Multiprotocol Attacks and the Public Key Infrastructure*

Jim Alves-Foss
Center for Secure and Dependable Software
University of Idaho
http://www.cs.uidaho.edu

*Supported in part by NSA Grant MDA 904-1-0108
What are Multiprotocol Attacks?

- Multiprotocol Attack
  - Interleaves messages from two separate protocols to attack one of them.
  - The attacked protocol is subverted using either:
    - An incidental collision with another protocol.
    - A deliberately *tailored* protocol.
  - An attacker may successfully masquerade as client A to server B using protocol P, even if A does not support P.
Why the Public-Key Infrastructure

- Attacks in this work are specific to public-key protocols.
  - Work for a shared, certified key
  - Work for newly generated, self-certified keys.
  - Work for fully signed messages, or signed hashes of messages.
  - Work against public-key usage for privacy.
  - May not work against all private-key protocols.
Cryptographic Protocol Notation

➤ Encryption
   ➤ \{\ldots\}_{K_{AB}} - Using private key shared between A and B
   ➤ \{\ldots\}_{K_A} - Using the public part of A’s public key
   ➤ \{\ldots\}_{K_{A^{-1}}} - Using the private part of A’s public key

➤ Other Techniques
   ➤ H(\ldots) - Hashing
   ➤ R_A - random value generated by A (for use as a nonce or part of a Diffie-Hellman key-distribution)
A “Secure” Protocol

\[ \{A, R_B, R_A\}_{K_B^{-1}} \]

\[ \{B, R_B\}_{K_A^{-1}} \]

“Protocol 1 - mutual authentication”

Simple Tailoring of a Protocol

M

A

R_B

{B,R_B}_{K_A^{-1}}

B

“Protocol 2 - one-way authentication”

Attack Against B in Protocol 1

A \rightarrow E_B \text{ with } M

E_A \rightarrow R_A \rightarrow B

A \leftarrow R_B

\{B, R_B\}_{K_A^{-1}} \rightarrow E_B

E_A \leftarrow \{A, R_B, R_A\}_{K_B^{-1}} \rightarrow B

E_A \leftarrow \{B, R_B\}_{K_A^{-1}} \rightarrow B
A Portion of a “Secure” Protocol

\[
\begin{align*}
\{B, M_1, M_2, M_3, K_{AB}, M_4, R_B \}^{\text{K}_A} \\
\{A, B, R_B \}^{\text{K}_{AB}}
\end{align*}
\]

“Protocol 3 - Portion of a Key Distribution Protocol”
Simple Tailoring of a Protocol

“Protocol 4 - Tailored Decoding Protocol”

A \{B,M_1,M_2,M_3,R_{B1},M_4,R_{B2}\}_K_A

\{A,E,R_{B1},R_{B2}\}_K_E

E
Attack Against B in Protocol 3

Attack A against B in Protocol 3:

- A → E: \{E,M_1,M_2,M_3,R_{B1},M_4,R_{B2}\}_{K_A}
- E → A: \{B,M_1,M_2,M_3,K_{AB},M_4,R_B\}_{K_A}
- A → E: \{A,E,R_{B1},R_{B2}\}_{K_E}
- E → A: \{A,B,R_B\}_{K_{AB}}
- E → B: \{B,M_1,M_2,M_3,K_{AB},M_4,R_B\}_{K_A}
Protection Against Tailored Protocol Attacks

Why do the attacks occur?

1. Keys (even certified keys) may be shared between multiple protocols.
2. Tailored (or chosen) protocol is installed on a victim’s machine.
Protection Against Tailored Protocol Attacks

➤ How do we stop the attacks?

➤ Kelsey, et. al:
  ➤ Limit the scope of the key
  ➤ Uniquely identify each application, protocol, version and protocol step
  ➤ All protocols should have a fixed unique identifier in a fixed position in the message
  ➤ Tie the unique identifier to encryption
  ➤ Include support in smartcards
Protection Against Tailored Protocol Attacks

➤ Do these work?
➤ For smartcards they may, but not for general computers.
  ➤ Requirements that insist on a unique identifier assumes that protocols follow the rules, a *tailored* protocol need not follow the rules.
  ➤ Without these identifiers, we can not limit key usage to a particular protocol.
Solution

➤ What is the solution?
  ➤ We must limit key usage to protected/trusted subsystems.
  ➤ The subsystems must only allow encryption by certified applications, (those that follow the rules).
  ➤ Operating system security must be in place to protect subsystems and stored keys.
Challenges

➤ Enhance PKI certificates to include protocol limitations

➤ Develop specific guidelines for protocol message content identifiers

➤ Enforce guidelines, limitations, and trust model in key management and crypto packages for protocols

➤ Establish protocol certification authority

➤ Prevent user apps from accessing certified keys
Suggested Protocol Architecture

➤ Develop a protocol message specification language.
  ➤ The protocol developer obtains certification of protocol message set, and releases to application developers
  ➤ Protocol application submits certification to crypto library to establish protocol
  ➤ Subsequent calls to crypto library specify protocol and message identifiers; crypto library performs operation ONLY if message format matches specification