An Encapsulated Authentication Logic for Reasoning about Key Distribution Protocols

Catherine Meadows
NRL

Dusko Pavlovic
Kestrel Institute

Iliano Cervesato
Tulane University
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

KD^0 \rightarrow KD^1 \rightarrow KD^2 \rightarrow KD^3 \rightarrow KD^4

DS \rightarrow NSSK^0 \rightarrow NSSKfix^0

K5core^0 \rightarrow K5core \rightarrow K4core^0 \rightarrow K4core \rightarrow NSSK^1 \rightarrow NSSK \rightarrow NSSKfix^1 \rightarrow NSSKfix
Key Distribution Protocols

- Secrecy depends on authentication
  - $k$ secret only if sent over authenticated channels

- Authentication depends on secrecy
  - Cryptographic authentication relies on secrecy of long-term keys
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**
  - $k \cdot m$ - encryption
  - $m, m'$ - pairing

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

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

- **Protocols**
  - Set of parametric roles
    - Akin to observations

**Abbreviation**
- $\langle \langle m \rangle \rangle_A$
- $\langle (m) \rangle_A$
- $\langle m \rangle_A$
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
      \( A: \text{Obs}_A \Rightarrow \Phi \)
      \( A: \Psi & \text{Obs}_A \Rightarrow \Phi \)
  - Iterated application of axioms
Logical Assumptions

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

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

**Honest S**

secret(k, G) =

\[
\langle\langle km\rangle\rangle_X \Rightarrow X \in G \\
\& \ (x/k \ y)_X \Rightarrow X \in G
\]

secret(k, [A, S])
Axioms

- Basic truths about domain

  ➢ **Receive axiom**
  
  \[ Y: ((m))_A \Rightarrow \langle m \rangle_X < ((m))_A \]

  ➢ **Timestamp axiom**
  
  \[ A: \text{honest } B & \]

  \[ \langle \tau \rangle_B \prec ((\tau))_A \]

  \[ \Rightarrow (t-\delta)_A \prec (t)_B \prec \langle \tau \rangle_B \prec ((\tau))_A \prec (t-\Delta)_A \]

- Allow inferring new actions/ordering
Schemas and Instances

- Desired functionalities
  - **Nonce-based Challenge-Response property**
    
    \[ A: \Phi \land \\
    (\nu n)_A \prec \langle \langle C \ n \rangle \rangle_{A^<} \prec \langle (R \ n) \rangle_A \\
    \Rightarrow (\nu n)_A \prec \langle \langle C \ n \rangle \rangle_{A^<} \prec \langle (C \ n) \rangle_B \prec \langle \langle R \ n \rangle \rangle_{B^<} \prec \langle (R \ n) \rangle_A \]

- Verified instances
  - **Challenge in the clear/Response encrypted**
    
    \[ A: \text{secret}(K, [A,B]) \land \\
    (\nu n)_A \prec \langle \langle n \rangle \rangle_{A^<} \prec \langle (K \ n) \rangle_A \\
    \Rightarrow (\nu n)_A \prec \langle \langle n \rangle \rangle_{A^<} \prec \langle (n) \rangle_B \prec \langle \langle K \ n \rangle \rangle_{B^<} \prec \langle (K \ n) \rangle_A \]
Abstract Key Distribution

- **S** spontaneously
  - Generates $k$
  - Sends it to $A, B$
    - $A, B$ hardwired
  - Encrypted with $K^A_S, K^B_S$
- **A** observes only $(K^A_S k)$

- **A** reconstructs run
  - Must assume
    - honest $S$
    - secret($K^A_S, [A,S]$)
    - Not secret($K^B_S, [B,S]$)
  - B’s reception unknown

- **Dual for B**

---

**A:** secret($K^A_S, [A,S]$) & honest $S$ & $(K^A_S k)_A$

$\Rightarrow (v \ k)_S < \begin{pmatrix} K^A_S k \\ K^B_S k \end{pmatrix} < (K^A_S k)_A$
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
- A may not be talking to B
  - Even if S honest
- Same for B
I. Cervesato: Encapsulated Authentication Logic

**Binding**

- A (B) authenticated to B (A)

Diagram with nodes labeled as KD0, KD1, KD2, KD3, KD4, NSSK0, NSSKfix0, NSSK1, NSSKfix1, K4core0, K5core0, A, B, S, KAS(B,k), KBS(A,k), v, k.
- 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

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

- With only \( \neg \text{secret}(K_{BS}, [B,S]) \)
  knows S generated k

- With also
  \( \neg \text{secret}(K_{AS}, [A,S]) \)
  knows A knows k
  \( A \) may not be honest
Additional Properties

• Recency
  - \((ν k)_S\) 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

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

\[ K^{BS}(A,k) \]
- 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
Under the assumption

\( \text{secret}(k, [A,B,S]) \)
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

\[
\text{secret}(K^A_S, [A,S])
\]

\[
K^A_S (m,t)
\]

\[
K^A_S m
\]

A \quad S

I. Cervesato: Encapsulated Authentication Logic
Denning-Sacco

- Guarantee recency to both A and B
- Same assurance as core NSSK-fix
  - Only 3 messages
Core Kerberos 4

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

Key confirmation

Repeated auth.

K4core

K5core

K4core

KDS

NSSK

- $A, B$
- $K^{AS}(B, k, t, K^{BS}(A, k, t))$
- $\tau, t'$
- $K^{BS}(A, k, t), k(A, t')$
- $k m[t']$
Core Kerberos 5

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

Key confirmation

Repeated auth.

\( K^A_S(B,k,t) \), \( K^B_S(A,k,t) \)

\( K^B_S(A,k,t), k(A,t') \)

\( k m[t'] \)
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