Transformations between Cryptographic Protocols*

Joshua D. Guttman

The MITRE Corporation

Abstract. A transformation F between protocols associates the messages sent and received by participants in a protocol Π_1 with messages sent and received in some Π_2. Transformations are useful for modeling protocol design, protocol composition, and the services that protocols provide.

A protocol transformation determines a map from partial behaviors $\mathbb{A}_1$ of Π_1—which we call "skeletons"—to skeletons $F(\mathbb{A}_1)$ of Π_2. Good transformations should act as functors, preserving homomorphisms (information-preserving maps) from one Π_1-skeleton to another. Thus, if $H\colon \mathbb{A}_1 \mapsto \mathbb{A}_2$ is a homomorphism between Π_1-skeletons, then there should be a homomorphism $F(H)\colon F(\mathbb{A}_1) \mapsto F(\mathbb{A}_2)$ between their images in Π_2.

We illustrate protocol transformation by examples, and show that our definition ensures that transformations act as functors.

1 Introduction

A protocol transformation F from a protocol Π_1 to a protocol Π_2 maps message transmissions and receptions of roles of Π_1 to transmissions and receptions of roles of Π_2. F may be used to show how Π_1 and Π_2 exhibit one of a number of relations, among them:

- Π_1 may be a choreography—typically defined without cryptography—which the cryptography protocol Π_2 is intended to implement faithfully, despite the presence of malicious parties;
- Π_1 may be a stage in designing the fuller protocol Π_2;
- Π_1 may be a simplification of an implemented protocol Π_2 [12]; or
- Π_2 may be a composition of Π_1 with other protocols [10, 2, 8].

In this paper, our goals are to give:

1. a definition of transformations flexible enough for these purposes, which requires us to avoid giving an excessively syntactic criterion;
2. examples to show how it accommodates these purposes; and
3. three key results ensuring that protocol transformations are well-behaved.

* Supported by the MITRE-Sponsored Research program.

Skeletons and the homomorphisms between skeletons are central to our approach. We have previously used them to develop an algorithm for protocol analysis [6]; to prove this algorithm covers all the possibilities [5]; and to give a criterion ensuring that a protocol composition will respect the security goals achieved by the protocols being combined [8].

A *skeleton* $\mathbb{A}$ *describes* the behavior of the regular (non-adversarial) participants in some set of executions. It contains some regular behavior; the executions $\mathbb{A}$ describes must also contain this behavior. Transmission and reception events are related by a partial order expressing causal precedence. The skeleton may stipulate that some keys are assumed uncompromised, and that some keys or other (nonce-like) values are freshly generated. Some skeletons are *realized* in the sense that they contain all the regular behavior needed for a complete execution. This means that for every message that the skeleton says was delivered, either this message should itself have been previously transmitted, or else the adversary can synthesize it using its own creations and messages that were previously sent. Naturally, adversary behavior must respect the freshness and noncompromise assumptions of the skeleton. A protocol transformation F determines a map lifting skeletons $\mathbb{A}$ of Π_1 to skeletons $F(\mathbb{A})$ of Π_2.

A *homomorphism* is an information-preserving map $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ between skeletons of one protocol. The executions that a skeleton $\mathbb{A}_0$ *describes* are all the *realized* skeletons $\mathbb{A}_r$ such that, for some homomorphism H, $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_r$. An information-preserving map $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ cannot increase (loosen) the set of executions described; instead, H's target $\mathbb{A}_1$ should describe a subset of the executions described by its source $\mathbb{A}_0$. This follows from the fact that the composition of homomorphisms is a homomorphism.

The three theorems of this paper state:

1. A transformation acts as a functor (Thm. 1). If H is a homomorphism $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ on skeletons of Π_1, F should determine a (unique) homomorphism between their images in Π_2, i.e. $F(H)\colon F(\mathbb{A}_0) \mapsto F(\mathbb{A}_1)$.
2. F, regarded as a functor on homomorphisms, does not introduce new ones (Thm. 2). If $G\colon F(\mathbb{A}_0) \mapsto F(\mathbb{A}_1)$ is a homomorphism on Π_2 skeletons that are the images of Π_1 skeletons $\mathbb{A}_0, \mathbb{A}_1$, then $G = F(H)$ for some $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$.
3. If $\mathbb{B}$ is any Π_2 skeleton, then there is a (unique) maximum $\mathbb{A}_1$ such that $F(\mathbb{A}_1)$ may be mapped injectively into $\mathbb{B}$ (Thm. 3). $\mathbb{A}_1$ defines the maximum Π_1 content that is contained in $\mathbb{B}$.

In all of these results, uniqueness is to within isomorphism.

Our criterion on transformations F involves, besides a few bookkeeping constraints, three main requirements. First, suppose a parameter of a role ρ_1 of Π_1 *originates* on its image $F(\rho_1)$. This means that it is transmitted without having previously been received. Then this parameter should also have originated on ρ_1. Second, is an analogous condition on parameters of a role ρ_1 of Π_1 that are simply transmitted on its image $F(\rho_1)$, rather than being used only as keys for encryption or decryption. Finally, suppose that two roles ρ_0, ρ_1 of Π_1 have a substitution instance s in common. Then their images $F(\rho_0), F(\rho_1)$ in Π_2 should have in common the image $F(s)$. This condition is necessary to ensure

that $F(\rho_i)$ does not commit to one branch of a branching behavior while ρ_0 and ρ_1 are uncommitted.

Section 2 presents a variety of examples. In Section 3, we pause to summarize the current strand space treatment of the algebra of messages, skeletons, and homomorphisms, following [6, 8]. Section 4 presents the main definition and shows that $F(\mathbb{A})$ is well determined, and Section 5 shows that homomorphisms are preserved. Section 6 mentions some key related work and concludes.

2 Some Examples

We discuss here several variants of a transaction in which a principal A answers a question asked by another principal B. First, in the choreography description [1],

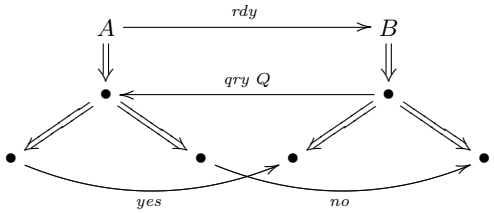


Fig. 1. A Question-Answering Choreography

Fig. 1, A starts by sending a message saying that he is ready; B responds with a query message containing the parameter Q, representing the question; and then A branches. Either A decides to answer the question affirmatively, and sends the constant *yes*, or else A decides to answer the question negatively, and sends the constant *no*.

The local behaviors of the individual parties are of four kinds. Two of them, the answerer A's behaviors C_1, C_2, are shown in Fig. 2. We refer to the local sequence of actions of one principal in one run of the protocol—possibly incomplete—as a *strand*. The questioner B's strands C_3, C_4 are dual, interchanging message sends and receives. In C_1, the final message transmission is the af-

Fig. 2. Question-Answering Choreography: A's Parametric Strands

firmative form *yes* while in C_2, it is *no*. In either case, the query can carry any value of the parameter Q. A strand which is only partly complete—because only the first step or the first two steps have occurred—is the same whether it is a C_1 behavior or a C_2 behavior. The execution has not yet committed to one branch or the other if it has not yet performed the last step. When one observes only the first two nodes of any of these behaviors, one cannot distinguish whether the last node will be affirmative or negative.

The matching behaviors C_3, C_4 are not shown since they are simply the duals.

A may later bill B a fee for this service. With so valuable a service, there are of course security considerations. A may not want to disclose to an adversary sniffing the network the answer to B's question; B may not want to disclose what question he is asking; and both parties may wish to ensure that the adversary cannot alter B's question Q to a different question Q', or alter A's answer from affirmative to negative or the reverse. To defeat these (standard) attacks, we may implement our choreography using a cryptographic protocol (Fig 3). Here, each principal is assumed to know a public (asymmetric) encryption key to which only the partner holds the matching decryption key; we write t encrypted in a form that only principal P can read as $\{|t|\}_P$. The parameters R, Y, N are nonces,

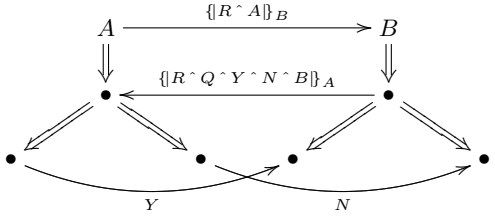


Fig. 3. Question-Answering Cryptoprotocol

i.e. randomly chosen bitstrings that are extremely unlikely to be used more than once. The last message is unencrypted. A transmits the first of the two nonces Y and N created by B to indicate the answer *yes*, and the second to indicate the answer *no*. This conveys the answer to B, while conveying nothing whatever to any principal that did not prepare $\{|R \hat{} Q \hat{} Y \hat{} N \hat{} B|\}_A$ and cannot decrypt it. The nonces themselves are perfectly arbitrary.

Each local behavior of the principal A is summarized in one of the strands in Fig. 4. Each one has the parameters A, B, R, Q, Y, N, and the *instances* of each strand are generated by substituting each possible name, nonce, or question for these parameters, respecting types. Again, the strands P_3, P_4 of B are dual.

Looking at the strands of Figs. 2, 4, we can see that we could perform a mapping in either direction. One could map the transmission and reception events of the cryptoprotocol strands to send and receive events in the choreography, discarding superfluous parameters. In this scheme, each j^{th} node of P_i would map to

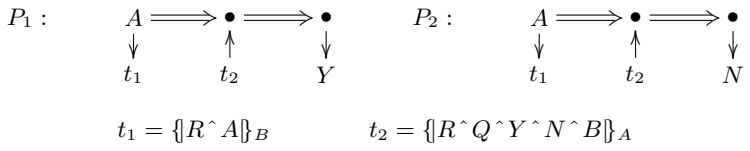


Fig. 4. Question-Answering Cryptoprotocol: Parametric Strands

the j^{th} node of C_i. Alternatively, one could map the events of the choreography to events on strands of the cryptoprotocol, allowing the additional parameters to be freely chosen. In this scheme, each j^{th} node of C_i would map to the j^{th} node of P_i.

In either case, we will map nodes on answerer strands at one level to nodes on answerer strands at the other level, and questioner nodes to questioner nodes at the other level. We will map affirmative nodes at one level to affirmative nodes at the other, and likewise for negative nodes. Along each strand, each successive transmission or reception must be associated with the corresponding transmission or reception of the strand at the other level. Thus, our only choice is whether to map from P_i to the C_i, or vice versa.

This example is, however, peculiar in that the two protocols have corresponding strands of the same length. Consider next the choreography of Fig. 5. It is more natural than the choreography of Fig. 1, since the questioner acts as client and the answerer acts as server; the *rdy* message at the beginning of Fig. 1 inverts the client-server roles. However, in Fig. 5, we can no longer map the j^{th} node of C_i' to the j^{th} node of P_i. Instead, it corresponds to the $j + 1^{\text{st}}$ node of P_i. For this same reason, we cannot speak of a transformation in the other direction; since the $C_i \mapsto P_i$ mapping is not surjective, the $P_i \mapsto C_i$ mapping would not be total.

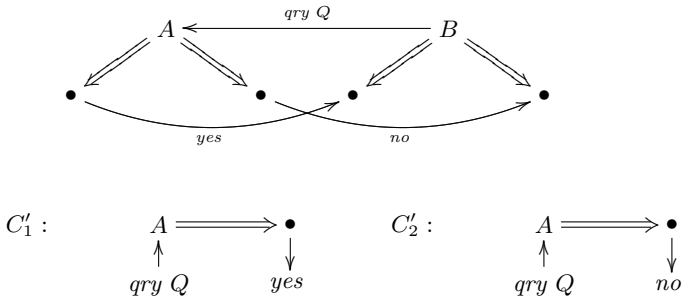


Fig. 5. A Question-Answering Service and Its Strands

Moreover, we can also think of protocol transformations where both source and target protocols involve cryptography. For instance, we can regard the cryptoprotocol of Fig. 3 as derived from the choreography of Fig. 5 by a succession of steps that progressively introduce suitable cryptography. As a first step, we may introduce the first readying message of Fig. 3, with the nonce R returned piggybacked with the question Q. We include these two elements within an encryption so that an adversary cannot reassociate the nonce R with a different question Q'. Thus, we obtain the intermediate form given in Fig. 6. The first

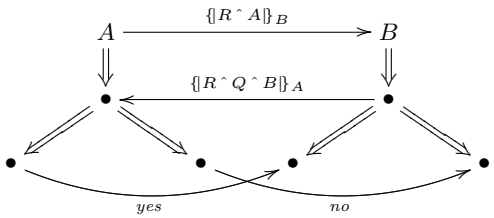


Fig. 6. From Choreography to Cryptoprotocol, Step 1

two messages provide an "outgoing test" [6] that allows A to authenticate this question Q as having been submitted by B. If the private decryption keys of A, B are uncompromised, then only B could liberate R from $\{|R \hat{} A|\}_B$, and moreover no third party could liberate R from any other message $\{|R \hat{} Q' \hat{} B|\}_A$. Thus, no third party could have repackaged R with the Q that A actually received.

Using the outgoing test idea again, B can supply a new nonce Y within the second message $\{|R \hat{} Q \hat{} Y \hat{} B|\}_A$; A may liberate this nonce to indicate an affirmative answer to Q (Fig 7). Repeating this idea with another nonce N for

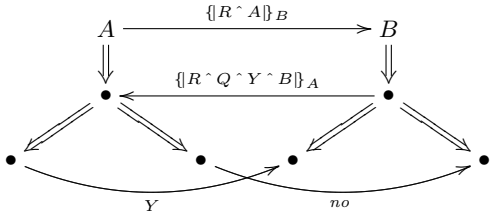


Fig. 7. From Choreography to Cryptoprotocol, Step 2

the negative case completes the protocol design process for Fig. 3.

Indeed, we can express the abstract schema of an "outgoing test" [6], in which a nonce is transmitted in encrypted form and received back later outside of that encryption, as a sort of germ protocol (Fig. 8). It allows the sender to infer that either the decryption key is compromised, or else a regular participant has received the encrypted unit and transformed its content. If we perform a

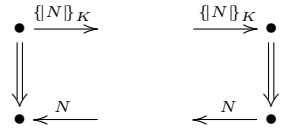


Fig. 8. The Outgoing (Nonce) Authentication Test

renaming here, mapping N to R and K to $\mathsf{pubk}(B)$, then we can map the left strand to the first two nodes on the left of Fig. 6 and the right strand to the first two nodes on the right. This appears to show Fig. 6 as a sort of "join" of the choreography of Fig. 5 with an instance of the outgoing test protocol. We can now repeat this process with a different renaming of Fig. 8 to produce Fig. 7 as a successive join, producing the final cryptoprotocol in a similar third step. These maps appear to characterize how the protocol of Fig. 4 achieves its goals.

3 Messages, Protocols, Skeletons

In this section, we provide an overview of the current strand space framework; cf. [8] or the extended version of [6] for more detail. Let $\mathfrak{A}_0$ be an algebra equipped with some operators and a set of homomorphisms $\eta\colon \mathfrak{A}_0 \to \mathfrak{A}_0$. We call members of $\mathfrak{A}_0$ *atoms*.

For the sake of definiteness, we will assume here that $\mathfrak{A}_0$ is the disjoint union of infinite sets of *nonces*, *atomic keys*, *names*, and *texts*. The operator $\mathsf{sk}(a)$ maps names to (atomic) signature keys, and K^{-1} maps an asymmetric atomic key to its inverse, and a symmetric atomic key to itself. Homomorphisms η are maps that respect sorts, and act homomorphically on $\mathsf{sk}(a)$ and K^{-1}.

Let X be an infinite set disjoint from $\mathfrak{A}_0$; its members—called *indeterminates*—act like unsorted variables. $\mathfrak{A}$ is freely generated from $\mathfrak{A}_0 \cup X$ by two operations: encryption $\{|t_0|\}_{t_1}$ and tagged concatenation $tag\ t_0\,\hat{}\,t_1$, where the tags tag are drawn from some set TAG. For a distinguished tag nil, we write $nil\ \ t_0\,\hat{}\,t_1$ as $t_0\,\hat{}\,t_1$ with no tag. In $\{|t_0|\}_{t_1}$, a non-atomic key t_1 is a symmetric key. Members of $\mathfrak{A}$ are called *messages*.

A homomorphism $\alpha = (\eta, \chi)\colon \mathfrak{A} \to \mathfrak{A}$ consists of a homomorphism η on atoms and a function $\chi\colon X \to \mathfrak{A}$. It is defined for all $t \in \mathfrak{A}$ by the conditions:

$$\begin{array}{ll} \alpha(a) = \eta(a), & \text{if } a \in \mathfrak{A}_0 \\ \alpha(x) = \chi(x), & \text{if } x \in X \end{array} \qquad \begin{array}{c} \alpha(\{|t_0|\}_{t_1}) = \{|\alpha(t_0)|\}_{\alpha(t_1)} \\ \alpha(tag\ t_0\,\hat{}\,t_1) = tag\ \alpha(t_0)\,\hat{}\,\alpha(t_1) \end{array}$$

Thus, atoms serve as typed variables, replaceable only by other values of the same sort, while indeterminates x are untyped. Indeterminates x serve as blank slots, to be filled by any $\chi(x) \in \mathfrak{A}$. Indeterminates and atoms are jointly *parameters*.

This $\mathfrak{A}$ has the most general unifier property, which we will rely on. That is, suppose that for $v, w \in \mathfrak{A}$, there exist α, β such that $\alpha(v) = \beta(w)$. Then there are α_0, β_0, such that $\alpha_0(v) = \beta_0(w)$, and whenever $\alpha(v) = \beta(w)$, then α and β are of the forms $\gamma \circ \alpha_0$ and $\gamma \circ \beta_0$.

Messages are abstract syntax trees in the usual way:

1. Let ℓ and r be the partial functions such that for $t = \{|t_1|\}_{t_2}$ or $t = tag\ t_1\hat{}t_2$, $\ell(t) = t_1$ and $r(t) = t_2$; and for $t \in \mathfrak{A}_0$, ℓ and r are undefined.
2. A *path* p is a sequence in $\{\ell, r\}^*$. We regard p as a partial function, where $\langle\rangle = \mathsf{Id}$ and $\mathsf{cons}(f, p) = p \circ f$. When the rhs is defined, we have: 1. $\langle\rangle(t) = t$; 2. $\mathsf{cons}(\ell, p)(t) = p(\ell(t))$; and 3. $\mathsf{cons}(r, p)(t) = p(r(t))$.
3. p *traverses a key edge* in t if $p_1(t)$ is an encryption, where $p = p_1{}^\frown\langle r\rangle{}^\frown p_2$.
4. p *traverses a member of* S if $p_1(t) \in S$, where $p = p_1{}^\frown p_2$ and $p_2 \neq \langle\rangle$.
5. t_0 *is an ingredient of* t, written $t_0 \sqsubseteq t$, if $t_0 = p(t)$ for some p that does not traverse a key edge in t.
6. t_0 *appears in* t, written $t_0 \ll t$, if $t_0 = p(t)$ for some p.

Strands and origination. A single local session of a protocol at a single principal is a *strand*, containing a linearly ordered sequence of transmissions and receptions that we call *nodes*. In Figs. 2, 4, and 5, the vertical columns of nodes connected by double arrows $\Rightarrow$ are strands.

We write $s \downarrow i$ for the i^{th} node on strand s, using 1-based indexing. We write $n \Rightarrow m$ when n, m are successive nodes on the same strand, i.e. when for some s, i, $n = s \downarrow i$ and $m = s \downarrow i + 1$. We write $\mathsf{msg}(n)$ for the message sent or received on the node n.

A message t_0 *originates* at a node n_1 if (1) n_1 is a transmission node; (2) $t_0 \sqsubseteq \mathsf{msg}(n_1)$; and (3) whenever $n_0 \Rightarrow^+ n_1$, $t_0 \not\sqsubseteq \mathsf{msg}(n_0)$.

Thus, t_0 originates when it was transmitted without having been either received or transmitted previously on the same strand. Values assumed to originate only on one node in an execution—*uniquely originating* values—formalize the idea of freshly chosen, unguessable values. Values assumed to originate nowhere may be used to encrypt or decrypt, but are never sent as message ingredients. They are called *non-originating* values. For a non-originating value K, $K \not\sqsubseteq t$ for any transmitted message t. However, $K \ll \{|t_0|\}_K \sqsubseteq t$ possibly, which is why we distinguish $\sqsubseteq$ from $\ll$. See [11, 6] for more details.

In the tree model of messages, to apply a homomorphism, we walk through, copying the tree, but inserting $\alpha(a)$ every time an atom a is encountered, and inserting $\alpha(x)$ every time that an indeterminate x is encountered.

Protocols. A *protocol* Π is a finite set of strands, representing the roles of the protocol. Four of the roles of the yes-no cryptoprotocol are the strands shown in Fig. 4. Their instances result by replacing A, B, R, Q, Y, N, etc., by any names, nonce, question, etc. The fifth role is the *listener* role $\mathsf{Lsn}[y]$ with a single reception node in which y is received. The instances of $\mathsf{Lsn}[y]$ are used to document

that values are available without cryptographic protection. For instance, $\mathsf{Lsn}[K]$ would document that K is compromised. Every protocol contains the role $\mathsf{Lsn}[y]$.

Indeterminates represent syntactically unconstrained messages received from protocol peers, or passed down as parameters from higher-level protocols. Thus, we require an indeterminate to be received as an ingredient before appearing in any other way:

If n_1 is a node on $\rho \in \Pi$, with an indeterminate $x \ll \mathsf{msg}(n_1)$,
then $\exists n_0$, $n_0 \Rightarrow^* n_1$, where n_0 is a reception node and $x \sqsubseteq \mathsf{msg}(n_0)$.

A principal executing a role such as P_3 in Fig. 4 may be partway through its run; for instance, it may have executed the first reception event and first transmission, without "yet" having executed the final reception event. Thus, it can not yet be distinguished from the first two nodes of an instance of role P_4. When $n_1 \Rightarrow n_2$ is the beginning of a strand $\alpha(P_3)$, we also regard it as an instance of the role P_4, since nothing that has happened shows that it is not:

Definition 1. *Node n is a* role node *of Π if $n = \rho \downarrow j$ lies on some $\rho \in \Pi$.*

Let $\rho \downarrow j$ be a role node of Π. Node m_j is an instance *of $\rho \downarrow j$ if, for some homomorphism α, the strand of m_j, up to m_j, takes the form: $\alpha(\rho \downarrow 1) \Rightarrow \ldots \Rightarrow \alpha(\rho \downarrow j) = m_j$.* □

That is, messages and their directions—transmission or reception—must agree up to node j. However, any remainders of the two strands beyond node j are unconstrained. They need not be compatible. When a protocol allows a principal to decide between different behaviors after step j, based on the message contents of their run, then this definition represents branching [7, 9]. At step j, one doesn't yet know which branch will be taken.

If s is a strand, then $t_0 \ll s$ iff for some j, $t_0 \ll \mathsf{msg}(s \downarrow j)$; and similarly $t_0 \sqsubseteq s$ iff for some j, $t_0 \sqsubseteq \mathsf{msg}(s \downarrow j)$.

Skeletons. A *skeleton* $\mathbb{A}$ consists of (possibly partially executed) role instances, i.e. a finite set of nodes, $\mathsf{nodes}(\mathbb{A})$, with two additional kinds of information:

1. A partial ordering $\preceq_{\mathbb{A}}$ on $\mathsf{nodes}(\mathbb{A})$;
2. Finite sets $\mathsf{unique}_{\mathbb{A}}, \mathsf{non}_{\mathbb{A}}$ of atomic values assumed uniquely originating and respectively non-originating in $\mathbb{A}$.

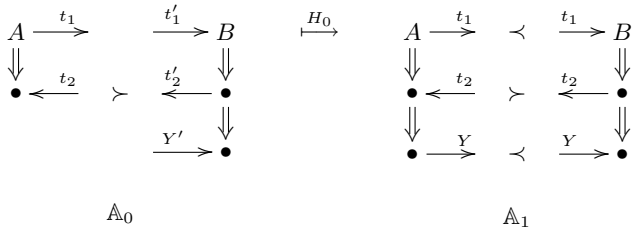
$\mathsf{nodes}(\mathbb{A})$ and $\preceq_{\mathbb{A}}$ must respect the strand order, i.e. if $n_1 \in \mathsf{nodes}(\mathbb{A})$ and $n_0 \Rightarrow n_1$, then $n_0 \in \mathsf{nodes}(\mathbb{A})$ and $n_0 \preceq_{\mathbb{A}} n_1$. If $a \in \mathsf{non}_{\mathbb{A}}$, then a must originate nowhere in $\mathsf{nodes}(\mathbb{A})$, though a or a^{-1} may be the key encrypting some $e \ll \mathsf{msg}(n)$ for $n \in \mathsf{nodes}(\mathbb{A})$. If $a \in \mathsf{unique}_{\mathbb{A}}$, then a must originate at most once in $\mathsf{nodes}(\mathbb{A})$.

$\mathbb{A}$ is *realized* if it is a possible run without additional activity of *regular* participants; i.e., for every reception node n, the adversary can construct $\mathsf{msg}(n)$ via the Dolev-Yao adversary actions,[1] using as inputs:

[1] The Dolev-Yao adversary actions are: concatenating messages and separating the pieces of a concatenation; encrypting a given plaintext using a given key; and decrypting a given ciphertext using the matching decryption key.

1. all messages $\mathsf{msg}(m)$ where $m \prec_{\mathbb{A}} n$ and m is a transmission node;
2. any atomic values a such that $a \notin (\mathsf{non}_{\mathbb{A}} \cup \mathsf{unique}_{\mathbb{A}})$, or such that $a \in \mathsf{unique}_{\mathbb{A}}$ but a originates nowhere in $\mathbb{A}$.

Two skeletons $\mathbb{A}_0, \mathbb{A}_1$ are shown in Fig. 9. They are skeletons of the protocol shown in Fig. 7, i.e. the second step in the derivation of the Question-Answering Cryptoprotocol. Of these skeletons, only $\mathbb{A}_1$ is realized. It is a realized skeleton



$$t_1 = \{|R \hat{} A|\}_B \qquad t_1' = \{|R' \hat{} A|\}_B$$
$$t_2 = \{|R \hat{} Q \hat{} Y \hat{} B|\}_A \qquad t_2' = \{|R' \hat{} Q' \hat{} Y' \hat{} B|\}_A$$

$$\mathsf{unique}_{\mathbb{A}_0} = \{Y'\} \qquad \mathsf{unique}_{\mathbb{A}_1} = \{Y\}$$
$$\mathsf{non}_{\mathbb{A}_0} = \{K_A^{-1}\} \qquad \mathsf{non}_{\mathbb{A}_1} = \{K_A^{-1}, K_B^{-1}\}$$

Fig. 9. Homomorphism $H_0 = [\psi_0, \alpha_0] \colon \mathbb{A}_0 \mapsto \mathbb{A}_1$

because every message to be received may be derived by the adversary from messages previously sent. In fact, every message received was itself previously sent: i.e. the trivial adversary derivation suffices. However, $\mathbb{A}_0$ is not realized. Y' is not derivable from its predecessors in $\mathbb{A}_0$, since the adversary can not use K_A^{-1}, nor re-originate Y'.

The initiator strand in $\mathbb{A}_0$ is only of height 2, rather than 3, so that it has not yet committed to branch to the affirmative or negative answer. Thus, it is a common instance of both initiator strands of its protocol. Whether we regard the not-yet-occurred third node as an affirmative one or a negative node makes no difference. Indeed, the map on skeletons that replaces the first two nodes of an affirmative strand by the first two nodes of a negative strand is an isomorphism.

Given any skeleton $\mathbb{A}_0$, assume that $\psi \colon \mathsf{nodes}_{\mathbb{A}_0} \to \mathsf{nodes}_{\mathbb{A}_1}$. $[\psi, \alpha]$ is an equivalence class of pairs ψ', α'. A pair ψ', α' is in $[\psi, \alpha]$ if (1) $\psi' = \psi$; and (2) $\alpha(a) = \alpha'(a)$, for every a such that $a \ll \mathsf{msg}(n)$ for any $n \in \mathsf{nodes}(\mathbb{A}_0)$. This definition for $[\psi, \alpha]$ implies that the action of α on atoms not mentioned in $\mathbb{A}_0$ is irrelevant.

Definition 2. *Let α be a homomorphism on messages of $\mathfrak{A}$, and $\psi \colon \mathsf{nodes}_{\mathbb{A}_0} \to \mathsf{nodes}_{\mathbb{A}_1}$, for skeletons $\mathbb{A}_0, \mathbb{A}_1$. $H = [\psi, \alpha]$ is a* skeleton homomorphism*, written $H \colon \mathbb{A}_0 \mapsto \mathbb{A}_1$, if:*

1*a*. *For all* $n \in \mathbb{A}_0$, $\mathsf{msg}(\psi(n)) = \alpha(\mathsf{msg}(n))$
1*b*. $\psi(n)$ *is a transmission node iff* n *is a transmission node;*
1*c*. ψ *acts strand-by-strand; i.e.*

$$\forall s\ \exists s'\ \forall j\ .\ s \downarrow j \in \mathsf{nodes}(\mathbb{A}) \quad \textit{implies} \quad \psi(s \downarrow j) = s' \downarrow j;$$

2. $n \preceq_{\mathbb{A}_0} m$ *implies* $\psi(n) \preceq_{\mathbb{A}_1} \psi(m)$*;*
3. $\alpha(\mathsf{non}_{\mathbb{A}_0}) \subseteq \mathsf{non}_{\mathbb{A}_1}$*;*
4*a*. $\alpha(\mathsf{unique}_{\mathbb{A}_0}) \subseteq \mathsf{unique}_{\mathbb{A}_1}$*;*
4*b*. *If* a *originates at* $n \in \mathsf{nodes}_{\mathbb{A}_0}$ *for* $a \in \mathsf{unique}_{\mathbb{A}_0}$*, then* $\alpha(a)$ *originates at* $\psi(n)$*.*

H *is an isomorphism if (as usual) for some* $G\colon \mathbb{A}_1 \mapsto \mathbb{A}_0$, $G \circ H = Id_{\mathbb{A}_0}$.

We sometimes write $H(n)$ for $\psi(n)$ or $H(a)$ for $\alpha(a)$, when $H = [\psi, \alpha]$. As an example of a homomorphism between skeletons, consider the map H_0 between skeletons shown in Fig. 9.

In H_0, $\alpha_0(R') = R$, $\alpha_0(Q') = Q$, and $\alpha_0(Y') = Y$; the remaining parameters are unchanged. The function ψ_0 on nodes maps successive nodes on the left strand of $\mathbb{A}_0$ to their counterparts on the left strand of $\mathbb{A}_1$; it maps nodes on the right strand of $\mathbb{A}_0$ to their counterparts on the right strand of $\mathbb{A}_1$. It also enriches the ordering. Specifically, the topmost nodes are ordered in $\mathbb{A}_1$ but not in $\mathbb{A}_0$; and the second node of the left strand precedes the last node of the right strand in $\mathbb{A}_1$, although they are not ordered in $\mathbb{A}_0$. Moreover, $\alpha_0(\mathsf{unique}_{\mathbb{A}_0}) = \mathsf{unique}_{\mathbb{A}_1}$, although $\alpha_0(\mathsf{non}_{\mathbb{A}_0}) \subsetneq \mathsf{non}_{\mathbb{A}_1}$.

When, for a homomorphism $H = [\psi, \alpha]\colon \mathbb{A} \mapsto \mathbb{A}'$, ψ is an injective function from nodes of $\mathbb{A}$ to nodes of $\mathbb{A}'$, we call H a *node-injective homomorphism*.

We proved in [6] that if $H\colon \mathbb{A} \mapsto \mathbb{A}'$ and $G\colon \mathbb{A}' \mapsto \mathbb{A}$ are node-injective, then $\mathbb{A}$ and $\mathbb{A}'$ are isomorphic. Thus, we write $\mathbb{A} \leq_N \mathbb{A}'$ to mean that for some node-injective H, $H\colon \mathbb{A} \mapsto \mathbb{A}'$. Regarding $\leq_N$ as a relation on skeletons factored by isomorphism, it is a well-founded partial order. In fact, there are only finitely many isomorphism classes below the isomorphism class of any skeleton $\mathbb{A}$.

4 Transformations

We define our notion of transformation via five clauses. Clauses 1–3 are simple bookkeeping requirements. Clause 4 says that a transformation respects origination and $\sqsubseteq$.

Clause 5 says that a transformation respects the instances of roles in the sense of Def. 1. Essentially, Clause 5 says that the transformed protocol does not commit to one branch any earlier than the source, whenever the source roles can branch.

Definition 3. *Suppose that* Π_1 *and* Π_2 *are protocols, and* F *is a map from the strands* $\rho_1 \in \Pi_1$ *to pairs* $F(\rho_1) = \rho_2, g$*, where* $\rho_2 \in \Pi_2$ *and* $g\colon \mathbb{N} \to \mathbb{N}$ *is a function on natural numbers.*

F *is a* protocol transformation *iff, for all* $\rho_1 \in \Pi_1$*, letting* $F(\rho_1) = \rho_2, g$*:*

1. *g is an increasing function, i.e. $i < j$ implies $g(i) < g(j)$.*
2. *$g(\mathsf{length}(\rho_1)) \leq \mathsf{length}(\rho_2)$.*
3. *$\rho_1 \downarrow j$ is a transmission node [resp. a reception node] iff $\rho_2 \downarrow g(j)$ is a transmission node [resp. a reception node].*
4. *For each role node $n = \rho_1 \downarrow i$, and each parameter a such that $a \ll \mathsf{msg}(n)$:*
   (a) *If a originates on $\rho_2 \downarrow k$, with $k \leq g(j)$, then a originates on some $\rho_1 \downarrow j'$ with $j' \leq j$; and*
   (b) *If $a \sqsubseteq \mathsf{msg}(\rho_2 \downarrow k)$, with $k \leq g(j)$, then $a \sqsubseteq \mathsf{msg}(\rho_1 \downarrow j')$ with $j' \leq j$.*
5. *Suppose for some ρ_1, ρ_1' and α, α', and all $j \leq k$,*

$$\alpha(\rho_1 \downarrow j) = \alpha'(\rho_1' \downarrow j).$$

Let $F(\rho_1') = \rho_2', g'$. For every $i \leq k$, $g(i) = g'(i)$; and for every $j \leq g(k)$,

$$\alpha(\rho_2 \downarrow j) = \alpha'(\rho_2' \downarrow j).$$

Node n is an image of *m iff for some ρ_1, α, j, letting $F(\rho_1) = \rho_2, g$, we have $m = \alpha(\rho_1 \downarrow j)$ and $n = \alpha(\rho_2 \downarrow g(j))$.* □

We use the indefinite article, *an* image of m, because there are many substitutions α' such that $m = \alpha'(\rho_1 \downarrow j)$, i.e. all those that differ only for parameters not appearing in $\rho_1 \downarrow j$. For instance, in Fig. 5, the parameters of $C_2' \downarrow 1$ are A, B, Q, so this node is unaffected by the action of α' on R, Y, N, and unaffected by the choice of the encryption keys $\mathsf{pubk}(A), \mathsf{pubk}(B)$. However, different instances of $P_2 \downarrow 2$ in Fig. 4 result as these parameters vary.

According to this definition, we have numerous transformations among the examples given in Section 2. The two-step choreography service of Fig. 5 may be transformed into any of the other choreographies and protocols of Figs. 1–7. Each j^{th} node of the source is mapped to the $j+1^{\text{th}}$ node of the target, so that the functions $g(j) = j + 1$. The first "ready" message is not in the range of the transformations.

Indeed, an alternate transformation maps the affirmative strands of Fig. 5 to negative strands of the targets; this represents an inversion of the convention about the meanings of the outcomes. Indeed, the definition also allows the "nonsensical" maps that send affirmative initiator strands to negative responder strands, and negative initiator strands to affirmative responder strands. These nonsensical transformations, however, have the property that the image of a well-formed choreography execution is not a realized skeleton of the crypto protocol.

All of the three-step protocols have transformations to all of the others, which in some cases introduce additional parameters and cryptographic structure; in the reverse cases, the transformation "forgets" parameters. If P_1 and P_1 sent their first messages in incompatible forms, then Clause 5 would imply that the map from the choreography in Fig. 1 to the resulting cryptoprotocol would not be a transformation. It would force commitment to one branch too early. This corresponds to the (pointless) service in which A allows B to ask a question to

which the answer is *yes*, and then answers *yes*, or allows B to ask a question to which the answer is *no*, and then answers *no*.

Clause 4 prohibits the map from Fig. 7 to a variant of Fig. 4 in which the public or private key of A is included in one of the messages as part of its plaintext. It is a parameter of the source, while not being an ingredient in the plaintext of the messages, and Clause 4(b) requires that this be preserved under transformation. Likewise, a variant of Fig. 4 in which the responder originates R in message 2 without having received it in message 1 would violate Clause 4(a).

When a transformation introduces parameters, then nodes of the source protocol have many possible images under the transformation. Naturally some choices may be unnecessarily constraining, when they select values for the newly introduced parameters that equal other parameters unnecessarily.

Suppose that X is a set of parameters, e.g. the parameters appearing in $\rho_1 \downarrow j$. Some parameters of $\rho_2 \downarrow g(j)$ may not be in X, for instance R if $\rho_2 = P_2$ and $j = 1$. Among substitutions α' that agree with α on X, some are unnecessarily specific, for instance those that map two parameters $\notin X$ to the same value. Selecting $\alpha'(Y) \neq \alpha'(N)$ would be more general than forcing $\alpha'(Y) = \alpha'(N)$. Likewise, forcing a parameter not in X to agree with one in X discards generality, e.g. forcing $\alpha'(Y) = \alpha'(Q)$.

Let α_0 be a substitution that agrees with α on X, and where $\alpha_0(x) = \alpha_0(x')$ implies $x, x' \in X$ or $x = x'$. Then α_0 is maximally general among substitutions that agree with α on X, in the following sense. If α_1 is any substitution agreeing with α on X, then any $\beta \circ \alpha_1 = \gamma \circ \alpha_0$ for exactly one γ. As a consequence, if α_1 also satisfies the same property as α_0, then α_0 and α_1 differ by a renaming.

Thus, for any set of parameters X, α_0 is *universal for* X if $\alpha_0(x) = \alpha_0(x')$ implies $x, x' \in X$ or $x = x'$. Such an α is however unique only up to isomorphism, and the same will remain true for our functions to lift skeletons $\mathbb{A}$. That is, $F(\mathbb{A})$ is determined only to within isomorphism.

In some circumstances (cf. e.g. [13]), it would be desirable to relax the order preserving requirement of Def. 3, Clause 1. In this case, letting

$$g^+(k) = \max_{1 \leq i \leq k} g(i),$$

we would rewrite Clause 2 as $g^+(\mathsf{length}(\rho_1)) \leq \mathsf{length}(\rho_2)$. Similarly, the occurrences of $g(j)$ in Clause 4(a) and (b) would become $g^+(j)$, and the quantification over all $i \leq g(k)$ in Clause 5 becomes $i \leq g^+(k)$. The main results of Section 5 are not substantially changed.

A *strand of* $\mathbb{A}$ is any s where $n \in \mathsf{nodes}(\mathbb{A})$, for at least s's first node $n = s \downarrow 1$.

Definition 4. *Let F transform Π_1 to Π_2, and let $\mathbb{A}$ be a Π_1-skeleton.*

1. *A Π_2-skeleton $\mathbb{B}$ is an F-*image *of $\mathbb{A}$ iff there is a bijection φ between strands of $\mathbb{A}$ and strands of $\mathbb{B}$ such that:*
   (a) For every strand s of $\mathbb{A}$, letting $s = \alpha(\rho_1)$ and $F(\rho_1) = \rho_2, g$:

$$\varphi(s) = \alpha(\rho_2) \quad \textit{and} \quad \mathsf{height}_{\mathbb{B}}(\varphi(s)) = g(\mathsf{height}_{\mathbb{A}}(s));$$

(b) If $s \downarrow j \preceq_{\mathbb{A}} s' \downarrow k$, *then* $(\varphi(s) \downarrow g(j)) \preceq_{\mathbb{B}} (\varphi(s') \downarrow g'(k))$, *where*

$$\begin{array}{ll} s = \alpha(\rho_1) & F(\rho_1) = \rho_2, g \\ s' = \alpha'(\rho'_1) & F(\rho'_1) = \rho'_2, g'. \end{array}$$

(c) Uniquely originating and non-originating values are preserved:

$$\mathsf{unique}_{\mathbb{A}} \subseteq \mathsf{unique}_{\mathbb{B}} \text{ and } \mathsf{non}_{\mathbb{A}} \subseteq \mathsf{non}_{\mathbb{B}}$$

2. *We write* $F(\mathbb{A}) = \mathbb{B}$ *if and only if* $\mathbb{B} \leq_N \mathbb{B}'$, *for every* F*-image* $\mathbb{B}'$ *of* $\mathbb{A}$.

Lemma 1. *Let* F *transform* Π_1 *to* Π_2, *and let* $\mathbb{A}$ *be a* Π_1*-skeleton. There is—to within isomorphism—a unique* $\mathbb{B}$ *such that* $F(\mathbb{A}) = \mathbb{B}$.

Proof sketch. We may regard the strands that contribute nodes to $\mathbb{A}$ as a family $\{\alpha_i(\rho_1^i)\}_{i \in I}$, where for each a representation has been chosen, as an instance of a particular role ρ_1^i under a particular substitution α_i. In some cases, more than one choice of role ρ_1^i may be compatible with the nodes that actually appear to $\mathbb{A}$, which may be only an initial segment of the whole role. Definition 3, Clause 5 ensures that this choice cannot affect the outcome. Moreover, α_i may freely vary for all parameters that do not appear in nodes of this segment of ρ_1^i. By the discussion above, we may require that the α_i are chosen such that:

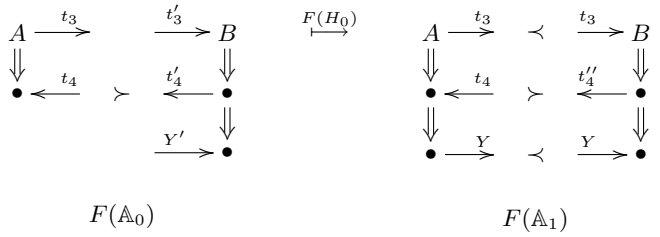
1. for atoms or indeterminates v not appearing in ρ_1^i, $\alpha_i(v) = \alpha_j(w)$ implies $w = v$ and $i = j$;
2. for atoms a not appearing in ρ_1^i, $\alpha_i(a) \notin \mathsf{unique}_{\mathbb{A}} \cup \mathsf{non}_{\mathbb{A}}$;
3. for indeterminates x not appearing in ρ_1^i, $\alpha_i(x)$ is an indeterminate.

We now construct $\mathbb{B}$ by (a) taking images of each strand $\alpha_i(\rho_1^i)$ using $\alpha_i(\rho_2^i)$ up to height $g(k)$, where k is the height of $\alpha_i(\rho_1^i)$ in $\mathbb{A}$ and $F(\rho_1^i) = \rho_2^i, g$; (b) selecting $\preceq_{\mathbb{B}}$ to be a minimal extension of $\varphi(\preceq_{\mathbb{A}})$ compatible with the strand ordering in $\mathbb{B}$; and (c,d) letting $\mathsf{non}_{\mathbb{B}} = \mathsf{non}_{\mathbb{A}}$ and $\mathsf{unique}_{\mathbb{B}} = \mathsf{unique}_{\mathbb{A}}$.

$\mathbb{B}$ is a Π_2 skeleton: The conditions on the nodes and ordering are immediate from the definitions. Moreover, Definition 3, Clause 4 ensures that any node in $\mathbb{B}$ in which $a \in \mathsf{non}_{\mathbb{B}}$ appears as an ingredient is an image of a node in $\mathbb{A}$ in which it already appeared as an ingredient. Thus, the condition on non-origination is satisfied in $\mathbb{B}$ because it was satisfied in $\mathbb{A}$. Similarly, Definition 3, Clause 4 ensures that any node in $\mathbb{B}$ in which $a \in \mathsf{unique}_{\mathbb{B}}$ originates is an image of a node in $\mathbb{A}$ in which it already originated. Thus, the condition on unique origination is satisfied in $\mathbb{B}$ because it was satisfied in $\mathbb{A}$.

Moreover, by the choice of the substitutions α_i, $\mathbb{B}$ has a nodewise injective homomorphism to any other image of $\mathbb{A}$. □

As an example, in Fig. 10, we provide the images of the skeletons $\mathbb{A}_0, \mathbb{A}_1$ from Fig. 9. The new parameters N, N' have been chosen to be different from each other, and different from any other value appearing in the skeletons. Despite the fact that the other primed parameters disappear from $\mathbb{A}_1$, N' remains; generality would have been lost if it were forced to agree with N. Hence, $F(\mathbb{A}_1)$ is not realized, although the result of the substitution $N' \mapsto N$ would yield a realized skeleton.



$F(\mathbb{A}_0)$ $\qquad$ $F(\mathbb{A}_1)$

$t_3 = \{\!|R \hat{} A|\!\}_B$
$t_4 = \{\!|R \hat{} Q \hat{} Y \hat{} N \hat{} B|\!\}_A$

$\mathsf{unique}_{F(\mathbb{A}_0)} = \{Y'\}$
$\mathsf{non}_{F(\mathbb{A}_0)} = \{K_A^{-1}\}$

$t_3' = \{\!|R' \hat{} A|\!\}_B$
$t_4' = \{\!|R' \hat{} Q' \hat{} Y' \hat{} N' \hat{} B|\!\}_A$
$t_4'' = \{\!|R \hat{} Q \hat{} Y \hat{} N' \hat{} B|\!\}_A$

$\mathsf{unique}_{F(\mathbb{A}_1)} = \{Y\}$
$\mathsf{non}_{F(\mathbb{A}_1)} = \{K_A^{-1}, K_B^{-1}\}$

Fig. 10. Lifted Homomorphism $F(H_0)\colon F(\mathbb{A}_0) \mapsto F(\mathbb{A}_1)$

5 Transformations and Homomorphisms

There are three main facts that show that our notion of transformation is reasonable, and these concern the way that transformations relate the homomorphisms among Π_1-skeletons with those among Π_2-skeletons. The first shows that transformations lift each Π_1-homomorphism to a Π_2-homomorphism which is unique to within isomorphism. We write this G subsequently as $F(H)$. See Fig. 11. An example appears in Fig. 10.

Theorem 1 *Let F transform Π_1 to Π_2, and suppose $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$ is a homomorphism on Π_1-skeletons. There is a homomorphism G on Π_2-skeletons $G\colon F(\mathbb{A}_0) \mapsto F(\mathbb{A}_1)$ such that (letting φ_i satisfy the conditions of Def. 4 for $\mathbb{A}_i$):*

1. *$\varphi_1(H(n)) = G(\varphi_0(n))$ for every $n \in \mathsf{nodes}(\mathbb{A}_0)$; and*
2. *$G(v) = H(v)$ for every atom or indeterminate v appearing in $\mathbb{A}_0$.*

Moreover, if G and G' both satisfy this property, then they differ by an isomorphism I, i.e. $G' = I \circ G$. G is node-injective iff H is.

Proof sketch. Suppose that the homomorphism $H = [\psi_1, \beta]$. We may regard the strands contributing nodes to $\mathbb{A}_0$ as a family $\alpha_i(\rho_1^i)$ in such as way that the strands of $\mathbb{A}_1$ in the image of $\mathbb{A}_0$ under ψ are all of the form $(\beta \circ \alpha_i)(\rho_1^i)$. $\mathbb{A}_1$ may restrict which roles are chosen more tightly than $\mathbb{A}_0$, since they may have greater height in $\mathbb{A}_1$. Let β be general for parameters not appearing in $\mathbb{A}_0$.

Thus, we define $G = [\psi_2, \beta]$, where $\psi_2 = \phi_1 \circ \psi_1 \circ \phi_0^{-1}$. G is a homomorphism because each strand $\alpha_i(\rho_2^i)$ in $F(\mathbb{A}_0)$ is mapped by ψ_2 to $\beta \circ \alpha_i(\rho_2^i)$ as desired, and the origination constraints follow as in Lemma 1.

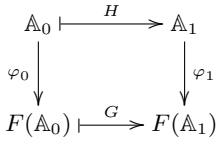


Fig. 11. Homomorphisms H, G

If $G' = [\psi_2', \beta']$ also satisfies the conditions, then let ϕ_1' a function such that, for $n \in \mathsf{nodes}_{\mathbb{A}_1}$, $\phi_1'(n)$ is (1) $\phi_1(n)$ if n is not in the range of ψ_1; and (2) $\psi_2'(\phi_0(m))$ for any m such that $\psi_1(m) = n$, otherwise. By the uniqueness condition on ϕ_1, $\phi_1' \circ (\phi_1)^{-1}$ is the desired isomorphism on $F(\mathbb{A}_1)$.

The node-injectiveness property is immediate from the definition of ψ_2 using the bijections ϕ_1, ϕ_0^{-1}. □

Retaining the notation of Fig. 11, we can also go in the other direction:

Theorem 2 *Let F transform Π_1 to Π_2, and suppose $G\colon F(\mathbb{A}_0) \mapsto F(\mathbb{A}_1)$. $G = F(H)$, for some $H\colon \mathbb{A}_0 \mapsto \mathbb{A}_1$. H is unique to within isomorphism.*

Proof sketch. Let $G = [\psi_2, \beta]$. We define $\psi_1 = \phi_1^{-1} \circ \psi_2 \circ \phi_0$. □

Finally, each Π_2-skeleton $\mathbb{B}$ has a decomposition into a maximal part of the form $F(\mathbb{A})$, to which a node-injective mapping is applied.

Theorem 3 *Let F transform Π_1 to Π_2, and let $\mathbb{B}$ be a Π_2-skeleton. Let*

$$S = \{F(\mathbb{A}_0)\colon \mathbb{A}_0 \text{ is a } \Pi_1\text{-skeleton and } F(\mathbb{A}_0) \leq_N \mathbb{B}\}.$$

Then S has a $\leq_N$-maximum member, i.e. there is a $\mathbb{A}_1$ such that $F(\mathbb{A}_1) \in S$ and for every $\mathbb{B}_0 \in S$, $\mathbb{B}_0 \leq_N F(\mathbb{A}_1)$.

Proof sketch. For each strand s with nodes in $\mathsf{nodes}_{\mathbb{B}}$, consider the nodes in $\mathbb{B}$ that equal the nodes of any $F(\alpha_i(\rho_1^i))$. For each s, choose a ρ_1^i that maximizes the k such that s is of the form $F(\alpha_i(\rho_1^i))$ up to $g(k)$. Let $\mathbb{B}_1$ be the subskeleton of $\mathbb{B}$ where:

1. $\mathsf{nodes}_{\mathbb{B}_1}$ consists of the nodes of $\mathbb{B}$ of height less than this $g(k)$ for each s;
2. $\preceq_{\mathbb{B}_1}$ is the weakest order generated from the strand orderings and $\preceq_{\mathbb{B}}$ restricted to nodes of the form $s \downarrow g(j)$;
3. $\mathsf{non}_{\mathbb{B}_1}$ is the subset of $\mathsf{non}_{\mathbb{B}}$ which are parameters appearing in some $\alpha_i(\rho_1^i)$, and likewise for $\mathsf{unique}_{\mathbb{B}_1}$.

This $\mathbb{B}_1$ is of the form $F(\mathbb{A}_1)$. □

6 Conclusion

In this paper we have simply provided a definition, and shown that the definition is well-behaved. Namely, we show that the definition of transformation fits the skeletons-and-homomorphisms framework of strand spaces. However, this in itself does not tell us when a transformation respects security properties.

For this, we believe that our work on protocol composition via the authentication tests provides the crucial hint [8]. We show there that if adding a new protocol Π_2 to an existing protocol Π_1 produces no new ways to solve the challenge-response patterns on which Π_1's security goals depend, then Π_2 preserves security goals that Π_1 achieves. We also provided a syntactic way to define and validate the "no new solutions" property. We believe that a rewriting of that property to our current framework provides a sufficient criterion for preserving security properties through protocol elaboration.

Such a result would complement the protocol transformation techniques developed by Datta, Derek, Mitchell, and Pavlovic in an outstanding series of papers including [3, 4]. The authors explore a variety of protocols with common ingredients, showing how they form a sort of family tree, related by a number of operations on protocols.

Our definition of multiprotocol from [8] covers both [4]'s *parallel composition* and its *sequential composition*. *Refinement* enriches the message structure of a protocol. Their *transformation* moves information between protocol messages, either to reduce the number of messages or to provide a tighter binding among parameters. Our notion of transformation appears to cover both refinement and their transformation, although the "no new solutions" property appears more likely to work with refinements than with transformations in their sense.

Despite their rich palette of operations, their main results are restricted to parallel and sequential composition [4, Thms. 4.4, 4.8]. Each result applies to particular proofs of particular security goals G_1. Each proof relies on a set Γ of invariant formulas that Π_1 preserves. If a secondary protocol Π_2 respects Γ, then G_1 holds of the parallel composition $\Pi_1 \cup \Pi_2$ (Thm. 4.4). Thm 4.8, on sequential composition, is more elaborate but comparable. By contrast, the key theorem of [8] is one uniform assertion about all security goals, independent of their proofs. It ensures that Π_2 will respect all usable invariants of Π_1. This syntactic property, checked once, suffices permanently, without looking for invariants to re-establish.

References

1. Marco Carbone, Kohei Honda, and Nobuko Yoshida. Structured communication-centred programming for web services. In *Programming Languages and Systems, 16th European Symposium on Programming, ESOP 2007*, pages 2–17, 2007.
2. Véronique Cortier, Jérémie Delaitre, and Stéphanie Delaune. Safely composing security protocols. In V. Arvind and Sanjiva Prasad, editors, *Proceedings of the 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07)*, LNCS, New Delhi, India, December 2007. Springer.

3. Anupam Datta, Ante Derek, John C. Mitchell, and Dusko Pavlovic. Abstraction and refinement in protocol derivation. In *Proceedings, Computer Security Foundations Workshop*. IEEE CS Press, 2004.
4. Anupam Datta, Ante Derek, John C. Mitchell, and Dusko Pavlovic. A derivation system and compositional logic for security protocols. *Journal of Computer Security*, 13(3):423–482, 2005.
5. Shaddin F. Doghmi, Joshua D. Guttman, and F. Javier Thayer. Completeness of the authentication tests. In J. Biskup and J. Lopez, editors, *European Symposium on Research in Computer Security (ESORICS)*, number 4734 in LNCS, pages 106–121. Springer-Verlag, September 2007.
6. Shaddin F. Doghmi, Joshua D. Guttman, and F. Javier Thayer. Searching for shapes in cryptographic protocols. In *Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, number 4424 in LNCS, pages 523–538. Springer, March 2007. Extended version at URL:`http://eprint.iacr.org/2006/435`.
7. Sibylle Fröschle. Adding branching to the strand space model. In *Proceedings of EXPRESS'08*, Electronic Notes in Theoretical Computer Science. Elsevier, 2008. To appear.
8. Joshua D. Guttman. Cryptographic protocol composition via the authentication tests. In Luca de Alfaro, editor, *Foundations of Software Science and Computation Structures (FOSSACS)*, number 5504 in LNCS, pages 303–317. Springer, March 2009.
9. Joshua D. Guttman, Jonathan C. Herzog, John D. Ramsdell, and Brian T. Sniffen. Programming cryptographic protocols. In Rocco De Nicola and Davide Sangiorgi, editors, *Trust in Global Computing*, number 3705 in LNCS, pages 116–145. Springer, 2005.
10. Joshua D. Guttman and F. Javier Thayer. Protocol independence through disjoint encryption. In *Proceedings, 13th Computer Security Foundations Workshop*. IEEE Computer Society Press, July 2000.
11. Joshua D. Guttman and F. Javier Thayer. Authentication tests and the structure of bundles. *Theoretical Computer Science*, 283(2):333–380, June 2002. Conference version appeared in *IEEE Symposium on Security and Privacy*, May 2000.
12. Mei Lin Hui and Gavin Lowe. Fault-preserving simplifying transformations for security protocols. *Journal of Computer Security*, 9(1/2):3–46, 2001.
13. Dimitris Mostrous, Nobuko Yoshida, and Kohei Honda. Global principal typing in partially commutative asynchronous sessions. In *ESOP Proceedings*, LNCS, March 2009.