Authentication Tests and the Structure of Bundles

Joshua D. Guttman
F. Javier Thayer

September 2000

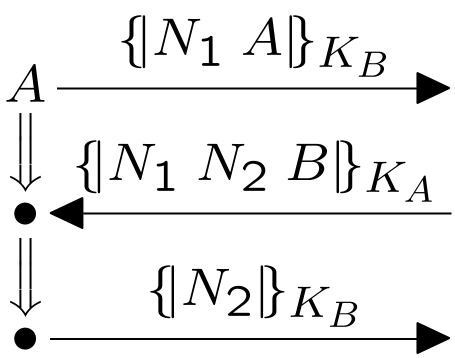
MITRE

Today's Lecture

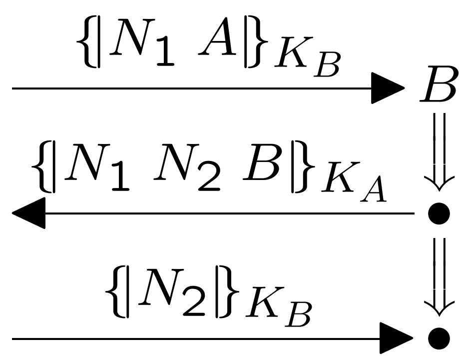
- Authentication Tests:
  - How to find out
    what a protocol achieves
  - How to prove it achieves that
  - Methods to establish
    - Secrecy (especially of keys)
    - Authentication
- Justifying authentication tests
  - Equivalence of bundles
  - Graph operations to simplify bundles
    - Well-behaved bundles
  - Paths through bundles
  - Transforming edges and pedigrees
  - The secrecy theorem
  - Authentication test theorems

MITRE

Needham-Schroeder-Lowe Protocol



$\mathsf{NSLInit}[A, B, N_1, N_2]$



$\mathsf{NSLResp}[A, B, N_1, N_2]$

MITRE

Secrecy of Keys in Needham-Schroeder-Lowe

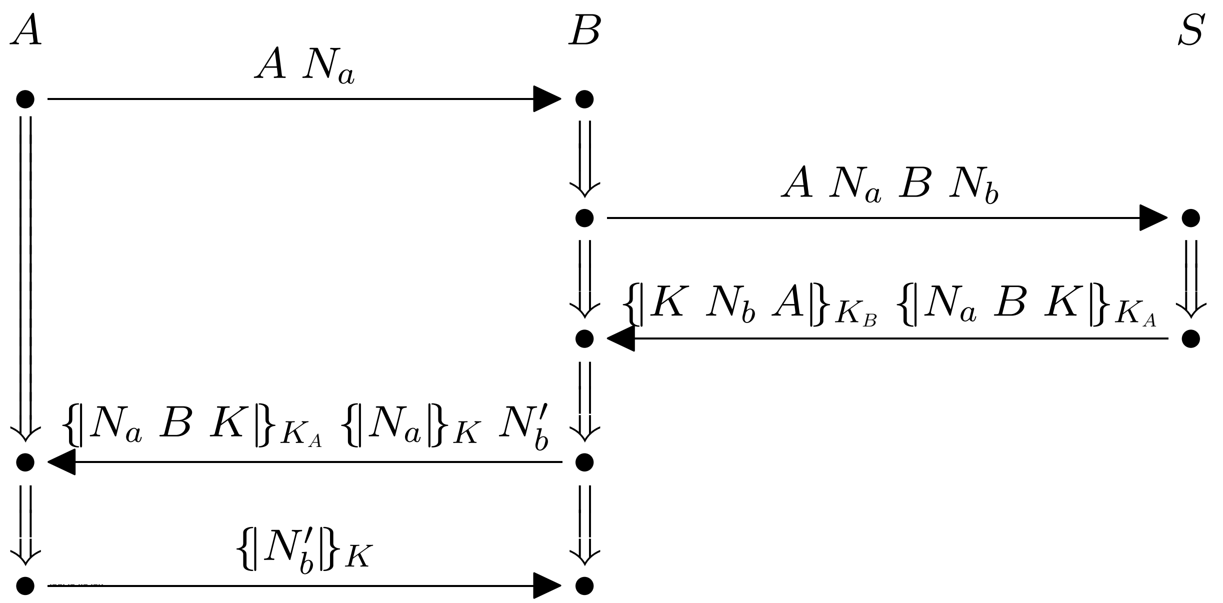
- Some keys $K_{\mathcal{P}}$ are known initially to penetrator
  - Any public key
  - Private keys of malicious principals
  - Private keys carelessly disclosed or stored in compromised devices
- No keys are disclosed in protocol

$$\begin{array}{ll} & K \sqsubset t \\ \text{implies} & t \neq \text{term}(n) \\ & \text{for any regular node } n \end{array}$$

- What the penetrator does not know, he cannot learn
  - Any key not in $K_{\mathcal{P}}$
  - All other keys immediately safe, called S_0

MITRE

Carlsen Protocol



S's behavior: CServ$[A, B, N_a, N_b, K]$
A's behavior: CInit$[A, B, N_a, K, N_b']$
B's behavior: CResp$[A, B, N_a, N_b, K, N_b', H]$
with trace

$$-A\ N_a \quad +A\ N_a\ B\ N_b \quad -\{\!|K\ N_b\ A|\!\}_{K_B}\ H$$

$$+H\ \{\!|N_a|\!\}_K\ N_b' \quad -\{\!|N_b'|\!\}_K$$

Secrecy in Carlsen, I
Safety

- Some keys $K_{\mathcal{P}}$ known initially to penetrator
  - Long-term keys: malice or compromise
  - Old compromised session keys
- Protocol disclosures on regular nodes
  - Long-term keys never disclosed in protocol $K_A \in S_0$ unless $K_A \in K_{\mathcal{P}}$
  - H terms: No new disclosure
    - Retransmission preserves safety
  - Session keys packed via long-term keys
- Keys in new context: protection by safe key gives derivative safety

If K occurs new only in $\{\!|\cdots K \cdots|\!\}_{K_A}$
and $K_A \in S_i$
then $K \in S_{i+1}$

MITRE

Components, "New"

- Term t_0 is a component of t, written $\boxed{t_0} \sqsubset t$

  If $t \in \mathsf{T} \cup \mathsf{K}$, then $\boxed{t} \sqsubset t$

  If $t = \{|h|\}_K$, then $\boxed{t} \sqsubset t$

  $\boxed{t_0} \sqsubset g$ implies $\boxed{t_0} \sqsubset g\ h$

  $\boxed{t_0} \sqsubset h$ implies $\boxed{t_0} \sqsubset g\ h$

  Largest non-concatenated part

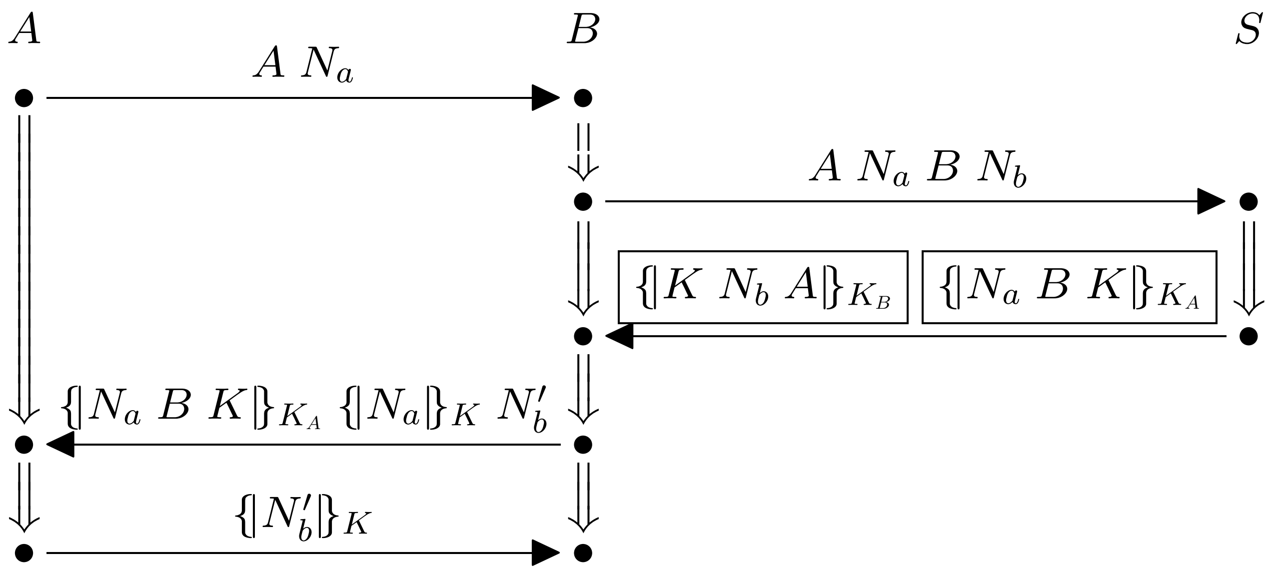
- E.g.: $N_b\ \{|K\ N_b\ A|\}_{K_B}\ \{|N_a\ B\ K|\}_{K_A}$

  $\boxed{N_b}$ $\boxed{\{|K\ N_b\ A|\}_{K_B}}$ $\boxed{\{|N_a\ B\ K|\}_{K_A}}$

- Penetrator controls concatenation fully
- $\boxed{t_0}^{\mathsf{new}} \sqsubset \mathsf{term}(n)$ means
  - $\boxed{t_0} \sqsubset \mathsf{term}(n)$
  - If $n' \Rightarrow^+ n$ then $\boxed{t_0} \not\sqsubset \mathsf{term}(n')$

MITRE

Carlsen: New Components with Keys as Subterms



- Key server (probabilistically) chooses session keys
  - Never used previously
  - Disjoint from long-term keys K_A
  - Not in $K_{\mathcal{P}}$
- CServ$[**, K]$ has at most one strand
- $K_A, K_B \in S_0$ implies $K \in S_1$

MITRE

Key Safety

- K unsafe if $K \notin S_i$ for all i
  - Define $S = \bigcup_i S_i$
- For almost all protocols, either
  - $K \in S_0$, or
  - $K \in S_1$, or
  - $K \notin S$ $\qquad$ K unsafe
- Theorem (proof later):

  If n a node in bundle $\mathcal{C}$
  and $\text{term}(n) = K$
  then $K \notin S$

- "Syntactic" property of protocol entails dynamic property of executions (bundles)
  - S depends on individual regular strands

MITRE

Authentication Tests

NSL: Responder's Guarantee

- Suppose:

  K_A^{-1} safe

  N_2 uniquely originating
- Responder's edge $\{\!|N_1\ N_2\ B|\!\}_{K_A} \Rightarrow \{\!|N_2|\!\}_{K_B}$ is a test
  - Penetrator can't decrypt $\{\!|N_1\ N_2\ B|\!\}_{K_A}$
  - Super-encrypting does no good
  - Penetrator's only choice: discard it or deliver it?
- If responder receives $\{\!|N_2|\!\}_{K_B}$ then test value was delivered
  - But to whom? Which regular strands will receive, change $\{\!|N_1\ N_2\ B|\!\}_{K_A}$?
  - Only regular strand NSLInit$[A, B, N_1, N_2]$, at node 2

MITRE

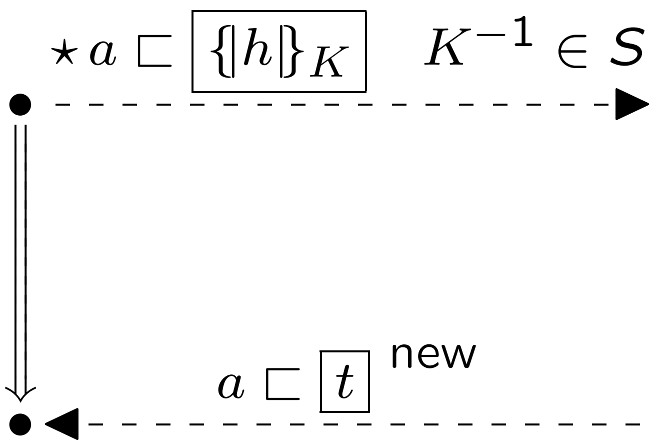
The Anatomy of the Case, I

- Find values originating uniquely on $s_r \in \mathsf{NSLResp}[A, B, N_a, N_b]$
  - N_b only, on node $n_0 = s_r \downarrow 2$
    in component $\{\!|N_a\ N_b\ B|\!\}_{K_A}$
- Find negative (receiving) nodes containing N_b

  $n_1 = s_r \downarrow 3$ with term $\{\!|N_b|\!\}_{K_B}$
- Check:
  - K_A^{-1} is safe
  - $\{\!|N_a\ N_b\ B|\!\}_{K_A}$ not a subterm of a regular node
  - N_b occurs in only one component of n_0
- Therefore, $n_0 \Rightarrow n_1$ is an "outgoing test"

MITRE

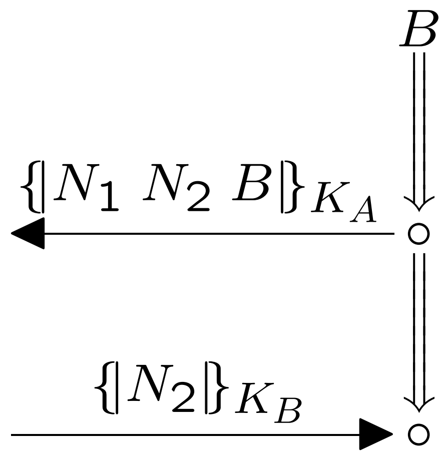
Outgoing Test



a uniquely originates at $\star$

$\boxed{t}$ means a component

NSL Responder Test



MITRE

Transforming Edge

- $n \Rightarrow^+ n'$ is a transforming edge for a if:
  - n negative
  - n' positive
  - $a \sqsubset \boxed{t} \sqsubset \text{term}(n')$
  - t new at n'
- A transforming edge does cryptographic work
  - Creates, transmits new component
- Outgoing test entails *regular* transforming edge

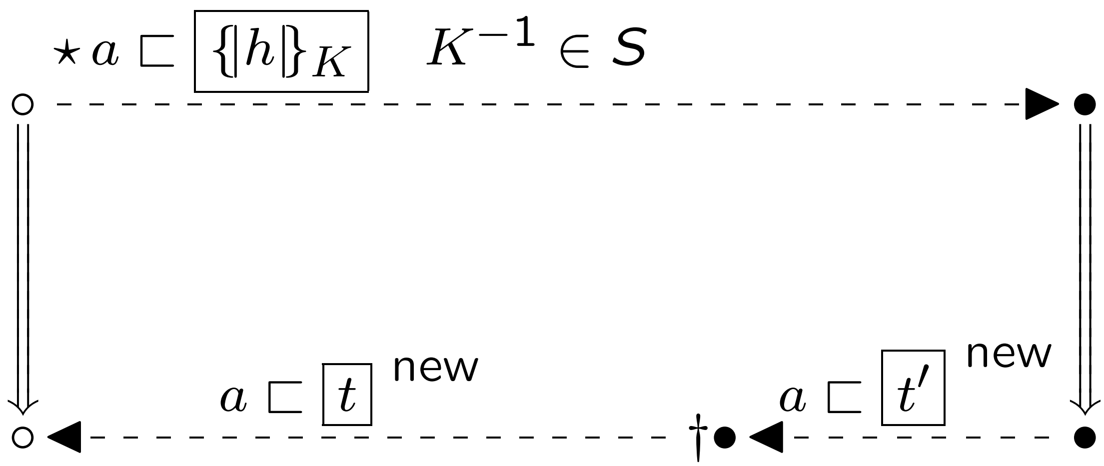
MITRE

The Anatomy of the Case, II

- Since $n_0 \Rightarrow n_1$ is an outgoing test, there's a regular transforming edge $m_0 \Rightarrow m_1$ such that
  - $\boxed{\{\!|N_a\ N_b\ B|\!\}_{K_A}} \sqsubset \mathsf{term}(m_0)$
  - m_0 negative (receiving)
  - m_1 contains N_b in a new component
- Inspecting protocol, $m_0 = s_i \downarrow 2$, where $s_i \in \mathsf{NSLInit}[A, B, N_a, N_b]$, so
  - $m_1 = s_i \downarrow 3$
  - s_i has $\mathcal{C}$-height 3
- This is the NSL responder's guarantee

MITRE

Outgoing test Authentication

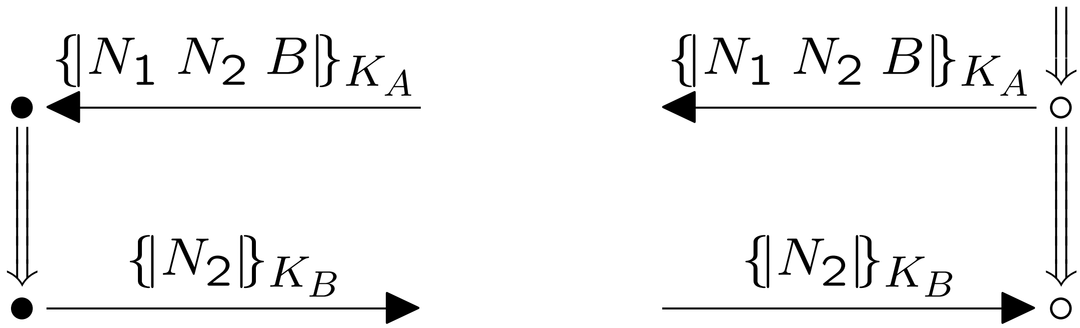


"•" means the test shows this regular node exists

† this node depends on extra conditions

MITRE

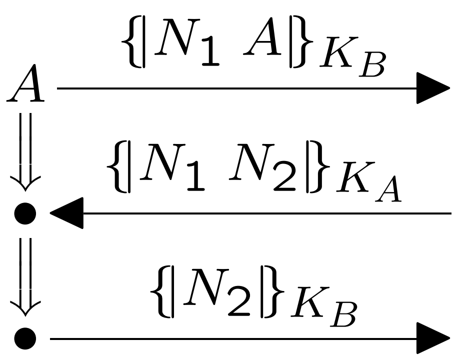
NSL Responder Authentication



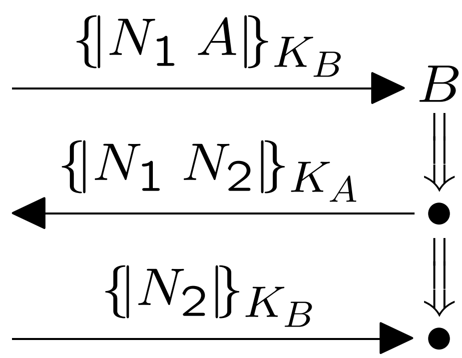
Outgoing test establishes

- nodes present and non-penetrator

Original Needham-Schroeder



NSInit$[A, B, N_1, N_2]$



NSResp$[A, B, N_1, N_2]$

Original NS Responder's Guarantee

- Suppose again:
  K_A^{-1} safe
  N_2 uniquely originating
- Responder's edge $\{\!|N_1\ N_2|\!\}_{K_A} \Rightarrow \{\!|N_2|\!\}_{K_B}$ is a test
  - Penetrator can't decrypt $\{\!|N_1\ N_2|\!\}_{K_A}$
  - Super-encrypting does no good
  - Penetrator's only choice: discard it or deliver it?
- If responder receives $\{\!|N_2|\!\}_{K_B}$, then test value was delivered
  - But to whom?
  - Only regular strand NSInit$[A, *, N_1, N_2]$ can receive $\{\!|N_1\ N_2|\!\}_{K_A}$ and change it
- Whoops: What if $* \neq B$?
  - Unintended service!

MITRE

Anatomy of Original NS

- Part I identifies outgoing test, as in NSL
- Since $n_0 \Rightarrow n_1$ is an outgoing test, there's a regular $m_0 \Rightarrow m_1$ such that
  - $\boxed{\{\!|N_a\ N_b|\!\}_{K_A}} \sqsubset \mathrm{term}(m_0)$
  - m_0 negative (receiving)
- Inspecting protocol, $m_0 = s_i \downarrow 2$, where $s_i \in \mathsf{NSInit}[A, *, N_a, N_b]$, so
  - $m_1 = s_i \downarrow 3$
  - s_i has $\mathcal{C}$-height 3
- This is the NS responder's guarantee; B unconstrained

MITRE

NSL Initiator's Guarantee, I

- Suppose:

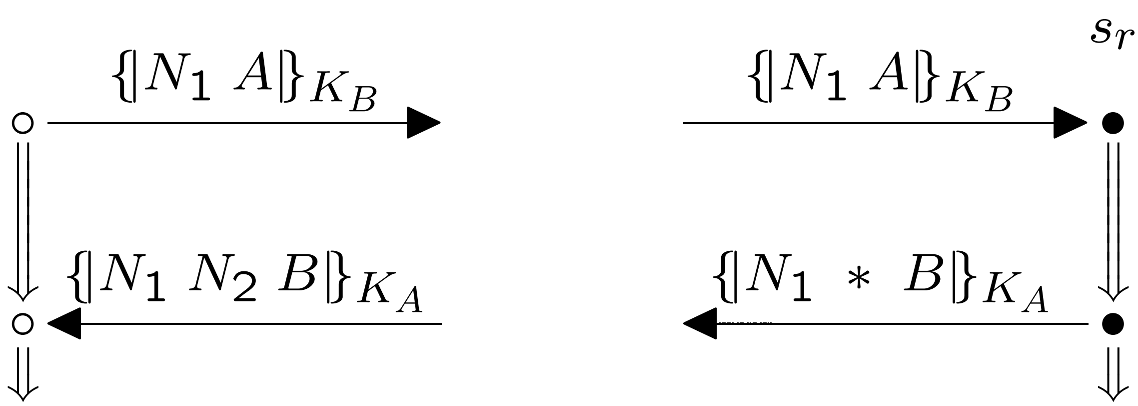
  K_B^{-1} safe

  N_1 uniquely originating

- Initiator's edge $\{|N_1\ A|\}_{K_B} \Rightarrow \{|N_1\ N_2\ B|\}_{K_A}$ is a test
  - Penetrator can't decrypt $\{|N_1\ A|\}_{K_B}$
  - Super-encrypting does no good
  - Penetrator's only choice: discard it or deliver it?
- If initiator receives $\{|N_1\ N_2\ B|\}_{K_A}$ then it was delivered
  - But to whom? Which regular strands will receive, change $\{|N_1\ A|\}_{K_B}$?
  - Only regular strand $s_r \in \mathsf{NSResp}[A, B, N_1, *]$, at node 1

MITRE

NSL Initiator Authentication, I

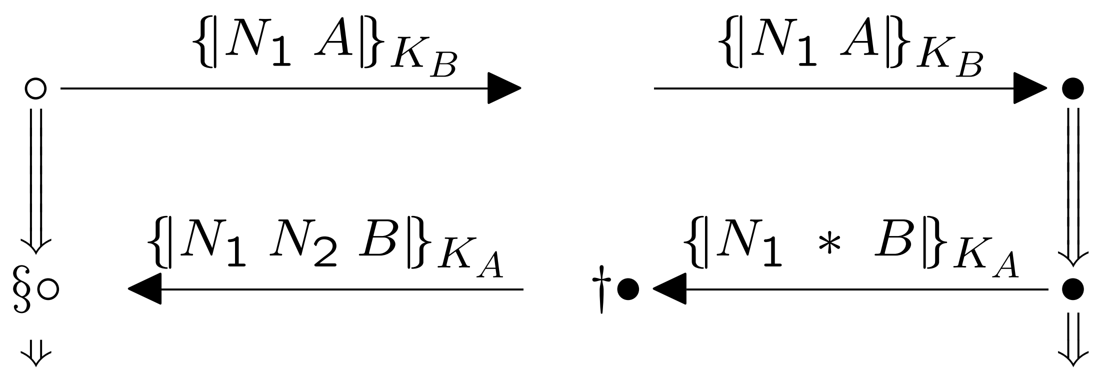


NSL Initiator's Guarantee, II

- Suppose K_A^{-1} also safe
- Penetrator choice:
  discard or deliver $\{|N_1 \ * \ B|\}_{K_A}$
  - Must have delivered it to a regular strand,
    an initiator strand NSInit$[A, B, N_1, *]$
  - But N_1 originates uniquely
    on a strand in NSInit$[A, B, N_1, N_2]$
- So $* = N_2$ and
  $s_r \in$ NSResp$[A, B, N_1, N_2]$
- Uses additional node in
  Outgoing Test Authentication

MITRE

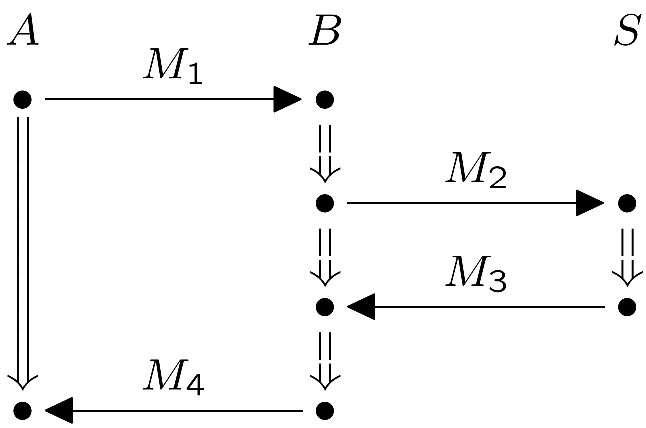
NSL Initiator Authentication, II



$\S\circ = \dagger\bullet$

because of unique origination of N_a

Exercise: Otway-Rees



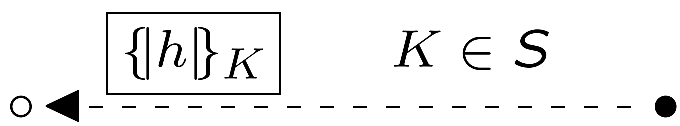
$$M_1 = M\ A\ B\ \{\!|N_a\ M\ A\ B|\!\}_{K_{AS}}$$
$$M_2 = M\ A\ B\ \{\!|N_a\ M\ A\ B|\!\}_{K_{AS}}\ \{\!|N_b\ M\ A\ B|\!\}_{K_{BS}}$$
$$M_3 = M\ \{\!|N_a\ K_{AB}|\!\}_{K_{AS}}\ \{\!|N_b\ K_{AB}|\!\}_{K_{BS}}$$
$$M_4 = M\ \{\!|N_a\ K_{AB}|\!\}_{K_{AS}}$$

What authentication properties
does this protocol achieve?

MITRE

The Server's Guarantee

Unsolicited Test:



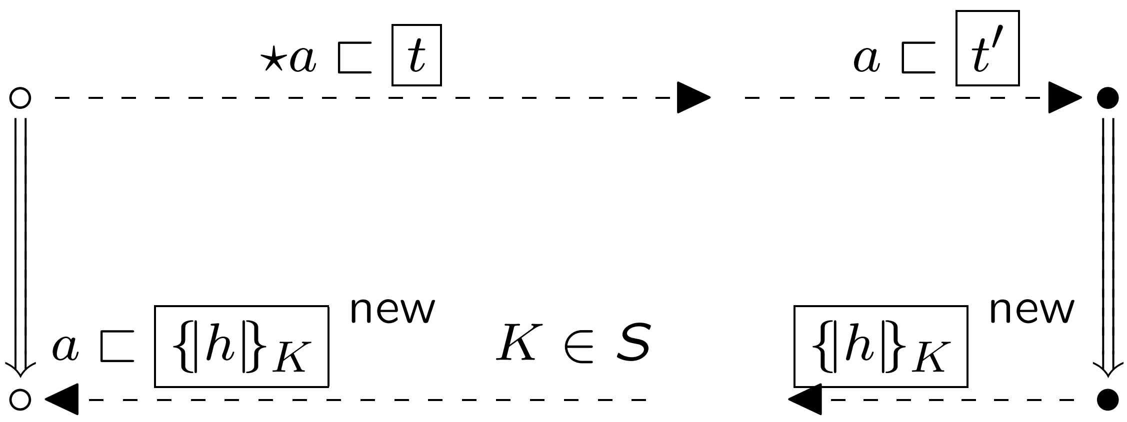
In Otway-Rees

Suppose $s_s \in \mathsf{Serv}[A, B, M, N_a, N_b, *]$ has $\mathcal{C}$-height 1:

- $K_A \notin \mathsf{K}_{\mathcal{P}}$ implies some $s_i \in \mathsf{Init}[A, B, M, N_a, *]$ has $\mathcal{C}$-height 1
- $K_B \notin \mathsf{K}_{\mathcal{P}}$ implies some $s_r \in \mathsf{Resp}[A, B, M, N_b, *]$ has $\mathcal{C}$-height 1

MITRE

Incoming Test Authentication



MITRE

Justifying Authentication Tests

MITRE

Goals for this Hour

- Justify authentication test method
  - Use three ideas
    - ◦ Use equivalence relation on bundles
      Security goals invariant
      under equivalence
    - ◦ Focus on "well-behaved" bundles
      For every bundle, an equivalent
      well-behaved bundle exists
    - ◦ Consider paths through bundles
- Tomorrow: Apply same proof methods
  to protocol mixing

MITRE

Definition: Bundles

A subgraph $\mathcal{C}$ of G_Σ is a *bundle* if $\mathcal{C}$ is finite and causally well-grounded, which means:

1. If $n_2 \in \mathcal{C}$ negative,
   there is a unique $n_1 \rightarrow n_2$ in $\mathcal{C}$
   (everything heard was said)
2. If $s \downarrow i+1 \in \mathcal{C}$, then
   $s \downarrow i \Rightarrow s \downarrow i+1$ in $\mathcal{C}$
   (everyone starts at the beginning)
3. $\mathcal{C}$ is acyclic
   (time never flows backward)

Causal partial ordering $n_1 \preceq_{\mathcal{C}} n_2$ means n_2 reachable from n_1 via arrows in $\mathcal{C}$

Induction: If $S \subset \mathcal{C}$ is a non-empty set of nodes, it contains $\preceq_{\mathcal{C}}$-minimal members

MITRE

Equivalent Bundles

- Bundles $\mathcal{C}, \mathcal{C}'$ are equivalent iff they have the same regular nodes
  - Written $\mathcal{C} \equiv \mathcal{C}'$
  - Penetrator nodes may differ arbitrarily
  - Ordering $\preceq$ may differ arbitrarily
- Authentication goals invariant under equivalence
- Secrecy goals may be expressed in invariant form

  Define to mean: v "uncompromised" in $\mathcal{C}$

  if for all $\mathcal{C}' \equiv \mathcal{C}$ and $n \in \mathcal{C}'$,

  then $v \not\sqsubset_\emptyset \text{term}(n)$
- "Regular nodes" means non-penetrator nodes

  $v \sqsubset_\emptyset t$ concatenating v to other terms yields t

  (v is visible in t, not protected by encryption)

MITRE

Paths and Normality

MITRE

Graph Operations

- A graph operation may:
  - Delete penetrator strands
  - Add edges $n \rightarrow n'$
    with term(n) $= +a$, term(n') $= -a$
  - Delete edges $n \rightarrow n'$
- A graph operation yields graph $\mathcal{C}'$
  - $\mathcal{C}'$ not necessarily a bundle
  - But if it is a bundle, then
    $\mathcal{C}' \equiv \mathcal{C}$

MITRE

Loneliness

- A lonely node in a graph has no edge
  - No incoming edge if negative
  - No outgoing edge if positive
- In definition of bundle:
  - Lonely negative nodes are ruled out:
    You can't hear something if nobody says it
  - Lonely positive nodes are allowed:
    Nobody hears what you say

MITRE

Gregariousness

- A gregarious node in a graph has
  - Several incoming edges if negative
  - Several outgoing edges if positive
- In definition of bundle:
  - Gregarious negative nodes are ruled out: Hear the soloists, not the choir
  - Gregarious positive nodes are allowed: Many people hear your words

MITRE

When are Graph Operations OK?

Suppose $\mathcal{C}'$ is obtained from bundle $\mathcal{C}$ by a graph operation such that

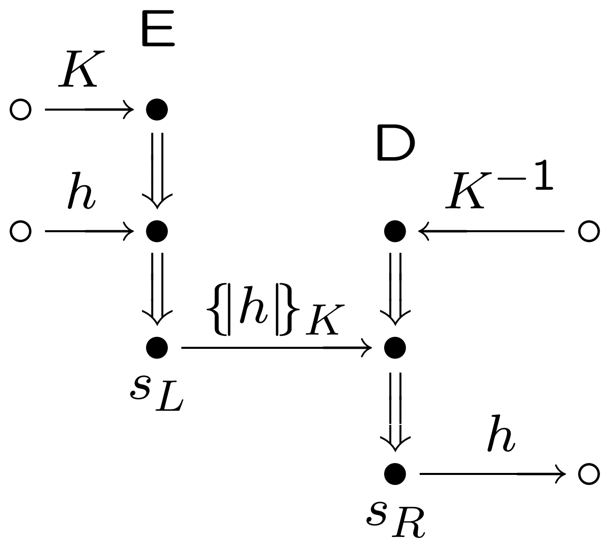
- For any edge new $n \mapsto n'$ of $\mathcal{C}'$, $\quad n \preceq_{\mathcal{C}} n'$
- $\mathcal{C}'$ has no lonely or gregarious negative nodes

Then

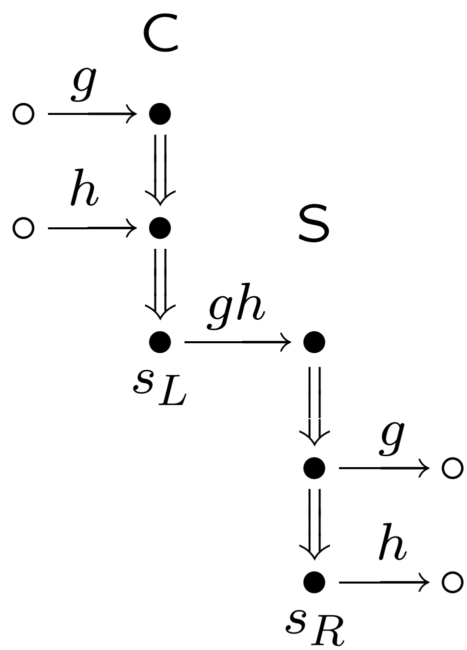
- $\mathcal{C}'$ is a bundle
- $\mathcal{C}' \equiv \mathcal{C}$
- The ordering $\preceq_{\mathcal{C}'}$ on $\mathcal{C}'$ weakens the ordering $\preceq_{\mathcal{C}}$ on $\mathcal{C}$

MITRE

E-D Redundancies



C-S Redundancies



MITRE

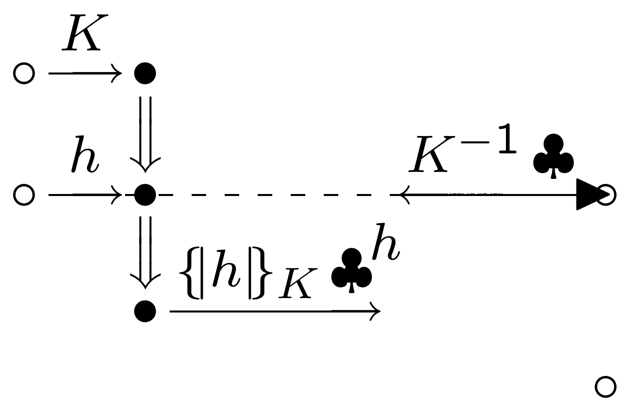
Redundancy Elimination

- Any bundle $\mathcal{C}$ is equivalent to a bundle $\mathcal{C}'$ with no redundancies. Moreover,
  - Penetrator nodes of $\mathcal{C}'$ is a subset of penetrator nodes of $\mathcal{C}$
  - The ordering $\prec_{\mathcal{C}'}$ weakens the ordering $\prec_{\mathcal{C}}$
- Proof: Next two slides
- Consequence: Can assume attacker always

First	Takes things apart
Next	Puts things together
Then	Delivers results

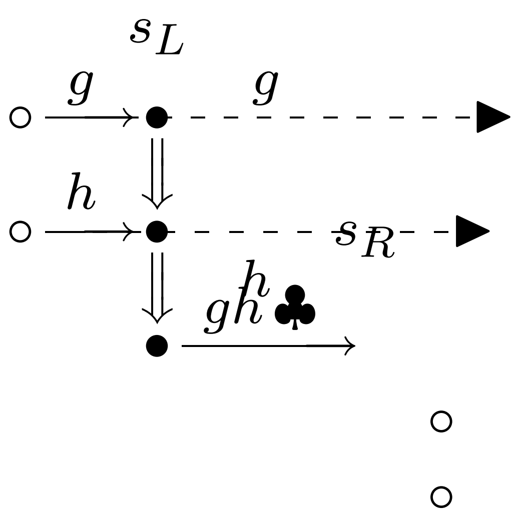
MITRE

E-D Redundancy Elimination



♣ Discarded message

C-S **Redundancy Elimination**



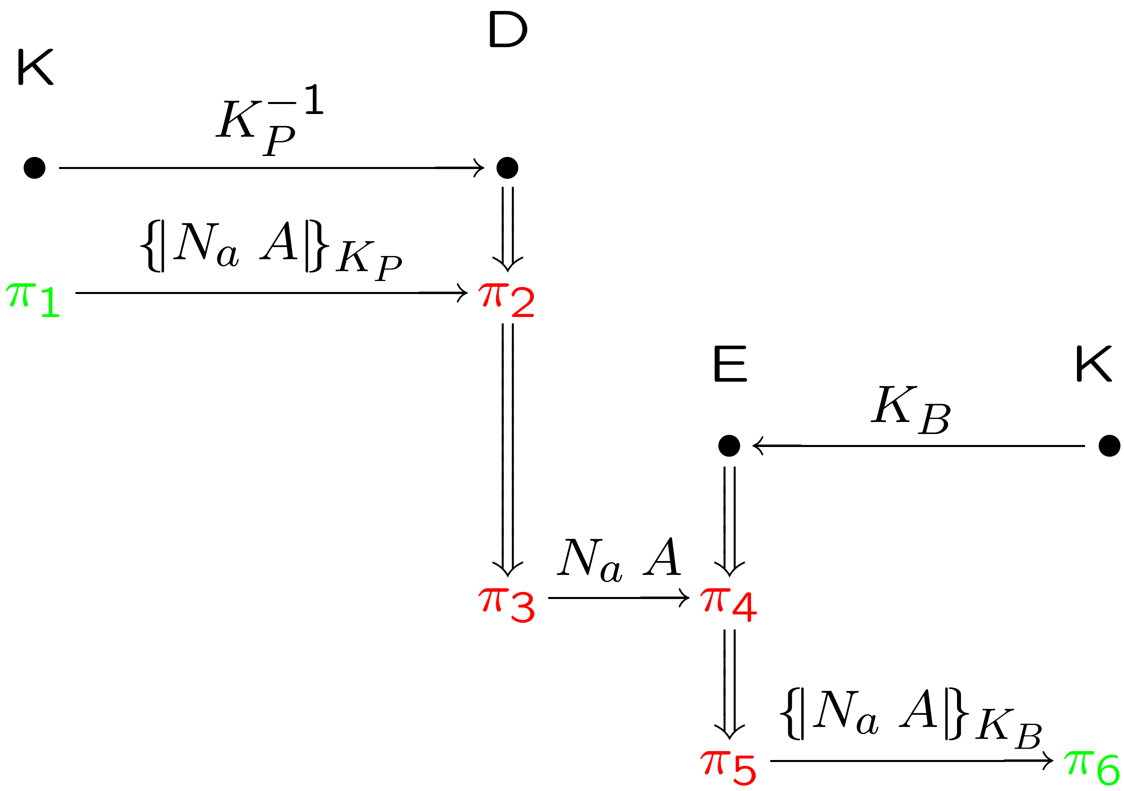
♣ Discarded message

Paths

- $m \Rightarrow^{+} n$ means
  n occurs after m on the same strand
- $m \longmapsto n$ means either 1 or 2:
  1. $m \rightarrow n$
  2. $m \Rightarrow^{+} n$ where
     term(m) negative and
     term(n) positive
- Path p through $\mathcal{C}$: sequence
  $p_1 \longmapsto p_2 \longmapsto \cdots \longmapsto p_k$
  - Typically assume p_1 positive node,
    p_k negative node
  - Notation:
    $|p| = k, \quad \ell(p) = p_k$
- Penetrator path: p_j penetrator node,
  except possibly $j = 1$ or $j = k$

MITRE

A Penetrator Path



MITRE

Construction and Destruction

- A $\Rightarrow^{+}$-edge between penetrator nodes is
  - Constructive if part of a E or C strand
  - Destructive if part of a D or S strand
  - Initial if part of a K or M strand
- Constructive edge followed by a destructive edge Possible forms:
  - Node on $\mathsf{E}_{h,K}$ immediately followed by node on $\mathsf{D}_{h,K}$ (for some h, K)
  - Node on $\mathsf{C}_{g,h}$ immediately followed by node on $\mathsf{S}_{g,h}$ (for some g, h)
- This uses freeness of term algebra

MITRE

Normality

- Bundle $\mathcal{C}$ normal iff

  No penetrator path p has constructive $\Rightarrow$ edge before destructive $\Rightarrow$ edge

- Any bundle is equivalent to a normal one
  - Eliminate redundancies
  - No other constructive/destructive pairs by freeness

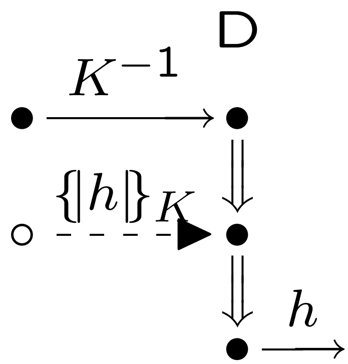
MITRE

Rising and Falling Paths

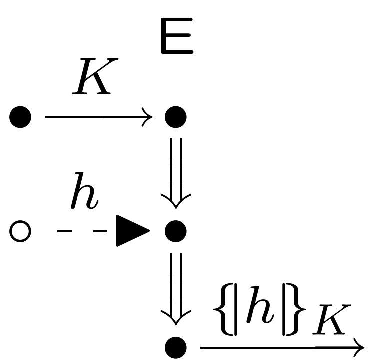
- Definitions: (p a penetrator path)

  Rising term(p_i) $\sqsubset$ term(p_{i+1})

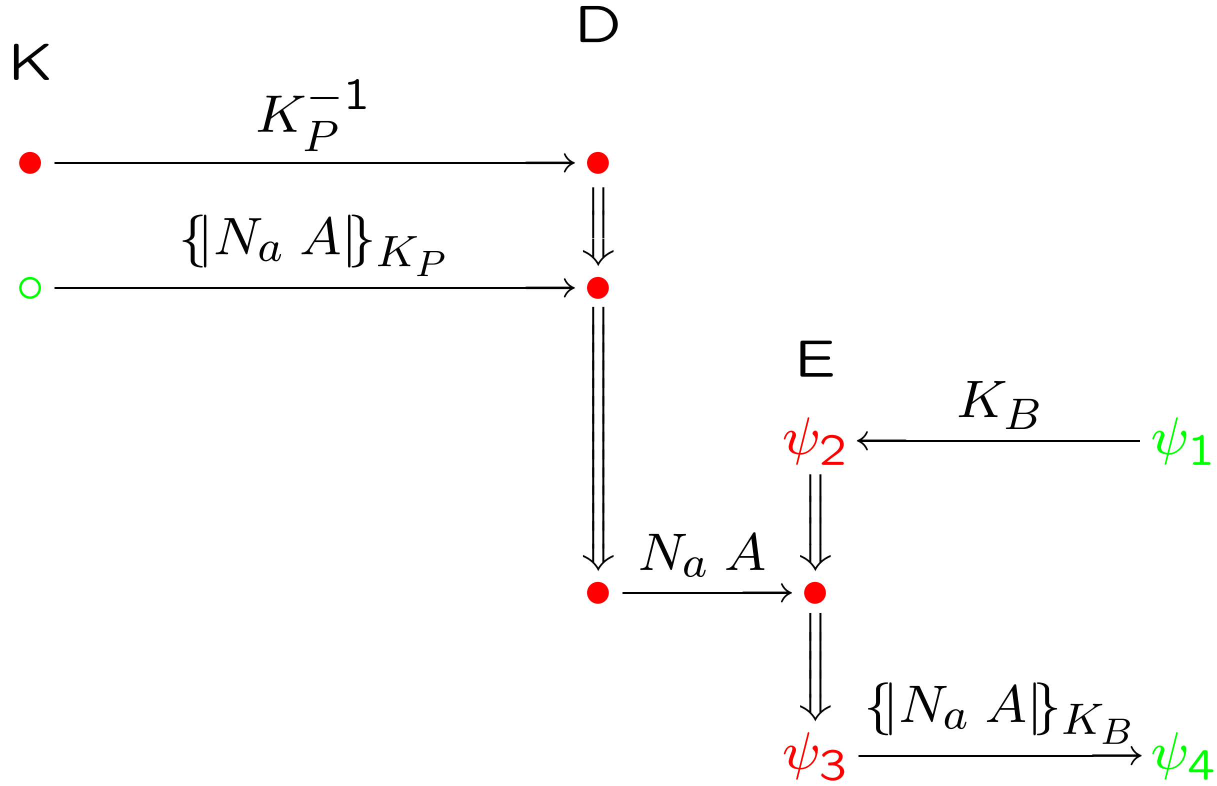
  Falling term(p_{i+1}) $\sqsubset$ term(p_i)
- Destructive paths may not be falling:



Constructive paths may not be rising:



Another Penetrator Path



MITRE

Paths that Avoid Key Edges

- If p is destructive and
  p never traverses D-key edge
  then p is falling

  $$\text{term}(\ell(p)) \sqsubset \text{term}(p_1)$$

- If p is constructive and
  p never traverses E-key edge
  then p is rising

  $$\text{term}(p_1) \sqsubset \text{term}(\ell(p))$$

- If bundle normal and p avoids key edges

  $p = q \rightarrow q'$
  q falling
  q' rising

- $\text{term}(\ell(q)) = \text{term}(q'_1) = \text{pbt}(p)$
  called "path bridge term"

  $\text{pbt}(p) \sqsubset p_1$
  $\text{pbt}(p) \sqsubset \ell(p)$

MITRE

Classifying Penetrator Paths

- Let p penetrator path; traverse backward.
  It may either:

	Reach an initial penetrator node (M, K)
or	Reach a non-initial E- or D-key edge
or	p_1 is regular

- If penetrator path p is useful, then either:

	$\ell(p)$ is regular
or	$\ell(p)$ is a key edge

- All penetrator activity divides into paths p
  where p never traverses key edge

$p_1, \ell(p)$ both regular	
p_1 initial, $\ell(p)$ reg.	* term(p_1) $\sqsubset$ term($\ell(p)$)
p_1 regular	$K =$ term($\ell(p)$)
p_1 a K-node	* $p = \quad p_1 \rightarrow p_2$

* If bundle $\mathcal{C}$ normal

MITRE

Falling Penetrator Paths

- Suppose p_i negative with $1 < i < |p|$
  Then term(p_i) not atomic and

  either term(p_i) $= \{|h|\}_K$ and p_i on D
  or term(p_i) $= g\ h$ and p_i on S

- If p_i positive, term(p_i) = term(p_{i+1})

- Suppose p traverses D with key edge K^{-1}
  only if $K \in \mathfrak{K}$

  Then term($\ell(p)$) $\sqsubset_{\mathfrak{K}}$ term(p_1)

- Definition: $t_0 \sqsubset_{\mathfrak{K}} t$ iff
  t can be built from t_0 using only

  – concatenation (with anything)
  – encryption using $K \in \mathfrak{K}$

  $$\cdots \{| \cdots t_0 \cdots |\}_K \cdots$$

MITRE

Well-Behaved Bundles

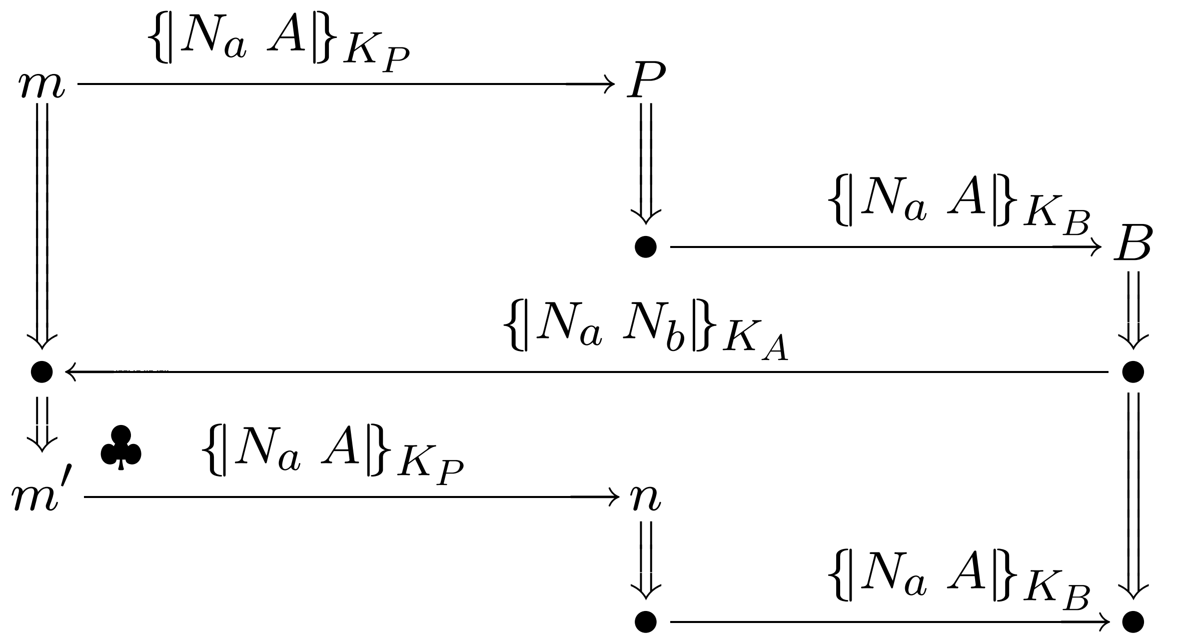
MITRE

Well-Behaved: Definition

- A bundle is well-behaved if
  - Normal
  - Efficient
  - Has simple bridges
- Will define "efficient," "simple bridges"
- Every bundle is equivalent to a well-behaved bundle

MITRE

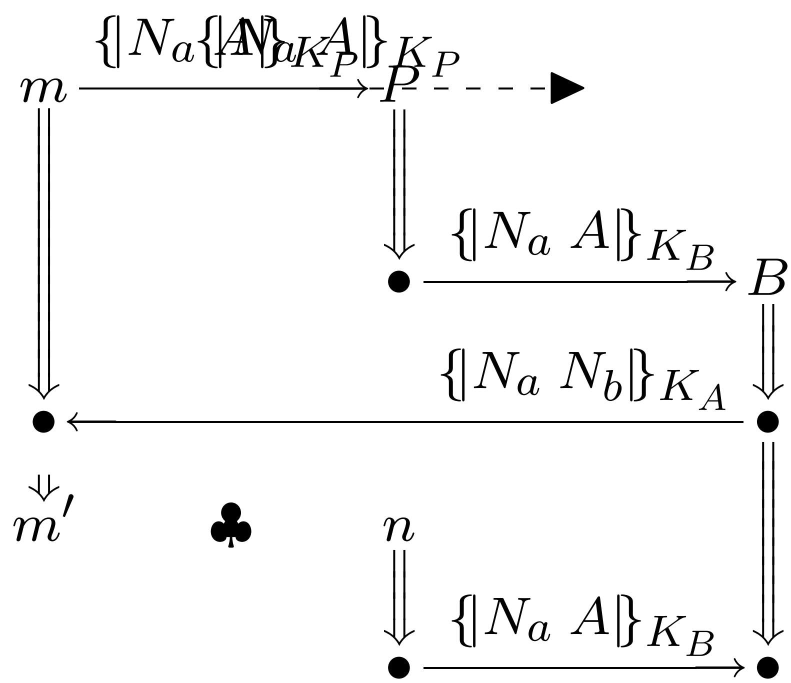
An Inefficient Bundle



- Note: This protocol is fictitious!

MITRE

An Efficient Bundle



MITRE

Efficient Bundles

- In efficient bundle,
  penetrator avoids unnecessary regular nodes
- $\mathcal{C}$ is an efficient bundle iff:
  If m, n are nodes

  n negative penetrator node
  every component of n is a component of m

  Then there are no regular nodes m' such that

  $m \prec m' \prec n$
- For all $\mathcal{C}$, there exists $\mathcal{C}'$ where

  $\mathcal{C} \equiv \mathcal{C}'$
  $\mathcal{C}'$ efficient, normal

MITRE

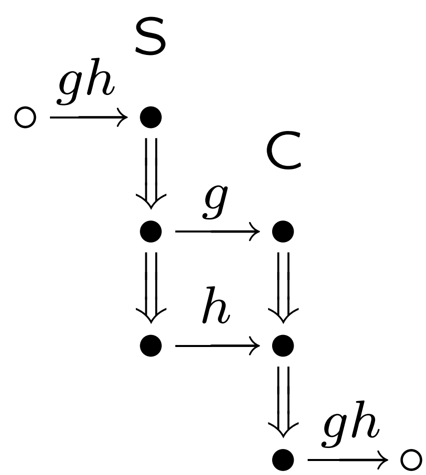
Simple Bridges

- Simple term is either

    An atomic value K, N_a, etc.
    An encryption $\{|h|\}_K$

  Anything but a concatenation
- $\mathcal{C}$ has simple bridges iff

    whenever p a penetrator path
    pbt(p) is simple
- Every $\mathcal{C}$ has an equivalent $\mathcal{C}'$ with simple bridges



MITRE

Transforming Edges, Transformation Paths, Pedigree

Transformed and Transforming Edges

$n_1 \Rightarrow^+ n_2$ is a *transformed edge* for $a \in A$ if

[$n_1 \Rightarrow^+ n_2$ is a *transforming edge* for $a \in A$ if]

1. n_1 is positive [n_1 is negative]
2. n_2 is negative [n_2 is positive]
3. $a \sqsubset \text{term}(n_1)$
4. There is a new component t_2 of n_2 such that $a \sqsubset t_2$

- Penetrator transforming edge: either D or E

MITRE

Transformation Paths

- Path p with p_i labelled by component $\mathcal{L}_i$ of p_i

  where $\mathcal{L}_i = \mathcal{L}_{i+1}$
  unless $p_i \Rightarrow^+ p_{i+1}$ and
  $\mathcal{L}_{i+1}$ is new at p_{i+1}

- $\mathcal{L}_i$ is the "component of interest" at node p_i
- Example:

$$\langle \quad (\pi_1, \{|N_a\ A|\}_{K_P}),\ (\pi_2, \{|N_a\ A|\}_{K_P}), \\ (\pi_3, N_a),\ (\pi_4, N_a),\ (\pi_5, \{|N_a\ A|\}_{K_B}), \\ (\pi_6, \{|N_a\ A|\}_{K_B}) \quad \rangle$$

MITRE

Separated Transformation Paths

Theorem:

- Suppose $\mathcal{C}$ well-behaved and

  $(p, \mathcal{L}), \quad (p', \mathcal{L}')$ transformation paths

  $\ell(p) \prec m \prec p'_1 \qquad m$ regular

  p_1, $\ell(p')$ simple

  (not concatenated)

- Then $\mathcal{L}_i \neq \mathcal{L}'_j$
  for all i, j where $1 \leq i \leq |p|$, $1 \leq j \leq |p'|$
- That means:

  Never repeat component of interest
  when penetrator paths
  separated by regular nodes

MITRE

Pedigree Path for a

- Transformation path $p, \mathcal{L}$ where

  a originates at p_1
  $a \sqsubset \mathcal{L}_i$ for all i
  p does not traverse any D, E key edges

- Whenever $a \sqsubset \text{term}(n)$,
  exists pedigree path with $\ell(p) = n$

  – Proof idea:
    Keep tracing backward,
    selecting components containing a

- Authentication test proof idea:
  Check which steps on pedigree path
  must occur on *regular* edges

MITRE

Key Safety, Authentication Tests Justified

MITRE

Penetrable Keys, Safe Keys

$P_0 = K_{\mathcal{P}}$

$K \in P_{i+1}$ iff

exists n regular, positive, with

$K \sqsubset_{\mathfrak{K}} \boxed{t}^{\text{new}} \sqsubset \text{term}(n)$, and

for all $K_0 \in \mathfrak{K}$, $K_0^{-1} \in P_i$

$P = \bigcup_i P_i$

- Let $\mathcal{C}$ be a bundle

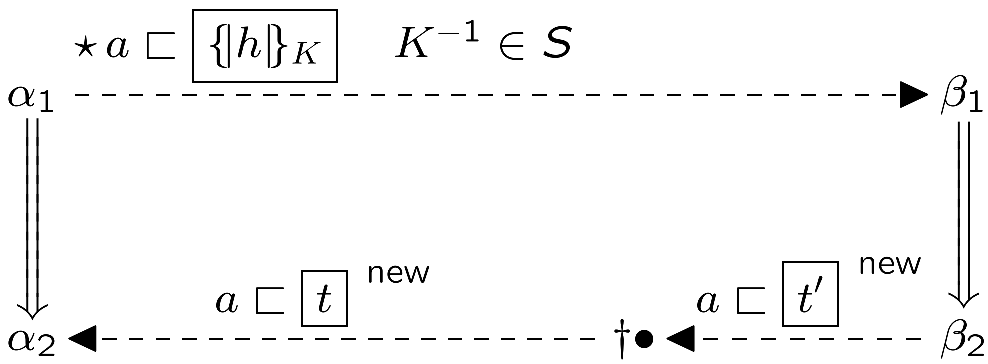
  If $n \in \mathcal{C}$
  and $\text{term}(n) = K$
  then $K \in P$

- Proof idea:
  - Use "Classifying Penetrator Paths"
  - Use "Falling Penetrator Paths"

- $S \cap P = \emptyset$

MITRE

Outgoing test Authentication



The test shows the β regular nodes exist. Proof:

- Exists pedigree path $p, \mathcal{L}$ where

  $$p_1 = \alpha_1 \qquad (a \text{ uniquely originates})$$
  $$\{|h|\}_K = \mathcal{L}_1 \neq \ell(\mathcal{L}) = t$$

- Must be first transforming edge $n_1 \Rightarrow^+ n_2$
  - Penetrator D impossible
  - Penetrator E contradicts normalcy
  - Otherwise regular strand, QED

MITRE

Summary of this Lecture

- Authentication tests:
  Methods to establish
  - Secrecy (especially of keys)
  - Authentication
- Justifying the authentication tests
  - Bundles, equivalence
    - Normal and well-behaved bundles
  - Paths through bundles
    - Transformation paths
    - Pedigree paths
- Tomorrow:
  - Protocol independence
    through disjoint encryption
  - Authentication tests as a design strategy

MITRE