A Formal Analysis of Some Properties of Kerberos 5 Using MSR

Frederick Butler, Iliano Cervesato, Aaron D. Jaggard, and Andre Scedrov

Project Goals

- Give precise statement and formal analysis of a real world protocol
  - Find a real world protocol – Kerberos 5
  - Pick favorite formalization method - MSR
- Identify and formalize protocol goals
- Give proofs of achieved protocol goals
  - Gain experience in reasoning with MSR
- Note any anomalous behavior
  - Suggest possible fixes, test these

Related Kerberos Work

- **Kerberos 4 - Bella & Riccobene**
  - Gurevich's Abstract State Machine
- **Bella & Paulson**
  - Inductive approach using theorem prover Isabelle
  - Proofs of authentication and confidentiality
  - Incorporated timestamps and temporal checks
- **Kerberos 5 - Mitchell, Mitchell, & Stern**
  - Analyzed simplified protocol with state exploration tool Murφ
  - Attack found, but fixed in full protocol

Related Formal Work

- MultiSet Rewriting (MSR) formalism
  - Lincoln, Mitchell, Scedrov, Durgin, and Cervesato
  - Extended to Typed MSR by Cervesato
- Rank functions
  - Defined by Schneider
  - Our proof methods adapted from this idea

Main Results

- Formalized Kerberos 5 at different levels of detail
  - Typed MSR + extensions
- Observed anomalous behavior
  - Recovery from key loss
  - Some properties of Kerberos 4 do not hold for Kerberos 5
- Proofs of properties which do hold here
  - Methods adapted from Schneider
- Interactions with Kerberos working group

Introduction
Kerberos Overview
Two Views of Kerberos 5
Anomalies
Proof Methods

Protocol Goals and History

- Protocol goals
  - Repeatedly authenticate a client to multiple servers
  - Minimize use of client's long term key(s)
  - Does not guard against DOS attacks
- Kerberos 4 - 1989
- Kerberos 5
  - Specified in RFC 1510 (1993)
  - Subsequent revisions by working group
- A real world protocol
  - Windows 2000 (RFC 1510 + extensions)
  - User login, file access, printing, etc.

Kerberos 5

- Client C wants ticket for end server S
  - Tickets are encrypted – unreadable by C
- C first obtains long term (e.g., 1 day) ticket from a Kerberos Authentication Server K
  - Makes use of C's long term key
- C then obtains short term (e.g., 5 min.) ticket from a Ticket Granting Server T
  - Based on long term ticket from K
  - C sends this ticket to S

Protocol Messages

$$C \xrightarrow{\text{Please give me ticket for } T} K$$

$$C \xleftarrow{\text{Ticket for } C \text{ to give to } T} K$$

$$C \xrightarrow{\text{Ticket from } K\text{, one for } S?} T$$

$$C \xleftarrow{\text{Ticket for } C \text{ to give to } S} T$$

$$C \xrightarrow{\text{Ticket from } T} S$$

$$C \xleftarrow{\text{Confirmation (optional)}} S$$

Introduction

Kerberos Overview

Two Views of Kerberos 5

Anomalies

Proof Methods

Abstract Formalization

- Contains core protocol
  - Other formalization refines this one
- Exhibits an anomaly
  - This appears to be structural and not due to omitted detail
- Allows us to prove authentication results

Messages in Abstract Level

$$C \xrightarrow{\quad C,T,n_1 \quad} K$$

$$C \xleftarrow{\quad C,\{k_{CT},C\}_{k_T},\ \{k_{CT},n_1,T\}_{k_C} \quad} K$$

$$C \xrightarrow{\quad \{k_{CT},C\}_{k_T},\{C\}_{k_{CT}},C,S,n_2 \quad} T$$

$$C \xleftarrow{\quad C,\{k_{CS},C\}_{k_S},\{k_{CS},n_2,S\}_{k_{CT}} \quad} T$$

$$C \xrightarrow{\quad \{k_{CS},C\}_{k_S},\{C,t\}_{k_{CS}} \quad} S$$

$$C \xleftarrow{\quad \{t\}_{k_{CS}} \quad} S$$

Detailed Formalization

- Uses richer message structure
  - Adds some fields for options
    - E.g., anonymous tickets
  - Models encryption type
  - Adds checksums
- Exhibits anomalies
  - Encryption type option specific to this level
  - Structural anomaly also seen at abstract level
    - Also variations which use added detail

Messages in Detailed Level

$$C \xrightarrow{KOpts, C, T, n_1, e_1} K$$

$$C \xleftarrow{C, \{Tflags, k_{CT}, C\}_{k_T}, \{k_{CT}, n_1, Tflags, T\}^{e_1'}_{k_C}} K$$

$$C \xrightarrow{\{Tflags, k_{CT}, C\}_{k_T}, \{C, MD, t\}_{k_{CT}}, Topts, C, S, n_2, e_2} T$$

$$C \xleftarrow{C, \{Sflags, k_{CS}, C\}_{k_S}, \{k_{CS}, n_2, Sflags, S\}^{e_2'}_{k_{CT}}} T$$

$$C \xrightarrow{SOpts, \{Sflags, k_{CS}, C\}_{k_S}, \{C, MD', t'\}_{k_{CS}}} S$$

$$C \xleftarrow{[\{t'\}^{e}_{k_{CS}}]} S$$

$$C \xleftarrow{KRB\_ERROR, [-|t|t'], t_{err}, ErrCode, C, (K|T|S)} K|T|S$$

Introduction
Kerberos Overview
Two Views of Kerberos 5
Anomalies
Proof Methods

Encryption Type Anomaly

- Kerberos 5 allows C to specify encryption types that she wants used in K's response

$$C \xrightarrow[\text{etype (sent unencrypted)}]{\text{Please give me ticket for T using}} K$$

$$C \xleftarrow[\text{info encrypted using etype)}]{\text{Ticket for C to give to T (other}} K$$

- C's key associated with the etype e_{bad} is k_{bad}
  - Intruder I learns k_{bad}
  - C knows this and attempts to avoid e_{bad}/k_{bad}
  - I can still force k_{bad} to be used
  - How to recover from a lost key

Ticket Anomaly

$C \xleftarrow{\text{Ticket for } C \text{ to give to } T} K$

- **Kerberos 4:**
  - Ticket is enclosed in another encryption

$$\xleftarrow{\{\text{Ticket, Other data}\}_{k_C}}$$

- **Kerberos 5:**
  - Ticket is separate from other encryption

$$\xleftarrow{\text{Ticket, }\{\text{Other data}\}_{k_C}}$$

Ticket Anomaly

- T grants the client C a ticket for S
- C has never sent a proper request for a ticket
  - C never has the ticket for T
  - C thinks she has sent a proper request
  - C's view of the world is inaccurate
  - Some properties of Kerberos 4 don't hold here
- Seen in both formalizations
  - Variations possible using added detail
    - Anonymous tickets
- Still can authenticate origin of data

Comments from Kerberos Designers

- Generally positive response
  - Methods helpful
  - Encouraged to pursue further
  - Should look at protocol extensions
- Anomalies
  - These scenarios can occur
  - Practical concern unclear
  - Anonymous ticket variation of interest
    - Status of this option may change
    - Good to highlight possible concerns here

Introduction
Kerberos Overview
Two Views of Kerberos 5
Anomalies
Proof Methods

Rank and Corank

- Inspired by work of Schneider
- Define functions on MSR facts
  - k-Rank – encryptions by k
    - Data origin authentication
  - E-Corank – level of protection by keys in E
    - Secrecy
- Proofs
  - State desired property
  - Find applicable (co)rank functions
  - Determine effect of MSR rules on these functions

An Authentication Theorem

- If T processes the message

  $\{k_{CT},C\}_{k_T},\{C\}_{k_{CT}},C,S,n_2$

  then some K sent the message

  $C,\{k_{CT},C\}_{k_T}, \{k_{CT},n_1,T\}_{k_C}$

  and C sent *some* message

  $X,\{C\}_{k_{CT}},C,S',n'_2$

- Authenticate data origin using rank
  - Show ticket $\{k_{CT},C\}_{k_T}$ originates with some K
  - Show authenticator $\{C\}_{k_{CT}}$ originates with C
    - This makes use of a corank argument for confidentiality
- In Kerberos 4, C *must* have sent the ticket and *not* the generic X (Bella & Paulson)

A Second Authentication Theorem

- If S processes the message

  $\{k_{CS},C\}_{k_S},\{C,t\}_{k_{CS}}$

  then some T sent the message

  $C,\{k_{CS},C\}_{k_S}, \{k_{CS},n_2,S\}_{k_{CT}}$

  and C sent *some* message

  $X,\{C,t\}_{k_{CS}}$

Conclusions

- Formalizations of Kerberos 5 at different levels of detail
  - Used MSR + extensions
  - MSR can handle real world protocols
- Anomalous behavior
  - Stated weakened authentication properties which hold for Kerberos 5
- Proofs of properties which hold here
  - Adapted methods from Schneider
  - Gained additional experience in reasoning with MSR
- Interactions with Kerberos designers

Future Work

- Investigate fixes for anomalies
- Look at additional properties
  - Further authentication, confidentiality
  - Defense against replay attacks
- Continue interaction with Kerberos designers
- Give additional formalizations
  - Additional structure and functionality
  - Public key extensions
- Explore use of automated tools