An Update on Network Protocol Security

Anupam Datta
Stanford University

Stanford Computer Forum, 2007
Roadmap

- Network protocol examples
- Are industrial protocols secure?
  - Case studies of industry standards
- Research state-of-the-art
  - Fully automated bug-finding tools
  - Methods for proving absence of bugs
    - Protocol Composition Logic
      - Modular proof-techniques
      - Cryptographic soundness
- Conclusions and Future Work
Many Protocols

- Authentication
  - Kerberos
- Key Exchange
  - SSL/TLS handshake, IKE, JFK, IKEv2,
- Wireless and mobile computing
  - Mobile IP, WEP, 802.11i, 802.16e, Wi-Fi
- Electronic commerce
  - Contract signing, SET, electronic cash, ...
802.11i Wireless Authentication

- Supplicant
- UnAuth/UnAssoc
- 802.1X Blocked No Key
- 802.11 Association
- 802.11i Wireless Authentication
- EAP/802.1X/RADIUS Authentication
- 4-Way Handshake
- Group Key Handshake
- Data Communication
- MSK
- Widely used in wireless LANs
TLS protocol layer over TCP/IP

Widely used on internet
IKE sub-protocol from IPSEC

Result: A and B share secret $g^{ab} \mod p$

Used in corporate Virtual Private Networks
Kerberos Protocol

Running example in this talk

Client

KAS

AS-REQ

AS-REP

Client

TGS

TGS-REQ

TGS-REP

Client

Server

AP-REQ

AP-REP

Used for network authentication
Roadmap

- Network protocol examples
- Are industrial protocols secure?
  - Case studies of industry standards
- Research state-of-the-art
  - Fully automated *bug-finding* tools
  - Methods for proving *absence of bugs*
    - Protocol Composition Logic
      - Modular proof-techniques
      - Cryptographic soundness
- Conclusions and Future Work
Microsoft Security Bulletin MS05-042
Vulnerabilities in Kerberos Could Allow Denial of Service, Information Disclosure, and Spoofing (899587)
Published: August 9, 2005

Affected Software:
- Microsoft Windows 2000 Service Pack 4
- Microsoft Windows XP Service Pack 1 and Microsoft Windows XP Service Pack 2
- Microsoft Windows XP Professional x64 Edition
- Microsoft Windows Server 2003 and Microsoft Windows Server 2003 Service Pack 1
- Microsoft Windows Server 2003 for Itanium-based Systems and Microsoft Windows Server 2003 with SP1 for Itanium-based Systems
- Microsoft Windows Server 2003 x64 Edition
Kerberos Error

Formal analysis of Kerberos 5
• Several steps
  - Detailed core protocol
  - Cross-realm authentication
  - Public-key extensions to Kerberos

Attack on PKINIT
• Breaks association of client request and the response
• Prevents full authentication and confidentiality

Formal verification of fixes preventing attack
• Close, ongoing interactions with IETF WG

I. Cervesato, A. D. Jaggard, A. Scedrov, J.-K. Tsay, and C. Walstad
Public-Key Kerberos

◆ Extend basic Kerberos 5 to use PKI
  • Change first round to avoid long-term shared keys
  • Originally motivated by security
    - If KDC is compromised, don’t need to regenerate shared keys
    - Avoid use of password-derived keys
  • Current emphasis on administrative convenience
    - Avoid need to register in advance of using Kerberized services

◆ This extension is called PKINIT
  • Current version is PKINIT-29
  • Attack found in PKINIT-25; fixed in PKINIT-27
  • Included in Windows and Linux (called Heimdal)
  • Implementation developed by CableLabs (for cable boxes)
At time $t_C$, client $C$ requests a ticket for ticket server $T$ (using nonces $n_1$ and $n_2$):

$C \xrightarrow{\text{Cert}_C, [t_C, n_2]_{sk_C}, C, T, n_1} I$

The attacker $I$ intercepts this, puts her name/signature in place of $C$'s:

$I \xrightarrow{\text{Cert}_I, [t_C, n_2]_{sk_I}, I, T, n_1} K$

Kerberos server $K$ replies with credentials for $I$, including: fresh keys $k$ and $AK$, a ticket-granting ticket $TGT$, and $K$'s signature over $k,n_2$:

(Ignore most of enc-part)

$I \xleftarrow{[k, n_2]_{sk_K}}_{pk_K}, I, TGT, \{AK, \ldots\}_k K$

$I$ decrypts, re-encrypts with $C$'s public key, and replaces her name with $C$'s:

$C \xleftarrow{[k, n_2]_{sk_K}}_{pk_C}, C, TGT, \{AK, \ldots\}_k I$

- $I$ knows fresh keys $k$ and $AK$
- $C$ receives $K$'s signature over $k,n_2$ and assumes $k$, $AK$, etc., were generated for $C$ (not $I$)
- Principal $P$ has secret key $sk_P$, public key $pk_P$
- $\{\text{msg}\}_{\text{key}}$ is encryption of $\text{msg}$ with key
- $[\text{msg}]_{\text{key}}$ is signature over $\text{msg}$ with key
The KDC signs k, cksum (place of k, n_2)
  - k is replyKey
  - cksum is checksum over AS-REQ
  - Easier to implement than signing C, k, n_2

Formal proof: this guarantees authentication
  • Assume checksum is preimage resistant
  • Assume KDC’s signature keys are secret
Attacks on Industry Standards

- **IKE** [Meadows; 1999]
  - Reflection attack; fix adopted by IETF WG
- **IEEE 802.11i** [He, Mitchell; 2004]
  - DoS attack; fix adopted by IEEE WG
- **GDOI** [Meadows, Pavlovic; 2004]
  - Composition attack; fix adopted by IETF WG
- **Kerberos V5** [Scedrov et al; 2005]
  - Identity misbinding attack; fix adopted by IETF WG; Windows update released by Microsoft

Identified using logical methods
Roadmap

◆ Network protocol examples
◆ Are industrial protocols secure?
  • Case studies of industry standards
◆ Security analysis state-of-the-art
  • Fully automated *bug-finding* tools
  • Methods for proving *absence of bugs*
    - Protocol Composition Logic
      • Modular proof-techniques
      • Cryptographic soundness
◆ Conclusions and Future Work
Security Analysis Methodology

- Kerberos
- Protocol
- Property
  - Authentication
  - Attacker model
  - Security proof or attack
- Analysis Tool
  - Our tool: Protocol Composition Logic (PCL)
  - ~40 line axiomatic proof
  - Complete control over network
  - Perfect crypto
Automated Finite-State Analysis

Define finite-state system
- Bound on number of steps
- Finite number of participants
- Nondeterministic adversary with finite options

Pose correctness condition
- Authentication, secrecy, fairness, abuse-freeness

Exhaustive search using “verification” tool
- Error in finite approximation $\Rightarrow$ Error in protocol
- No error in finite approximation $\Rightarrow$ ???

Example
- SSL analysis with 3 clients and 2 servers
Bug-finding Tools and Case Studies

◆ Murphi model-checking of protocols
  • Generic model-checker developed by David Dill’s group at Stanford
  • Method for security protocol analysis developed by Mitchell, Shmatikov et al (1997-)
  • Many case studies including SSL, 802.11i
  • Tool and case studies available at http://cs259.stanford.edu
◆ Many other fully automated tools
  • AVISPA project, SRI constraint solver, ...
◆ Ready for industrial use
Roadmap

- Network protocol examples
- Are industrial protocols secure?
  - Case studies of industry standards
- Security analysis state-of-the-art
  - Fully automated *bug-finding* tools
  - Methods for proving *absence of bugs*
    - Protocol Composition Logic
      - Modular proof-techniques
      - Cryptographic soundness
- Conclusions and Future Work
Proving Security of Protocols

Cryptographic reductions
- More realistic model involving probabilistic polynomial time attackers
- Difficult to scale to industrial protocols

Symbolic methods and proof tools
- NRL Protocol Analyzer, Paulson’s Inductive Method, Process calculi, Specialized protocol logics (see http://cs259.stanford.edu)
- 2 challenges:
  - Scale to industrial protocols: modular analysis desired
  - Use cryptographic model instead of idealized symbolic model
- Progress on challenges in last 5 years
Our Result

Protocol Composition Logic (PCL):

• Unbounded number of sessions (vs. model-checking)
• Short high-level proofs: 2-3 pages
• Sound wrt
  - symbolic model
  - computational cryptography model
• Modular proof techniques

[DDMP03, ..., RDDM06]
PCL Results: Industrial Protocols

- IEEE 802.11i [IEEE Standards; 2004] [HSDDM05]
  - TLS/SSL [RFC 2246] is a component
    (Attack using model-checking; fix adopted by WG)
- GDOI Secure Group Communication [RFC 3547] [MP04]
  (Attack using PCL; fix adopted by IETF WG)
- Kerberos V5 [IETF ID; 2004] [CMP05,RDDM06]

- Mobile IPv6 [RFC 3775] in progress [RDM06]
- IKE/JFK family
  - IKEv2 [IETF ID;2004] in progress [RDM06]

Except Kerberos, results currently apply only to symbolic model
A protocol is a set of programs, one for each role
**PCL: Syntax**

- **Action formulas**
  \[ a ::= \text{Send}(P,t) \mid \text{Receive}(P,t) \mid \ldots \]

- **Formulas**
  \[ \varphi ::= a \mid \text{Indist}(P,t) \mid \text{GoodKeyAgainst}(X,k) \mid \text{Honest}(N) \mid \neg \varphi \mid \varphi_1 \land \varphi_2 \mid \exists x \varphi \mid a < a \mid \ldots \]

- **Modal formula**
  \[ \varphi [ \text{actions} ]_P \varphi \]

- **Examples**
  - secret indistinguishable from random
    \[ (X \neq A \land X \neq B) \supset \text{Indist}(X, \text{secret}) \]
Kerberos Stage 1 Property

- **Client guarantee**

  \[
  \text{true } [ \text{Client1}(C, T, K) ]_C \\
  \text{Honest}(C, T, K) \supset \\
  (\text{GoodKeyAgainst}(X, AKey) \lor \\
  X \in \{C, T, K\})
  \]

- **Key usable for encryption**
Complexity-theoretic semantics

\( Q \models \varphi \) if \( \forall \) adversary \( A \) \( \forall \) distinguisher \( D \) \( \exists \) negligible function \( f \) \( \exists n_0 \ \forall n > n_0 \) s.t.

\[
\frac{|[[\varphi]](T,D,f(n))|}{|T|} > 1 - f(n)
\]

- Fix protocol \( Q \), PPT adversary \( A \)
- Choose value of security parameter \( n \)
- Vary random bits used by all programs
- Obtain set \( T = T(Q,A,n) \) of equi-probable traces
PCL: Proof System

◆ Property of signature:
  \[\text{Honest}(X) \land \text{Verifies}(Y, m, X) \Rightarrow \text{Signed}(X, m)\]

◆ Soundness proof:
  ◆ Assume axiom not valid
    \[\exists A \exists D \forall \text{negligible } f \forall n_0 \exists n > n_0 \text{ s.t.} \]
    \[\|\varphi\|(T, D, f(n))/|T| < 1 - f(n)\]
  ◆ Construct attacker A' that uses A, D to break CMA-secure signature scheme
  ◆ Standard cryptographic reduction

[DDMST05, DDMW06]
Inductive Secrecy

- Pick a nonce \( s \) and set of keys \( K = \{k_0, k_1, k_2\} \)

- Secretive\((s, K)\)
  - Terms explicitly containing \( s \) are encrypted by a key in \( K \) before sending out.
  - New terms obtained through decryption by a key in \( K \) are re-encrypted by a key in \( K \) before sending out by an honest principal.

[RDDM06]
Inductive Secrecy $\Rightarrow$ “Good” Keys

- **Secrecy axiom**
  \[
  \text{Secretive}(s, K) \land \text{GoodInit}(s, K) \Rightarrow \text{GoodKeyFor}(s, K)
  \]

- **Read**
  - If
    - protocol is “secretive”
    - nonce-generator is honest
    - key-holders are honest
  then
    - the key generated from the nonce is a “good” key (usable for encryption)

  *Soundness proof is by reduction to a multi-party IND-CCA game [BBM00]*
  *One-time effort*
CPCL analysis of Kerberos V5

- Kerberos has a staged architecture
  - First stage generates a nonce and sends it encrypted
  - Second stage uses this nonce as a key to encrypt another nonce.
  - Third stage uses the nonce exchanged in the second stage to encrypt other terms
- We prove “GoodKey”-ness of both the nonces assuming encryption scheme is IND-CCA
- Authentication properties proved assuming encryption scheme is INT-CTXT secure
- Modular proofs (including PKINIT) using composition theorems
- Result by Boldyreva et al showing that encryption scheme provides required properties
Logic and Cryptography: Big Picture

- Complexity-theoretic crypto definitions (e.g., IND-CCA2 secure encryption)
- Crypto constructions satisfying definitions (e.g., Cramer-Shoup encryption scheme)
- Semantics and soundness theorem
- Axiom in proof system
- Protocol security proofs using proof system
Conclusions

◆ Practical protocols may contain errors
  • Automated methods find bugs that humans overlook
◆ Variety of tools
  • Model checking can find errors
  • Proof method can show correctness
    – with respect to specific model of execution and attack
◆ Modular analysis is a challenge
◆ Closing gap between logical analysis and cryptography
  • Symbolic model supports useful analysis
    – Tools, case studies, high-level proofs
  • Computational model more informative
    – Includes probability, complexity
    – Does not require strong cryptographic assumptions
    – More accurately reflects realistic attack
  • Two approaches can be combined
    – Several current projects and approaches [BPW, MW, Blan, CH, ...]
    – One example: computational semantics for symbolic protocol logic

◆ Research area coming of age
  • Interactions with and impact on industry
Thanks!

Questions?