An Encapsulated Authentication Logic for Reasoning about Key Distribution Protocols

Catherine Meadows  
NRL

Dusko Pavlovic  
Kestrel Institute

Iliano Cervesato  
Tulane University

ONR Review Meeting  
August 23, 2005
Contributions

- Separate
  - Authentication reasoning
  - Secrecy reasoning
- Define a logic of pure authentication
  - Secrecy as assumptions
    - Proof obligations
- Embed it in derivational framework
- Apply to key distribution protocols
  - Taxonomy
  - Comparative study
  - Clear understanding of underlying mechanisms
Server-Assisted Shared Key Distribution Protocols

I. Cervesato: Encapsulated Authentication Logic
Key Distribution Protocols

Secrecy
- k secret only if sent over authenticated channels

Authentication
- Authentication depends on secrecy
  - Cryptographic authentication relies on secrecy of long-term keys

- Secrecy depends on authentication
  - k secret only if sent over authenticated channels
Verifying KD Protocols

Historically single monolithic proofs

... BUT ...

secrecy and authentication rely on very different proof methods

- **Authentication**
  - Completing partial order of actions
    - Get piping right
  - Local reasoning
  - Positive inference

- **Secrecy**
  - Secret goes only to intended recipients
    - Pipes do not leak
  - Global reasoning
  - Negative inference
Divide et Conquera

- Two coordinated logics
  - Logic of authentication
    - Relies on secrecy assumptions
      - Proof obligation in secrecy logic
  - Logic of secrecy
    - Relies on authentication assumptions
      - Proof obligation in auth. logic

- Benefits
  - Much simpler proofs
  - Modularity
    - Independent of notion of secrecy
Describing Protocol Runs

• **Messages**
  - $km$ - encryption
  - $m,m'$ - pairing

• **Principal actions**
  - $\langle m: A \rightarrow B \rangle_A$ - send
  - $\langle X: Y \rightarrow Z \rangle_A$ - receive
  - $(m/p(x))_A$ - match
  - $(\nu n)_A$, $(\tau t)_A$ - new nonce, timestamp

<table>
<thead>
<tr>
<th>Abbrev.</th>
<th>Description</th>
</tr>
</thead>
</table>
| $\langle m \rangle_A$ | $m$ sends
| $((m))_A$ | $m$ receives
| $\langle m \rangle_A$ | $m$ matches

• **Runs**
  - Partial order of actions
    - Every receive has a send
    - Every match has succeeded
  - Observations

• **Protocols**
  - Set of parametric roles
    - Akin to observations
Authentication Logic

• First-Order logic with 3 predicates
  - \( a_A \) - action \( a_A \) has occurred
  - \( a_A < b_B \) - \( a_A \) has occurred before \( b_B \)
  - \( a_A = b_B \) - \( a_A \) and \( b_B \) are the same action
  Nothing else!

• Usage
  - Given A's observations, extend them with other principal's actions
    - Derive compatible runs
      - \( A: \text{Obs}_A \Rightarrow \Phi \)
      - \( A: \Psi \land \text{Obs}_A \Rightarrow \Phi \)
  - Iterated application of axioms
Logical Assumptions

- **Honesty**
  - Principal does not deviate from role

- **Secrecy**
  - Key uncompromised for given principals

**honest S**

\[
\text{secret}(k, G) = \langle\langle k \ m \rangle\rangle_x \Rightarrow X \in G \\
\& \ (x/k \ y)_x \Rightarrow X \in G
\]

\[
k m
\]

secret(k, [A,S])
Axioms

- Basic truths about domain
  - **Receive axiom**
    
    \[ Y: ((m)_A \rightarrow \langle\langle m\rangle\rangle_X < ((m)_A) \]

  - **Timestamp axiom**
    
    \[ A: \text{honest } B \land \langle\langle t\rangle\rangle_B < ((t)_A) \]
    
    \[ \rightarrow (t-\delta)_A < (t)_B < \langle\langle t\rangle\rangle_B < ((t)_A) < (t-\Delta)_A \]

- Allow inferring new actions/ordering
Schemas and Instances

• Desired functionalities
  - **Nonce-based Challenge-Response property**
    \[ A: \Phi \& (v \ n)_A ^{\{C \ n\}_A} ^{\{R \ n\}_A} \]
    \[ (v \ n)_A ^{\{C \ n\}_A} ^{\{C \ n\}_B} ^{\{R \ n\}_B} ^{\{R \ n\}_A} \]

• Verified instances
  - **Challenge in the clear/Response encrypted**
    \[ A: \text{secret}(K, [A,B]) \& (v \ n)_A ^{\{\text{K n}\}_A} ^{\{\text{K n}\}_B} ^{\{\text{K n}\}_A} \]
    \[ (v \ n)_A ^{\{\text{K n}\}_B} ^{\{\text{K n}\}_A} \]
Abstract Key Distribution

• S spontaneously
  ➢ Generates k
  ➢ Sends it to A, B
    ▪ A, B hardwired
  ➢ Encrypted with $K_{AS}$, $K_{BS}$

• A observes only ($K_{AS}$ k)

• A reconstructs run
  ➢ Must assume
    ▪ honest S
    ▪ secret($K_{AS}$, [A,S])
    ▪ Not secret($K_{BS}$, [B,S])
  ➢ B’s reception unknown

• Dual for B

\[
A : \text{secret}(K_{AS}, [A,S]) \land \text{honest S} \land (K_{AS} k)_A^\nu < \left( \left\langle \langle K_{AS} k \rangle_{S^X} \rangle_{K_{BS} k} \right\rangle_{S^X} \right) < (K_{AS} k)_A
\]
Derivational Approach

• Use rules, not just axioms
  ▪ Operate on protocol and properties
    • Refinements
    • Transformations

• Advantages
  • Abstract general constructions
  • Reuse protocol fragments
  • Structured understanding of
    ▪ Mechanism
    ▪ Properties
    ▪ Relations between protocols
  • Open-ended taxonomies
Key Request

- A may not be talking to B
  - Even if S honest
- Same for B
Binding

- **A** (B) authenticated to **B** (A)
- A knows S sent $K^{AS}(B,k), K^{BS}(A,k)$
- A received $K^{AS}(B,k), M$
- A doesn’t know if $M = K^{BS}(A,k)$
- Documented anomaly of Kerberos 5
A authenticates B assuming

\[ \text{secret}(K^{BS}, [B, S]) \]
B’s Point of View

- With only
  - $\text{secret}(K^B_S, [B,S])$ knows S generated k

- With also
  - $\text{secret}(K^A_S, [A,S])$
    - knows A knows k
    - A may not be honest
Additional Properties

• Recency
  ➢ \((v \ k)_S\) bracketed by events controlled by A/B
    ➢ Otherwise, intruder can infer k and attack protocol
    ➢ Even if S is honest
  ➢ Not satisfied so far

• Key confirmation
  ➢ A/B knows that B/A has k
    ➢ Essential for using k
  ➢ Only B in KD^4 (under assumption)
Recency with Nonces

- Use challenge-response as bracket

\[ K_{AS}(n, B, k, K_{BS}(A, k)) \]

\[ K_{BS}(A, k) \]
Core NSSK

- Ensures recency of $k$ to $A$
- $A$ can reconstruct run up to $B$'s action
- No such guarantees for $B$
  - Denning-Sacco attack
Core NSSKfix

Nonce-based CR

I.Cervesato: Encapsulated Authentication Logic
Key Confirmation

- Under the assumption
  - $\text{secret}(k, [A, B, S])$

Post-composition
NSSK does more!

- B concludes with CR
  - $k$ not confirmed to $A$
    - Unless tagging
  - $B$ already knows $A$ has $k$

- Exchange typical of repeated authentication
  - $B$ repeatedly request service from $A$
    - ... but $A$ is initiator!

- Similarly for NSSK-fix
Recency with Timestamps

- Timestamp as bracketing device
  - Requires loosely synchronized clocks

\[ K^{AS} (m, t) \]

\[ \text{secret}(K^{AS}, [A, S]) \]
I.Cervesato: Encapsulated Authentication Logic

Denning-Sacco

- Guarantee recency to both A and B
- Same assurance as core NSSK-fix
  - Only 3 messages

\[ K^{AS}(B, k, t, K^{BS}(A, k, t)) \]

Timestamping

\[ K^D_0 \rightarrow K^D_1 \rightarrow K^D_3 \rightarrow K^D_4 \]

\[ K^D_0 \rightarrow K^D_1 \rightarrow K^D_3 \rightarrow K^D_4 \]

\[ A \rightarrow A, B \rightarrow S \rightarrow B \]

\[ K^{BS}(A, k, t) \]

\[ K^{BS}(A, k, t) \]

\[ K^{AS}(B, k, t, K^{BS}(A, k, t)) \]
Core Kerberos 4

- Key confirmation
- Repeated auth.

**Kerberos 4**
- 2 rounds
- Many more fields, options, ...

\[ K^{AS}(B,k,t,K^{BS}(A,k,t)) \]

\[ k \cdot m[t'] \]
Core Kerberos 5

- Kerberos 5
  - 2 rounds
  - Even more fields, options, ...

Key confirmation

Repeated auth.

\[ K_{A}^{S}(B,k,t), K_{B}^{S}(A,k,t) \]
Define Secrecy Logic

- Authentication as assumptions
- Modular model of secrecy
  - Dolev-Yao
  - Information-theoretic
  - Computational
- Apply to examples
  - Diffie-Hellman hierarchy
  - Full Kerberos 5
  - PKINIT
- Implement within Kestrel’s PDA