Public Key Protocols for Wireless Communications

Colin Boyd, Dong_Gook Park

1998

Information Security Research Centre
Queensland University of Technology,
Brisbane Australia
boyd@fit.qut.edu.au

Wireless Communications Research
Laboratory
Korea Telecom
Park@isrc.qut.edu.au

Preview

♦ Overview
♦ Background
♦ Requirements for Mobile Protocols
  – Key Agreement
  – Key Transport
♦ Recent Protocol Proposals
  – Park’s Protocol
    • Attack on Park’s protocol
  – The ASPeCT protocol
♦ A New Proposal
♦ Review
♦ Questions
Overview

♦ In next generation of wireless environment,
  – More important security requirements
  – Special requirements for authentication and key establishment protocols
  – More likely, public key based protocols

♦ Most critical security interface, between user and network
  – Preventing false access to network resources
  – Preserving user privacy

♦ Shared key establishment between two entities

♦ Protection of digitally encoded speech and control information by
  – symmetric encryption and
  – integrity mechanisms

Background

♦ Current Second generation authentication protocols
  – Shared long term key between users and their home networks to establish session keys
  – Home authentication center
    • be on-line at the time of setup
    • Provide a high level of reliability and availability (expensive…)

♦ Third generation systems
  – More widely distributed
  – More problems
  – Solution: asymmetric public key cryptography
    • No need for on-line server
    • Not available for second generation systems
      – Computational limitations of handheld devices
Goal of the paper

♦ For key establishment and authentication in third generation wireless systems,
  – examine various aspects of the use of public key based protocols

Requirements for Mobile Protocols

♦ European ASPeCT Project: public key based protocols for third generation wireless systems

♦ Six goals for authentication protocols between mobile entities and fixed network (proposed by Horn and Preneel):
  – HP1. Mutual authentication of user and network
  – HP2. Agreement between user and network on a secret session key with mutual implicit key authentication
  – HP3. Mutual key confirmation
  – HP4. Mutual assurance of key freshness (mutual key control)
  – HP5. Non-repudiation of origin for relevant user data
  – HP6. Confidentiality of relevant data

♦ And addition to above
  – Limited power of mobile hand set
    • Limited size
    • Power and storage capabilities
Key Agreement/Key Transport

♦ Key establishment protocols classified as:
  – key transport protocols :
    • one entity in the protocol chooses the session key
      unilaterally and sends it encrypted to the other entity
  – key agreement protocols:
    • Both entities contribute to the session key.
    • Based on Diffie-Hellman key exchange
    • Nonce-like random property
      – Assurance, resultant key value is fresh

Some arguments for Diffie-Hellman key agreement protocols

♦ 1. Key control
  – Key quality: either the mobile or the network not sufficiently component to chose session key
♦ 2. No encryption required
  – Complex export rules in different countries
  – Protocols without explicit encryption steps easier to export
  – Diffie-Hellman key agreement requires only signatures not encryption
♦ 3. Forward secrecy
  – Attractive property for Diffie-Hellman Key agreement:
    • If the long-term private key of any user becomes compromised
    • Does not allow previous session keys to be found by an attacker.
  – Not shared by key transport protocols
Recent Protocol Proposals

Notation

A: mobile user
B: network
x_A: A’s private key
x_B: B’s private key
y_A: A’s public key
y_B: B’s public key
p: a large prime
g: value (discrete algorithm problem with respect to g is believed to be difficult)
r_A: random value chosen by A
r_B: random value chosen by B
K: symmetric key
{X}_K: encryption of message X with K.

Park’s protocol

♦ Modified version of earlier protocol by Yacobi and Shmuely

<table>
<thead>
<tr>
<th>A: mobile user</th>
<th>B: network</th>
</tr>
</thead>
<tbody>
<tr>
<td>x_A: A’s private key</td>
<td>x_B: B’s private key</td>
</tr>
<tr>
<td>r_A: random value chosen by A</td>
<td>r_B: random value chosen by B</td>
</tr>
<tr>
<td>y_A = g^-x_A: A’s public key</td>
<td>y_B = g^-x_B: B’s public key</td>
</tr>
</tbody>
</table>

Original Yacobi and Shmuely protocol:

- B → A: x_B + r_B
- A → B: x_A + r_A

K_{AB} = g^{-r_A r_B}: session key
K_{AB} = (g^{x_A r_B y_B})^{-r_A}: session key calc. by A
K_{AB} = (g^{x_B r_A y_A})^{-r_B}: session key calc. by B

Parks protocol:

- B → A: g^{x_B r_B}
- A → B: x_A + r_A

K_{AB} = g^{-r_A r_B}: session key
New attack on Park’s Protocol

♦ In original Yacobi-Shmuely protocol
  − $x_B + r_B$ is sent by B from any entity that can retrieve $g^{r_B}$.
  − No entity can form the message $x_B + r_B$ because $x_B$ only known to B.

♦ In Park’s protocol,
  − any entity E, can form $y_B^{-1} g^{r_B} = g^{x_B + r_B}$ from $y_B$ and
  − $g^{r_B}$ with any random value $r_B$ chosen by E.

Therefore, no signature effect in modified form of $g^{x_B + r_B}$.

♦ Any entity E can execute this protocol successfully with entity A without the knowledge of private key of B. The attack procedure:

1. $E \xrightarrow{\cdot} A : x_A + r_A$
2. $A \xrightarrow{\cdot} E : g^{x_A + r_A}$

$E$ arbitrarily random value chosen by B not E.

♦ $E$ can calculate session key just like B does as $$K_{AB} = (y_A g^{x_A + r_A}) r_B = g^{r_B}$$

Entity A believes the entity B is prepared to communicate with A and established the same session key with A.

♦ Attacker successfully derives the same session key with A.

The ASPeCT protocol

A: mobile user
B: network
$y_A = g^{x_A}$: A’s public key
$y_B = g^{x_B}$: B’s public key
CA: trusted certification authority
$K_{KS} : g^{r_X} r_D$: session key
$K_{AB} = (g^{x_B+r_B} y_A)^{r_D}$: session key calc. by A
$K_{AB} = (g^{x_A+r_A} y_B)^{r_D}$: session key calc. by B
$h_1, h_2, h_3$: hash functions
$\text{Sig}_A(X)$: A’s signature transformation on message X
$\text{Acert}$: A’s certificate (contains A’s public sign. info)
$\text{Beert}$: B’s certificate (contains B’s public key $g^b$)
$\text{chd}$: charging data
$\text{pay}$: payment data
$T_B$: time stamp issued by B.

The ASPeCT protocol:

- $A \rightarrow B : g^{r_A}$, $CA$
- $B \rightarrow A : r_B, h_1(K_{AB}, r_B, B), \text{chd}, T_B, \text{Beert}$
- $A \rightarrow B \text{ Sig}_{A}(h_3(g^{r_A} g^b, B, \text{chd}, \text{Ts}, \text{pay}), \text{Acert}, \text{pay}) K_{KS}$

The session key calculated by A: $K_{KS} = h_1(r_B, (y_A)^{r_B})$
The session key calculated by B: $K_{KS} = h_1(r_B, (g^{x_B} + b)^{r_B})$
The weakness of the ASPeCT Protocol

- The delay in the identification of the entity A to point of message 3

A New Proposal

- COUNT: orthogonal to the requirements HP1-HP6, instead it is used to detect cloning fraud in mobile handsets
- Two versions:
  - Provides key transport and
  - Provides key agreement

1. \( A \rightarrow B : \text{Enc}_B(A, K_{AB}, \text{COUNT}) \)
2. \( B \rightarrow A : \{\text{COUNT}, r_B\} K_{AB} \)
3. \( A \rightarrow B \text{ Sig}_A(B, h(\text{COUNT}, K_{AB}, r_B)) \)

1. \( A \rightarrow B : \text{Enc}_A(A, r_A, \text{COUNT}) \)
2. \( B \rightarrow A : \{\text{COUNT}, r_A\} K_{AB} \)
3. \( A \rightarrow B \text{ Sig}_A(B, h(\text{COUNT}, r_A, K_{AB})) \)

- Session key: \( K_{AB} = h(r_A \cdot r_B) \)
- Disadvantage: Lack of forward secrecy
Conclusions

♦ Weakness of some recent protocols
♦ Key agreement using Diffie-Hellman protocols is unnecessary and computationally expensive

Review

♦ Overview
♦ Background
♦ Requirements for Mobile Protocols
  – Key Agreement
  – Key Transport
♦ Recent Protocol Proposals
  – Park’s Protocol
    • Attack on Park’s protocol
  – The ASPeCT protocol
♦ A New Proposal
♦ Review
♦ Questions
Questions?