the remaining cases (a) and (b) are given below.

The following protocols each communicate a predefined message m authentically and confidentially from H to S via the paths in case (a) and (b), respectively. The hash function $\mathrm{h}(m)$ represents the mapping from a clear-text message m to the code. At the end of Protocol 8 (a), S compares the code supposedly received from H with the corresponding tuple $\langle m, \mathrm{h}(m)\rangle$.

Protocol Lemma 8 (a)

$$
\begin{aligned}
D &: \mathrm{knows}(\langle H, S, m, \mathrm{h}(m)\rangle)\\
S &: \mathrm{knows}(\langle H, D, m, \mathrm{h}(m)\rangle)\\
D \bullet\!\!\rightarrow\!\!\bullet\ H &: \langle S, m, \mathrm{h}(m)\rangle \ /\ \langle S, m, hash\rangle\\
H \circ\!\!\rightarrow\!\!\circ\ P &: hash\\
P \circ\!\!\rightarrow\!\!\circ\ S &: hash \ /\ \mathrm{h}(m)
\end{aligned}
$$

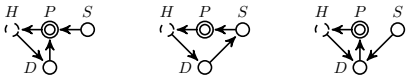
Protocol Lemma 8 (b)

$$
\begin{aligned}
D &: \mathrm{knows}(\langle H, S\rangle)\\
D \bullet\!\!\rightarrow\!\!\bullet\ H &: \mathrm{fresh}(m).\langle m, \mathrm{h}(m)\rangle \ /\ \langle m, hash\rangle\\
H \circ\!\!\rightarrow\!\!\circ\ P &: hash\\
P \circ\!\!\rightarrow\!\!\circ\ D &: hash \ /\ \mathrm{h}(m)\\
D \bullet\!\!\rightarrow\!\!\bullet\ S &: \langle H, m\rangle
\end{aligned}
$$

We used Tamarin to prove that both protocols provide a secure communication channel from H to S.

Lemma 9. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology with $(S, H) \in E^+$ and $(D, H) \notin E$. If $(H, D) \in E$ and $(D, H) \in E^+$, then there exists a protocol for τ that provides an originating authentic channel from S to H even if $K_H = \emptyset$.*

Proof. The minimal graphs satisfying the lemma's hypothesis are obtained as follows. There are two acyclic paths from S to H with $(D, H) \notin E$. One satisfies $(D, H) \in E^+$ and leads to case (c). The other leads to cases (a) and (b), since there are two acyclic paths form D to H with $(D, H) \notin E$.



(a) (b) (c)

The following protocols provide an originating authentic channel from S to H for the three topologies. In each of the protocols, S generates a fresh message m, which is communicated to H, thus they provide an originating channel. Tamarin proves that the channel is authentic.

Protocol Lemma 9 (a)

$$
\begin{aligned}
D &: \mathrm{knows}(\langle H, S, k\rangle)\\
S &: \mathrm{knows}(\langle H, D, k\rangle)\\
S \circ\!\!\rightarrow\!\!\circ\ P &: \mathrm{fresh}(m).\langle m, \mathrm{h}(\langle k, m\rangle)\rangle \ /\ \langle m, hash\rangle\\
P \circ\!\!\rightarrow\!\!\circ\ H &: \langle m, hash\rangle\\
H \bullet\!\!\rightarrow\!\!\bullet\ D &: \mathrm{fresh}(x).\langle S, x, m, hash\rangle \ /\ \langle S, x, m, \mathrm{h}(\langle k, m\rangle)\rangle\\
D \circ\!\!\rightarrow\!\!\circ\ P &: x\\
P \circ\!\!\rightarrow\!\!\circ\ H &: x
\end{aligned}
$$

Protocol Lemma 9 (b)

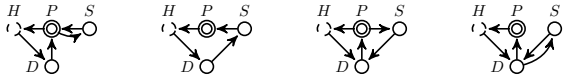
$$
\begin{aligned}
D &: \mathrm{knows}(\langle H, S, k\rangle)\\
S &: \mathrm{knows}(\langle H, D, k\rangle)\\
S \circ\!\!\rightarrow\!\!\circ\ P &: \mathrm{fresh}(m).\langle m, \mathrm{h}(\langle k, m\rangle)\rangle \ /\ \langle m, hash\rangle\\
P \circ\!\!\rightarrow\!\!\circ\ H &: \langle m, hash\rangle\\
H \bullet\!\!\rightarrow\!\!\bullet\ D &: \mathrm{fresh}(x).\langle x, m, hash\rangle \ /\ \langle x, m, \mathrm{h}(\langle k, m\rangle)\rangle\\
D \bullet\!\!\rightarrow\!\!\bullet\ S &: x\\
S \circ\!\!\rightarrow\!\!\circ\ P &: x\\
P \circ\!\!\rightarrow\!\!\circ\ H &: x
\end{aligned}
$$

Protocol Lemma 9 (c)

$$
\begin{aligned}
D &: \mathrm{knows}(\langle H, S\rangle)\\
S &: \mathrm{knows}(\langle H, D\rangle)\\
S &\bullet\!\!\rightarrow\bullet\, D : \mathrm{fresh}(m).m\\
D &\circ\!\!\rightarrow\!\!\circ\, P : m\\
P &\circ\!\!\rightarrow\!\!\circ\, H : m\\
H &\bullet\!\!\rightarrow\bullet\, D : \mathrm{fresh}(x).\langle S, x, m\rangle\\
D &\circ\!\!\rightarrow\!\!\circ\, P : x\\
P &\circ\!\!\rightarrow\!\!\circ\, H : x
\end{aligned}
$$

Lemma 10. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology with $K_H = \emptyset$, $(S, H) \in E^+$, $(D, H) \notin E$, $(H, D) \in E$, and $(D, H) \in E^+$. Then there exists a protocol for τ that provides a secure channel from S to H if and only if $(H, S) \in E^+$.*

Proof. The minimal graphs satisfying the lemma's hypothesis are obtained from the graphs of Lemma 9 and the additional condition $(H, S) \in E^+$.



(a) (b) (c) (d)

To see that $(H, S) \in E^+$ is necessary, suppose that $(H, S) \notin E^+$. The initial knowledge of H is empty and any fresh constant that H generates cannot be known to S because $(H, S) \notin E^+$. Any message that H receives is known to P because the only incoming edge to H is $(P, H) \in E$. Thus every message sent by S and learned by H can be learned by the adversary. It follows that every term t that can be derived from the knowledge of H using pairing and projection and that can be derived from the knowledge of S (using all functions in Σ), can also be derived using the knowledge of P. Thus there cannot be a protocol that provides a confidential channel and consequently there cannot be a protocol that provides a secure channel from S to H.

It remains to find a protocol that provides a secure channel from S to H for each of the four minimal topologies when $K_H = \emptyset$. The protocols are given below.

Protocol Lemma 10 (a)

$$
\begin{aligned}
D &: \mathrm{knows}(\langle H, S, \mathrm{h}(\langle k, D, S\rangle)\rangle)\\
S &: \mathrm{knows}(\langle H, D, \mathrm{h}(\langle k, D, S\rangle)\rangle)\\
H &\bullet\!\!\rightarrow\bullet\, D : \mathrm{fresh}(x_1, x_2).\langle S, x_1, x_2\rangle\\
D &\circ\!\!\rightarrow\!\!\circ\, P : \langle S, \mathrm{senc}(\langle x_1, x_2\rangle, \mathrm{h}(\langle k, D, S\rangle))\rangle \;/\; \langle S, \mathit{ciphertext}\rangle\\
P &\circ\!\!\rightarrow\!\!\circ\, S : \mathit{ciphertext} \;/\; \mathrm{senc}(\langle x_1, x_2\rangle, \mathrm{h}(\langle k, D, S\rangle))\\
S &\circ\!\!\rightarrow\!\!\circ\, P : x_2\\
P &\circ\!\!\rightarrow\!\!\circ\, H : x_2
\end{aligned}
$$

Protocol Lemma 10 (b)

$$
\begin{aligned}
D &: \text{knows}(\langle H, S\rangle)\\
S &: \text{knows}(\langle H, D\rangle)\\
H \bullet\!\!\rightarrow\!\!\bullet\ D &: \text{fresh}(x_1, x_2).\langle S, x_1, x_2\rangle\\
D \bullet\!\!\rightarrow\!\!\bullet\ S &: \langle H, x_1, x_2\rangle\\
S \circ\!\!\rightarrow\!\!\circ\ P &: x_2\\
P \circ\!\!\rightarrow\!\!\circ\ H &: x_2
\end{aligned}
$$

Protocol Lemma 10 (c)

$$
\begin{aligned}
D &: \text{knows}(\langle H, S, \text{h}(\langle k, D, S\rangle)\rangle)\\
S &: \text{knows}(\langle H, D, \text{h}(\langle k, D, S\rangle)\rangle)\\
H \bullet\!\!\rightarrow\!\!\bullet\ D &: \text{fresh}(x_1, x_2).\langle S, x_1, x_2\rangle\\
D \circ\!\!\rightarrow\!\!\circ\ P &: \langle S, \text{senc}(\langle x_1, x_2\rangle, \text{h}(\langle k, D, S\rangle))\rangle \ /\ \langle S, \mathit{ciphertext}\rangle\\
P \circ\!\!\rightarrow\!\!\circ\ S &: \mathit{ciphertext} \ /\ \text{senc}(\langle x_1, x_2\rangle, \text{h}(\langle k, D, S\rangle))\\
S \bullet\!\!\rightarrow\!\!\bullet\ D &: \langle H, x_1\rangle\\
D \circ\!\!\rightarrow\!\!\circ\ P &: x_2\\
P \circ\!\!\rightarrow\!\!\circ\ H &: x_2
\end{aligned}
$$

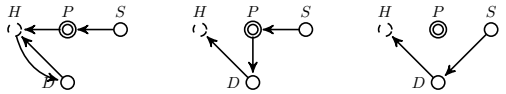
Protocol Lemma 10 (d)

$$
\begin{aligned}
D &: \text{knows}(\langle H, S\rangle)\\
S &: \text{knows}(\langle H, D\rangle)\\
H \bullet\!\!\rightarrow\!\!\bullet\ D &: \text{fresh}(x_1, x_2).\langle S, x_1, x_2\rangle\\
D \bullet\!\!\rightarrow\!\!\bullet\ S &: \langle H, x_1, x_2\rangle\\
S \bullet\!\!\rightarrow\!\!\bullet\ D &: \langle H, x_1\rangle\\
D \circ\!\!\rightarrow\!\!\circ\ P &: x_2\\
P \circ\!\!\rightarrow\!\!\circ\ H &: x_2
\end{aligned}
$$

We used Tamarin to prove that these protocols provide a secure communication channel from S to H.

Lemma B *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology with $(S, D) \in E^+$ and $(D, H) \in E$. Then there exists a protocol for τ that provides an originating secure channel from S to H even if $K_H = \emptyset$.*

Proof. Below are all acyclic paths from S to D together with an additional edge $(D, H) \in E$.



(a) (b) (c)

Case (c) is equal to case (c) in Lemma 7 where the given protocol already provides an originating secure channel from S to H. In the following we provide protocols for the remaining cases (a) and (b).

The following protocol provides an originating secure channel from S to H in case (a).

Protocol Lemma B (a)

$$
\begin{aligned}
D &: \text{knows}(\langle H, S, \text{h}(\langle k, D, S\rangle)\rangle)\\
S &: \text{knows}(\langle H, D, \text{h}(\langle k, D, S\rangle)\rangle)\\
S \circ\!\!\rightarrow\!\!\circ\ P &: \text{fresh}(m).\text{senc}(\langle S, D, H, m\rangle, \text{h}(\langle k, D, S\rangle))\ /\\
&\quad\ ciphertext\\
P \circ\!\!\rightarrow\!\!\circ\ H &: ciphertext\\
H \bullet\!\!\rightarrow\!\!\bullet\ D &: ciphertext\ /\ \text{senc}(\langle S, D, H, m\rangle, \text{h}(\langle k, D, S\rangle))\\
D \bullet\!\!\rightarrow\!\!\bullet\ H &: \langle S, \text{senc}(\langle S, D, H, m\rangle, \text{h}(\langle k, D, S\rangle)), m\rangle\ /\\
&\quad\ \langle S, ciphertext, m\rangle
\end{aligned}
$$

For case (b), we adapt the protocol as follows.

Protocol Lemma B (b)

$$
\begin{aligned}
D &: \text{knows}(\langle H, S, \text{h}(\langle k, D, H, S\rangle)\rangle)\\
S &: \text{knows}(\langle H, D, \text{h}(\langle k, D, H, S\rangle)\rangle)\\
S \circ\!\!\rightarrow\!\!\circ\ P &: \text{fresh}(m).\text{senc}(\langle S, D, H, m\rangle, \text{h}(\langle k, D, H, S\rangle))\ /\\
&\quad\ ciphertext\\
P \circ\!\!\rightarrow\!\!\circ\ D &: ciphertext\ /\ \text{senc}(\langle S, D, H, m\rangle, \text{h}(\langle k, D, H, S\rangle))\\
D \bullet\!\!\rightarrow\!\!\bullet\ H &: \langle S, m\rangle
\end{aligned}
$$

In both cases, S first freshly generates the secret message m and sends it encrypted with the key k to P using $S \circ\!\!\rightarrow\!\!\circ\ P$. P sends the message to H in case (a) or directly to D in case (b). The message is decrypted by D and sent to H using $D \bullet\!\!\rightarrow\!\!\bullet\ H$.

We used Tamarin to prove that these protocols provide a secure communication channel from S to H.

Lemma C *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology with $K_H = \emptyset$, $(D, H) \in E$, $(H, D) \notin E$, and $(H, S) \in E^+$. If there is an incoming edge to D, then there exists a protocol for τ that provides an originating authentic channel from H to S.*

Proof. There must be an edge $(H, P) \in E$ because $(D, H) \in E$, $(H, D) \notin E$, and $(H, S) \in E^+$. Since D has an incoming edge, it must have either an incoming edge from P or one from S. The first of the following two protocols provides an originating authentic channel in the former case, and the second in the latter case.

Protocol Lemma C (a)

$$
\begin{aligned}
D &: \mathrm{knows}(\langle H, S, k\rangle)\\
S &: \mathrm{knows}(\langle H, D, k\rangle)\\
H &\circ\!\!\rightarrow\!\!\circ\ P : \mathrm{fresh}(m).m\\
P &\circ\!\!\rightarrow\!\!\circ\ D : m\\
D &\bullet\!\!\rightarrow\!\!\bullet\ H : \langle m, \mathrm{h}(\langle k, m, S, D, H\rangle)\rangle \ /\ \langle m, hash\rangle\\
H &\circ\!\!\rightarrow\!\!\circ\ P : hash\\
P &\circ\!\!\rightarrow\!\!\circ\ S : \langle m, hash\rangle \ /\ \langle m, \mathrm{h}(\langle k, m, S, D, H\rangle)\rangle
\end{aligned}
$$

Protocol Lemma C (b)

$$
\begin{aligned}
D &: \mathrm{knows}(\langle H, S, k\rangle)\\
S &: \mathrm{knows}(\langle H, D, k\rangle)\\
H &\circ\!\!\rightarrow\!\!\circ\ P : \mathrm{fresh}(m).m\\
P &\circ\!\!\rightarrow\!\!\circ\ S : m\\
S &\bullet\!\!\rightarrow\!\!\bullet\ D : \langle m, \mathrm{h}(\langle k, m\rangle)\rangle\\
D &\bullet\!\!\rightarrow\!\!\bullet\ H : \langle m, S, \mathrm{h}(\langle k, m\rangle)\rangle \ /\ \langle m, S, hash\rangle\\
H &\circ\!\!\rightarrow\!\!\circ\ P : hash\\
P &\circ\!\!\rightarrow\!\!\circ\ S : hash \ /\ \mathrm{h}(\langle k, m\rangle)
\end{aligned}
$$

B.3 Proof Details: Proofs of Theorems

Theorem 1. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology. There exists a protocol for τ that provides an originating confidential channel from H to S if and only if $(H, D) \in E$ and $(D, S) \in E^+$.*

Proof. Let τ be a HISP topology such that $(H, D) \in E$ and $(D, S) \in E^+$. By Lemma 6, there exists a protocol for τ that provides an originating confidential channel H to S for each of the three acyclic paths from D to S.

Conversely, let $\mathcal{R}$ be a protocol for τ that provides an originating confidential channel H to S. Then there is a trace in which a fresh constant m originating with H is transmitted to S. Thus there must be a path from H to S. Suppose $(H, D) \notin E$. Then the only outgoing edge from H is $(H, P) \in E$. Since H can only perform pairing and projection, any fresh constant m generated by H can only be paired with other terms. Thus, if H sends a message of which m is a subterm, the adversary can learn m. Thus there must be an edge from H to D and a path from D to S.

Theorem 2. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology. Then there exists a protocol for τ that provides an originating authentic channel from H to S if and only if $(H, S) \in E^+$, there exists an edge between H and D, and there exists an edge incoming to D as well as an edge outgoing from D.*

Proof. We first show that the topological conditions are necessary for the existence of a protocol providing an originating authentic channel. It is obvious that $(H, S) \in E^+$ is a necessary condition for a protocol to provide a communication

channel from H to S. We show by case distinction that there must be an edge incoming to D as well as an edge outgoing from D.

1. All edges adjacent to D are outgoing from D.
   Then D never learns any fresh constant m generated by H. Thus m is never a subterm of any message sent from D to H. Thus for every message m' sent from H to P, the adversary may compute all projections of m' and substitute each m by a fresh constant $\tilde{m}$, then pair the terms up again. For all messages received by H from P, the adversary replaces in the same manner all projections to $\tilde{m}$ by m. Thus S learns $\tilde{m}$ whereas H sends m. Since m originates with H, S cannot distinguish between terms involving m and terms involving $\tilde{m}$. Since H cannot perform any functions other than pairing and projections, H cannot distinguish terms that are obtained by applying any other function to m from terms that are obtained by applying such functions to $\tilde{m}$. It follows that there is no protocol that provides an originating authentic channel.
2. There are no edges to or from D.
   If there is a protocol that provides an originating authentic channel when there are no edges to or from D, then there is one in which there are outgoing edges from D. This contradicts Case 1.
3. All edges adjacent to D are incoming to D.
   If all edges adjacent to D are incoming to D, then there is no protocol that provides an originating authentic channel, otherwise there would be one without a role specification for D which is impossible by Case 2.

If there are no edges between D and H, we can combine the roles of D and S, since H communicates with both through P. Then, by the reasoning in Case 1 above, there cannot be an originating authentic channel from H to S.

Conversely, consider all the HISP topologies such that $(H, S) \in E^+$ and there exists an edge between H and D and there exists an edge incoming to D as well as an edge outgoing from D. There are two types of protocols that provide an originating authentic channel from H to S, depending on the edge(s) between H and D.

- $(H, D) \in E$. Since there is a path $(H, S) \in E^+$ and an outgoing edge from D, there must be a path $(D, S) \in E^+$. It follows from Lemma 6 that there exists a protocol that provides an originating authentic channel from H to S.
- $(D, H) \in E$ and $(H, D) \notin E$. Then there exists a protocol that provides an originating authentic channel from H to S by Lemma C.

Theorem 3. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology. There exists a protocol for τ that provides an originating confidential channel from S to H if and only if $(D, H) \in E$ and $(S, D) \in E^+$.*

Proof. Let τ be a HISP topology such that $(D, H) \in E$ and $(S, D) \in E^+$. By Lemma B, there is a protocol for τ that provides an originating confidential channel S to H for each of the three acyclic paths from S to D.

Conversely, let $\mathcal{R}$ be a protocol for τ that provides an originating confidential channel S to H. Then there is a trace in which a fresh constant m originating with S is transmitted to H. Thus there must be a path from S to H. Suppose $(D, H) \notin E$. Then the only incoming edge to H is $(P, H) \in E$. Since H can only perform pairing and projection, any fresh constant m learned, but not generated by H can only be learned as a singleton or paired with other terms. Thus, if H receives a message of which m is a subterm and H learns m, then the adversary can learn m. Thus there must be an edge from D to H. Suppose now that there is no path $(S, D) \in E^+$. Then there are only outgoing edges from D, because there is a path S to H and an edge (D, H). Thus m is not in D's knowledge, since it originates with S and there is no communication path from S to D. Thus, as above, since H can only perform pairing and projecting of terms, any fresh constant m learned by H and generated by S can be learned by the adversary. Thus there must be a path $(S, D) \in E^+$.

Theorem 4. *Let $\tau = (V, E, \eta, \mu)$ be a HISP topology. Then there exists a protocol for τ that provides an originating authentic channel from S to H if and only if $(S, H) \in E^+$, there exists an edge between H and D, and there exists an edge incoming to D as well as an edge outgoing from D.*

Proof. We first show that the topological conditions are necessary for the existence of a protocol providing an originating authentic channel. It is obvious that $(S, H) \in E^+$ is a necessary condition for a protocol to provide a communication channel from S to H. We show by case distinction that there must be an edge incoming to D as well as an edge outgoing from D.

1. All edges adjacent to D are outgoing from D.
   Then D never learns any fresh constant m generated by S. Thus m is never a subterm of any message sent from D to H. Thus for every message m' sent from S to P, the adversary may compute all projections of m' and substitute each m by a fresh constant $\tilde{m}$, then pair the terms up again. For all messages sent by H to P, the adversary replaces in the same manner all projections to $\tilde{m}$ by m. Thus H learns $\tilde{m}$ whereas S sends m. Since m originates with S, H cannot distinguish between terms involving m and terms involving $\tilde{m}$. Since H cannot perform any functions other than pairing and projections, H cannot distinguish terms that are obtained by applying any other function to m from terms that are obtained by applying such functions to $\tilde{m}$. It follows that there is no protocol that provides an originating authentic channel.
2. There are no edges to or from D.
   If there is a protocol that provides an originating authentic channel when there are no edges to or from D, then there is one in which there are outgoing edges from D. This contradicts Case 1.
3. All edges adjacent to D are incoming to D.
   If all edges adjacent to D are incoming to D, then there is no protocol that provides an originating authentic channel, otherwise there would be one without a role specification for D which is impossible by Case 2.

If there are no edges between D and H, we can combine the roles of D and S. Then, by the reasoning in Case 1 above, there cannot be an originating authentic channel from S to H.

Conversely, consider all the HISP topologies such that $(S, H) \in E^+$ and there exists an edge between H and D and there exists an edge incoming to D as well as an edge outgoing from D. There are two types of protocols that provide an originating authentic channel from S to H, depending on the edge(s) between H and D.

- $(D, H) \in E$. Since there is a path $(S, H) \in E^+$ and an incoming edge to D, there must be a path $(S, D) \in E^+$. It follows from Lemma B that there exists a protocol providing an originating authentic channel from S to H.
- $(H, D) \in E$ and $(D, H) \notin E$. Then there must be a path $(D, H) \in E^+$, since there is an outgoing edge from D. By Lemma 9, all such HISP topologies have a protocol that provides an originating authentic channel from S to H.