Automating Strand Spaces Protocol Analysis

Shaddin Doghmi

Jonathan Herzog

Joshua Guttman

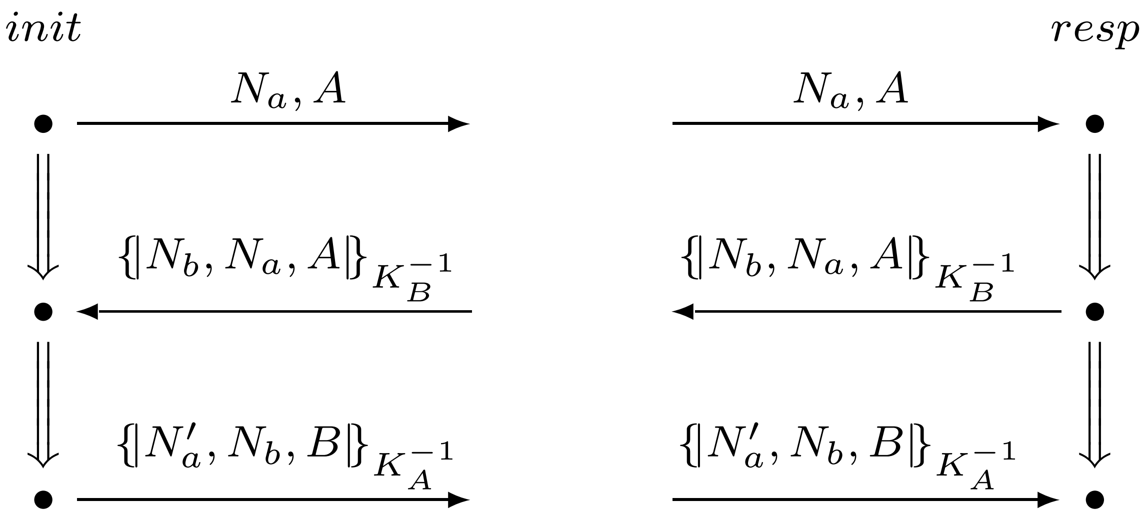
Javier Thayer

MITRE

Background

- A *protocol* Π is a set of parametrised strands, called *roles*
  - For each role r, sets of atoms uniq_r and non_r
    - uniq_r uniquely originating
    - non_r non originating
- A *bundle* $\mathcal{B}$ over Π is a set of regular strands(obeying the protocol) and some penetrator strands connected by message sends/receives
  - $\mathcal{B}$ respects uniq_r and non_r of regular strands
  - Causally well-ordered, inducing partial order $\preceq$ on nodes
  - Models a protocol execution

Protocol Example: ISO Reject

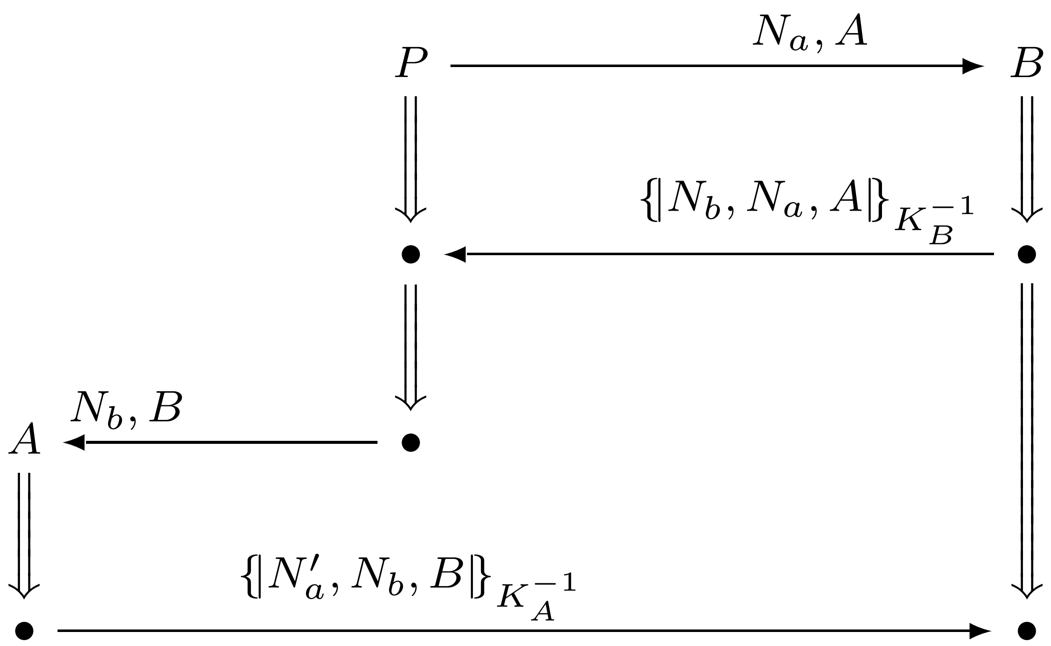


$\mathsf{non}_{init} = \{K_A^{-1}\}$ $\qquad$ $\mathsf{non}_{resp} = \{K_B^{-1}\}$

$\mathsf{uniq}_{init} = \{N_a, N_a'\}$ $\qquad$ $\mathsf{uniq}_{resp} = \{N_b\}$

MITRE

Bundle Example: ISO Reject Attack



MITRE

Motivation

- What does a protocol Π guarantee to participant s
- One way to answer
  — Find all executions containing s
  — Inspect all executions to see if they guarantee security property P
- Problem: infinite number of executions (or 0)
- Question: How do we “characterize” an infinite set of executions in a finite way?

Characterizing Executions: Skeletons

- To finitely characterize infinite set of bundles: list some finite set of shared characteristics
- Need structure to define/formalise characteristics
  — Characteristics of interest : security properties
- Answer: *skeletons*
- A *skeleton* is a set of regular strands, with a partial order on their nodes, and two sets of atoms : set non is non originating in the skeleton and uniq is uniquely originating
- Any bundle characterised by skeleton $\mathbb{A}$ must obey at least the security properties of that skeleton
  — what participants exist
  — what variables participants agree on
  — order of events (nodes)

MITRE

Skeleton Examples

$$
\begin{tikzcd}
{} \arrow[r, "{N_a, A}"] & B \arrow[d, Rightarrow] \\
{} & \bullet \arrow[l, "{\{\!|N_b, N_a, A|\!\}_{K_B^{-1}}}"'] \arrow[d, Rightarrow] \\
{} \arrow[r, "{\{\!|N_a', N_b, B|\!\}_{K_A^{-1}}}"] & \bullet
\end{tikzcd}
$$

$$\mathsf{non} = \{K_A^{-1}, K_B^{-1}\}$$

$$\mathsf{uniq} = \{N_a\}$$

Skeleton Examples (cont.)

$$
\begin{tikzcd}
 & & & & {} \arrow[r, "{N_a, A}"] & B \arrow[d, Rightarrow] \\
A \arrow[d, Rightarrow] & {} \arrow[l, "{N_b, B}"'] & {} \arrow[r, dotted, dash, "\succeq"] & {} & {} & \bullet \arrow[l, "{\{\!|N_b, N_a, A|\!\}_{K_B^{-1}}}"'] \arrow[d, Rightarrow] \\
\bullet \arrow[r, "{\{\!|N'_a, N_b, B|\!\}_{K_A^{-1}}}"] & {} & {} \arrow[r, dotted, dash, "\preceq"] & {} & {} \arrow[r, "{\{\!|N'_a, N_b, B|\!\}_{K_A^{-1}}}"] & \bullet
\end{tikzcd}
$$

$$\mathsf{non} = \{K_A^{-1}, K_B^{-1}\}$$

$$\mathsf{uniq} = \{N_a, N'_a\}$$

Skeletons (cont.)

- We have a structure, a *skeleton*, characterizing some executions
- Want to start with the skeleton $\mathbb{A}$ containing just a single strand s
- $\mathbb{A}$ is a small characterization of all executions containing s
- Somehow get to more informative characterization
  - — extend $\mathbb{A}$ to $\mathbb{A}_2$ by adding information
  - — preserving security properties
- Question: how do we formalize this notion of extension by adding information?

MITRE

Adding Information: Homomorphisms

- Solution: define information preserving map between skeletons
- A *homomorphism* is an information preserving map from some skeleton $\mathbb{A}_1$ to another $\mathbb{A}_2$
- A homomorphism may add information to a skeleton by
  - — adding strands, or extending existing strands
  - — equating variables
  - — equating nodes/strands
  - — extending non or uniq
  - — extending the partial order $\preceq$
- May not "lose" info (remove strands, unequate vars.. etc)
- We say $\mathbb{A}_2$ *enriches* $\mathbb{A}_1$ via H
- $\mathbb{A}_2$ is (weakly) *richer* than $\mathbb{A}_1$ if there is some $H : \mathbb{A}_1 \to \mathbb{A}_2$

MITRE

Homomorphism Example

$$
\begin{tikzcd}
{} \arrow[r, "{N_a, A}"] & B \arrow[d, Rightarrow] \\
{} & \bullet \arrow[l, "{\{\!|N_b, N_a, A|\!\}_{K_B^{-1}}}"']
\end{tikzcd}
$$

Add last node. Equate A and B. Add second strand.

$$
\begin{tikzcd}
 & {} \arrow[r, "{N_a, B}"] & B \arrow[d, Rightarrow] \\
B \arrow[d, Rightarrow] & \succeq \arrow[l, "{N_b, B}"'] & \bullet \arrow[l, "{\{\!|N_b, N_a, B|\!\}_{K_B^{-1}}}"'] \arrow[d, Rightarrow] \\
\bullet \arrow[r, "{\{\!|N_a', N_b, B|\!\}_{K_B^{-1}}}"] & \preceq \arrow[r, "{\{\!|N_a', N_b, B|\!\}_{K_B^{-1}}}"] & \bullet
\end{tikzcd}
$$

Homomorphisms (cont.)

- $\mathbb{A}$ *characterizes* $\mathcal{B}$ if there is $H : \mathbb{A} \to \mathsf{skeleton}(\mathcal{B})$
  - $\mathsf{skeleton}(\mathcal{B})$ regular part of $\mathcal{B}$
    - same $\preceq$, non and uniq
- If $H : \mathbb{A}_1 \to \mathbb{A}_2$, every bundle characterised by $\mathbb{A}_2$ also characterised by $\mathbb{A}_1$
  - Homomorphisms preserve authentication/secrecy properties
- Not necessarily vice-versa
  - Not all information added is necessary to complete $\mathbb{A}_1$ to a bundle
- Are there nontrivial homomorphisms whose target skeleton characterizes the same bundles as the source?
  - If so, we have a richer skeleton characterizing the same executions

MITRE

Authentication Tests as Factor Homomorphisms

- Authentication tests can be phrased as such homomorphisms
- Take $G : \mathbb{A} \to \mathbb{C}$ solving a transformed edge $n_0 \Rightarrow^+ n_1$ in $\mathbb{A}$
  - $\mathbb{C}$ has a transforming edge $m_0 \Rightarrow^+ m_1$ solving $n_0 \Rightarrow^+ n_1$
  - G embeds information of $\mathbb{A}$ into the richer $\mathbb{C}$
- If $m_0 \Rightarrow^+ m_1$ is only possible transforming edge solving $n_0 \Rightarrow^+ n_1$ then $\mathbb{C}$ characterizes same bundles as $\mathbb{A}$
  - Any homomorphism $H : \mathbb{A} \to \mathsf{skeleton}(\mathcal{B})$ factors through G

Authentication Test Example

$$
\begin{tikzcd}
{} \arrow[r, "{N_a, A}"] & B \arrow[d, Rightarrow] \\
{} & \bullet \arrow[l, "{\{\!|N_b, N_a, A|\!\}_{K_B^{-1}}}"'] \arrow[d, Rightarrow] \\
{} \arrow[r, "{\{\!|N'_a, N_b, B|\!\}_{K_A^{-1}}}"] & \bullet
\end{tikzcd}
$$

Solve incoming authentication test involving N_b

$$
\begin{tikzcd}
 & & {} \arrow[r, "{N_a, A}"] & B \arrow[d, Rightarrow] \\
A \arrow[d, Rightarrow] & {} \arrow[l, "{N_b, B}"'] & {} \arrow[l, dotted, "\succeq"'] & \bullet \arrow[l, "{\{\!|N_b, N_a, A|\!\}_{K_B^{-1}}}"'] \arrow[d, Rightarrow] \\
\bullet \arrow[r, "{\{\!|N'_a, N_b, B|\!\}_{K_A^{-1}}}"] & {} \arrow[r, dotted, "\preceq"] & {} \arrow[r, "{\{\!|N'_a, N_b, B|\!\}_{K_A^{-1}}}"] & \bullet
\end{tikzcd}
$$

Not unique solution. Intended run also possible!!

Authentication Test Example (cont.)

$$
\begin{tikzcd}
{} \arrow[r, "{N_a, A}"] & B \arrow[d, Rightarrow] \\
{} & \bullet \arrow[l, "{\{\!|N_b, N_a, A|\!\}_{K_B^{-1}}}"'] \arrow[d, Rightarrow] \\
{} \arrow[r, "{\{\!|N'_a, N_b, B|\!\}_{K_A^{-1}}}"] & \bullet
\end{tikzcd}
$$

Can also be solved to:

$$
\begin{tikzcd}
A \arrow[r, "{N_a, A}"] \arrow[d, Rightarrow] & {} \arrow[r, phantom, "\preceq"] & {} \arrow[r, "{N_a, A}"] & B \arrow[d, Rightarrow] \\
\bullet \arrow[d, Rightarrow] & {} \arrow[l, "{\{\!|N_b, N_a, A|\!\}_{K_B^{-1}}}"'] \arrow[r, phantom, "\succeq"] & {} & \bullet \arrow[l, "{\{\!|N_b, N_a, A|\!\}_{K_B^{-1}}}"'] \arrow[d, Rightarrow] \\
\bullet \arrow[r, "{\{\!|N'_a, N_b, B|\!\}_{K_A^{-1}}}"] & {} \arrow[r, phantom, "\preceq"] & {} \arrow[r, "{\{\!|N'_a, N_b, B|\!\}_{K_A^{-1}}}"] & \bullet
\end{tikzcd}
$$

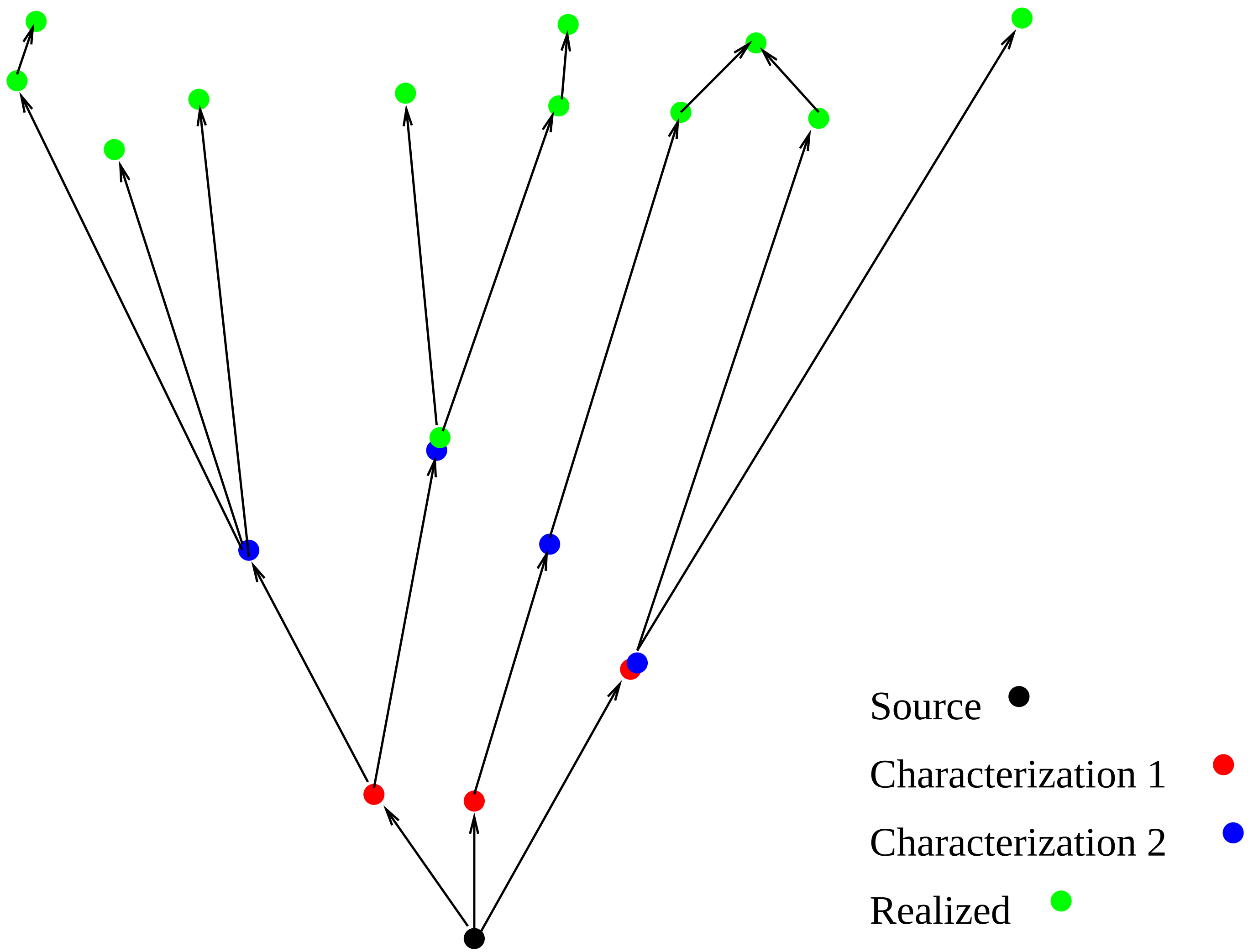
Authentication Tests: Branching

- Sometimes there is more than one possible transforming edge solving a transformed edge $n_0 \Rightarrow^+ n_1$ in $\mathbb{A}$
- We get (finite) set of solution homomorphisms $G_1 ... G_k$ with $G_i : \mathbb{A} \to \mathbb{C}_i$ for some $\mathbb{C}_1 ... \mathbb{C}_k$
  — Any $H : \mathbb{A} \to \mathsf{skeleton}(\mathcal{B})$ factors through one of $G_1 ... G_k$
- Disjunctivly, $\mathbb{C}_1 ... \mathbb{C}_k$ characterize the same bundles as $\mathbb{A}$
- Authentication Tests serve as a way to deduce richer information
- Next: state the problem we want to use authentication tests to solve

Authentication Problem Statement: Characterizations

- Recall: $\mathbb{C}$ *characterizes* all bundles $\mathcal{B}$ where there is a homomorphism from $\mathbb{C} \rightarrow \mathsf{skeleton}(\mathcal{B})$
  — Call that set $\beta(\mathbb{C})$
- Generalize: $X = \{\mathbb{C}_1, \mathbb{C}_2, ...\}$ *characterizes* $\beta(X) = \beta(\mathbb{C}_1) \cup \beta(\mathbb{C}_2)...$
- We start with single strand skeleton $\mathbb{A}$
  — Want to find more informative, finite characterization of $\beta(\mathbb{A})$
- Recall $\mathbb{C}$ is (weakly) richer than $\mathbb{D}$ if there is $H : \mathbb{D} \rightarrow \mathbb{C}$
- A characterization $X = \{\mathbb{C}_1, \mathbb{C}_2, ...\}$ is (weakly) *richer* than $Y = \{\mathbb{D}_1, \mathbb{D}_2 ...\}$ if for any $\mathbb{C}_i$ there exists a $\mathbb{D}_j$ and an $H : \mathbb{D}_j \rightarrow \mathbb{C}_i$
  — Any scenario $\mathbb{C}_i$ is an enrichment of some $\mathbb{D}_j$
  — Preserves security properties

Characterizations Diagram



MITRE

Complete Characterizations

- Recall that we start with a single strand skeleton $\mathbb{A}$ and want to find richer characterization for $\beta(\mathbb{A})$
- How well can we do?
- A *complete characterization* for $\beta(\mathbb{A})$ is one that is richer than any other characterization of $\beta(\mathbb{A})$
  — Always exists. $\{\mathsf{skeleton}(\mathcal{B}) : \mathcal{B} \in \beta(\mathbb{A})\}$ is a (infinite) complete characterization of $\beta(\mathbb{A})$
  — Does a finite complete characterization exist?

MITRE

Complete Characterizations (cont.)

- Definition: $\mathbb{C}$ is *realized* if $\mathbb{C} = \mathsf{skeleton}(\mathcal{B})$ for some $\mathcal{B}$
- A *realized characterization* is a characterization consisting of realized skeletons
- A realized characterization for $\beta(\mathbb{A})$ is complete
- Can read off yes/no answer to any authentication property P from a finite realized characterization $\{\mathbb{C}_1 ... \mathbb{C}_k\}$ of $\beta(\mathbb{A})$
  - — If all $\mathbb{C}_i$ obey P, then all bundles in $\beta(\mathbb{A})$ obey P
  - — If some $\mathbb{C}_j$ contradicts P, then bundle realising $\mathbb{C}_j$ is attack
- Realizedness of a skeleton is decidable
  - — Can recognize a realized characterization
- Characterizations allow us to analyze authentication. What about secrecy?

Secrecy

- For each skeleton $\mathbb{A}$ define a set of atoms $\mathsf{Safe}_{\mathbb{A}}$
  — $a \in \mathsf{Safe}_{\mathbb{A}}$ iff for any bundle $\mathcal{B}$ characterized by $\mathbb{A}$ the penetrator does not learn a
- We have a procedure to compute a reasonable approximation $\mathsf{Safe\_d}_{\mathbb{A}} \subseteq \mathsf{Safe}_{\mathbb{A}}$
- Useful for:
  — Intermediate step in finding authentication tests
  — Discovering future secrecy (e.g. session keys)
- We have tools to prove authentication and secrecy. Time to put those in an algorithm

CPSA Algorithm Overview

- Parameter skeleton $\mathbb{A}$
- Compute $\mathsf{Safe\_d}_{\mathbb{A}}$
- If no unsolved authentication tests, return $\mathbb{A}$ with attached $\mathsf{Safe\_d}_{\mathbb{A}}$
- Else, using $\mathsf{Safe\_d}_{\mathbb{A}}$ as the known set of secrets, find and apply an authentication test (+ related skeleton preserving homomorphism)
  — Authentication test finitely branches, producing a sequence of Homomorphisms $H_1 \ldots H_k$ and images $\mathbb{D}_1 \ldots \mathbb{D}_k$
- Recurse on each of $\mathbb{D}_1 \ldots \mathbb{D}_k$ and output the union of results

MITRE

Interpreting Algorithm Output

- Algorithm called with initial single-strand skeleton $\mathbb{A}$
- Algorithm is *sound*
  - — If terminates, outputs finite *characterization* $X = \{\mathbb{C}_1, ..., \mathbb{C}_n\}$ of $\beta(\mathbb{A})$
  - — $\mathsf{Safe\_d}_{\mathbb{C}_i} \subseteq \mathsf{Safe}_{\mathbb{C}_i}$
- Inspecting output for authentication property P
  - — If all $\mathbb{C}_1 ... \mathbb{C}_k$ obey P, then all bundles containing initial strand obey P
  - — If some $\mathbb{C}_i$ contradicts P, then manual analysis needed (recall realizability decidable)
    - If offending $\mathbb{C}_i$ realizable then $\mathbb{C}_i$ attack
    - else manualy try to enrich $\mathbb{C}_i$ to attack bundle
      - success : attack
      - provable failure: not attack

MITRE

Interpreting Algorithm Output (cont.)

- Inspecting output to see if an atom a occuring in $\mathbb{A}$ is secret
  - If $a \in \mathsf{Safe\_d}_{\mathbb{C}_j}$ for all $\mathbb{C}_j \in X$ then a is secret
  - else, there is some skeleton $\mathbb{C}_i$ with $a \notin \mathsf{Safe}_{\mathbb{C}_i}$
    - If a occurs in the clear and $\mathbb{C}_i$ realizable, then attack
    - else attempt to manually complete $\mathbb{C}_i$ to reveal a
      - Success: attack
      - provable failure: not attack
- Algorithm may not terminate (never encountered in practice)

Future Work and Goals/Predictions

- Improve proof steps of algorithm
  - Extend authentication tests
  - Extend secrecy proof
- Desired result: Algorithm is *complete*
  - If there exists a finite complete characterization, algorithm terminates and outputs one
    - Canonicalised to realizable characterization
  - For each $\mathbb{C}_i$ in output, $\mathsf{Safe\_d}_{\mathbb{C}_i} = \mathsf{Safe}_{\mathbb{C}_i}$
  - Eliminates need for manual analysis. Can read off yes/no answer to any authentication/secrecy property
  - Does not contradict undecidability: may not terminate
- Find a reasonable/natural termination class of protocols
  - If guaranteed finite complete characterization, gives us decidability for that class