

Kerberos Version IV: Inductive Analysis of the Secrecy Goals

Pieter Rogaar



Summary

- Introduction
- The inductive method
- Kerberos IV
- Conclusion

Other Proof Techniques

- Belief logics (e.g. BAN)
 - Fast, short proofs
 - Lot of interpretation and assuming
- State exploration methods (e.g. PRISM)
 - Exhaustive search
 - Dangerous assumptions

Structure: Traces

- Set of all events on a network
- Contains two kinds of items:
 - Says A B X
 - Notes Spy X
- Operators *parts*, *analz* and *synth*

Structure: Messages

- Agent names A, B, \dots
- Nonces N_a, N_b, \dots
- Keys K_a, K_b, K_{ab}, \dots
- Compound Messages $\{X, X'\}$
- Hashed Messages $\text{Hash } X$
- Encrypted Messages $\text{Crypt } K X$

Structure: The attacker

- Has access to the trace
- Sometimes *Notes* session keys
- Generates messages from *synth(analz(trace))*

Structure: Rules

- Protocol events
- Fake messages from the spy
- Oops events (session key loss)

Theorems and Proofs

$K \notin []$ (empty trace)

$K \notin \text{tr} \xrightarrow{\text{PR1}} K \in \text{tr}$

$K \notin \text{tr} \xrightarrow{\text{PR2}} K \in \text{tr}$

$K \notin \text{tr} \xrightarrow{\text{PR3}} K \in \text{tr}$

$K \notin \text{tr} \xrightarrow{\text{Fake}} K \in \text{tr}$

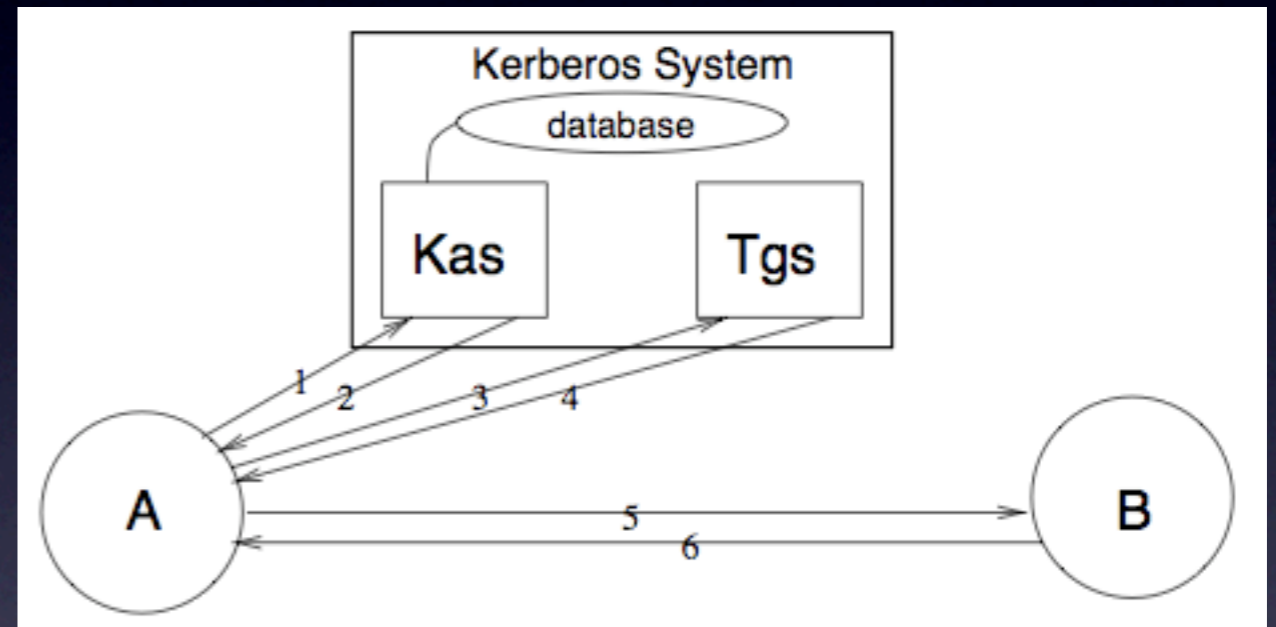
$K \notin \text{tr} \xrightarrow{\text{Oops}} K \in \text{tr}$

Theorems and Proofs

- Regularity Lemmas
 - In terms of the *parts* operator
- Secrecy Theorems
 - In terms of the *analz* operator
- Further Theorems

Kerberos IV: Overview

- A: Alice (client)
- B: Bob (network resource)
- Kas: Kerberos Authentication Server
- Tgs: Ticket Granting Server



Kerberos Protocol Run

Authenticating to acquire an AuthKey

1. $A \rightarrow K_{as}: A, T_{gs}, T_a \mid$

2. $K_{as} \rightarrow A: \{AuthKey, T_{gs}, T_k$
 $\{A, T_{gs}, AuthKey, T_k\}_{K_{tgs}}\}_{K_a}$

Kerberos Protocol Run

Acquiring a ServTicket to access a network resource

3. $A \rightarrow Tgs: \{A, Tgs, AuthKey, Tk\}_{K_{tgs}},$
 $\{A, Ta2\}_{AuthKey}, B$

4. $Tgs \rightarrow A: \{ServKey, B, Tt,$
 $\{A, B, ServKey, Tt\}_{K_b}\}_{AuthKey}$

Kerberos Protocol Run

Using the ServTicket to access the network resource

5. $A \rightarrow B: \{A, B, \text{ServKey}, Tt\}_{K_b}, \{A, T_a3\}_{\text{ServKey}}$

6. $B \rightarrow A: \{T_a3 + I\}_{\text{ServKey}}$

Secrecy Goals

- Key-compromise theorems
 - AuthKeys, ServKeys
- Confidentiality theorems
 - Is AuthKey/ServKey good for communication?

Bob's Confidentiality

- Secure if AuthKey and ServKey not expired
- Bob never sees AuthKey
- If AuthKey lost, ServKey is also lost

Bob's Fix

- Tgs checks whether ServKey won't outlive AuthKey
- Bob then infers freshness of AuthKey from freshness of ServKey

Conclusion

- Inductive method
 - Proofs, not assumptions
 - In large protocols, computations take very long to terminate
- Kerberos IV
 - Most of the secrecy goals are met
 - Strong confidentiality for Bob is possible

Bibliography & Questions

- G. Bella, L.C. Paulson - Kerberos Version IV: Inductive Analysis of the Secrecy Goals
- L.C. Paulson - The Inductive Approach to Verifying Cryptographic Protocols
- Questions?