The Computation Result Protection

- Introduction
- Notations and Security requirements.
- Possible attacks on collected computation result
- Overview of KAG protocol and Problem statement.
- Related work and its vulnerability
- Solutions
- Security Analysis
- Conclusion and Future work

Introduction

- What is the mobile agent?
- Why it is necessary to protect the result of computation carried by a mobile agent?
- Goal of this paper

Notations (1)

Π	Code
TTP	Trusted Third Party
ψ_i	Protected list of already visited host at S_i
S_0	ID of the originator
S_i	ID of server i
o_0	Dummy offer from originator
o_i	An offer for S_i
O_i	An encapsulated offer from S_i
$O_0, O_1, \ldots., O_n$	The chain of encapsulated offer
r_i	A nonce generated by S_i
T_{S_i}	Timestamp chosen by S_i
$H(m)$	A one-way collision resistant hash function

Notations (2)

$(\upsilon_i, \overline{\upsilon_i})$	A public/private key pair of S_i.
$(y_i, \overline{y_i})$	A one time key pair to be used by S_i. The key pair is generated by S_{i-1}.
$(\mu_i, \overline{\mu_i})$	A one time key pair to be used by S_{i+1} and S_i. The key pair is generated by TTP.
$(\sigma_i, \overline{\sigma_i})$	A one time key pair to be used by S_{i+1} and S_i. The key pair is generated by S_i. When TTP is offline.
$SiG_{\overline{\upsilon_i}}(m)$	Signature of S_i on message *m*.
$ENC_{\upsilon_i}(m)$	Message *m* encrpyted with the key associated with S_i.

Notations (3)

$S_0 \rightarrow S_1 : m$	S_0 sending the message m to S_1
$\alpha_{S_i}, \alpha_{S_{i+1}}$	Random integer for ephemeral key chosen by S_i and S_{i+1}.
$t_{S_i}, t_{S_{i+1}}$	Ephemeral public key:
$Z_{S_i S_{i+1}}$	The share secret computed by S_i and S_{i+1}.
$K_{i,i+1}$	The session key calculated from key derivation function.
G	A subgroup of $\mathbb{Z}_p^*$ and g a generator of G
p	A large prime
q	A prime with $q \mid p-1$

Security Requirements

1. *Data Confidentiality*
2. *Non-Repudiability*
3. *Forward Privacy*
4. *Strong Forward Integrity*
5. *Insert Resilience*
6. *Truncation Resilience*

Possible Attacks (1)

After capturing, an agent holds a chain $O_0, O_1, \ldots, O_m$
The attacker might :

- *Modify*

$$O_0, O_1, O_2, O_3, \ldots, O_m \longrightarrow O_0, O_1, O_2, O_3^M, \ldots, O_m$$

- *Insert*

$$O_0, O_1, O_2, O_3, \ldots, O_m \longrightarrow O_0, O_1, O_2, O_D^I, O_3, \ldots, O_m$$

- *Delete*

$$O_0, O_1, O_2, O_3, \ldots, O_m \longrightarrow O_0, O_3, \ldots, O_m$$

Possible Attacks (2)

- *Truncation*

$$O_0, O_1, O_2, O_3, \ldots, O_m \longrightarrow O_0, O_1, O_2, O_3^T, \ldots, O_{m-1}^T, O_m$$

- *Collusion**

There are at least two hosts who perform attacks, e.g deletion,truncation, on the chain of encapsulted offer without being detected.

Overview of KAG protocol (P4)

Assumption:

1. There is no Public Key Infrastructure.
2. Every visited host knows originator's public key.

- At the originator

1. Offer Encapsulation: at the originator

$$O_0 = SIG_{\overline{v_0}}(ENG_{v_0}(o_0, r_0), h_o, y_1)$$

$$h_0 = H(r_0, S_1)$$

2. Agent Transmission:

$$S_0 \rightarrow S_1 : \Pi, O_0, \left[\overline{y_1} \right]$$

Overview of KAG protocol (P4)

At host S_1:

- Encapsulated offer verification:

  Host receives $\Pi, O_0, \left[\ \overline{y_1}\ \right]$

1. By checking the encapsulated offer from originator O_0

$$O_0 = SIG_{\overline{v_0}}(ENG_{v_0}(o_0, r_0), h_o, y_1)$$

- Offer Encapsulation : S_1 should do as follows;

$$O_1 = SIG_{\overline{y_1}}(ENG_{v_0}(o_1, r_1), h_1, y_2)$$
$$h_1 = H(O_0, S_2)$$

Overview of KAG protocol (P4)

At host S_n:

- Encapsulated offer verification

  Host receives $\Pi, \{O_k \mid 0 \le k \le n-1\}, [\ \overline{y_n}\ ]$

1. By checking the chain of encapsulted offers $O_0, O_1, \ldots, O_{n-1}$

$ver(O_0, v_0)$ Recover h_0, y_1

$ver(O_1, y_1)$

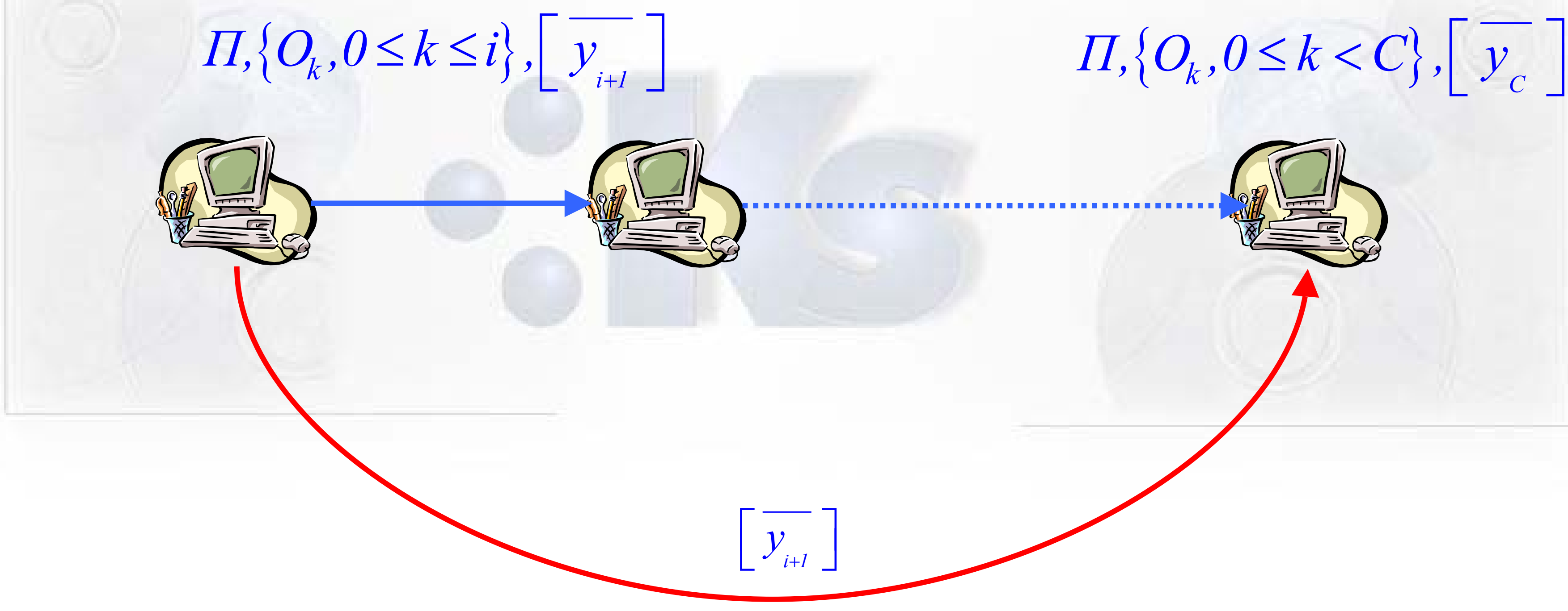
$ver(O_{n-1}, y_{n-1})$ h_{n-1}, y_n

- Offer Encapsulation : S_n should do as follows;

$$O_n = SIG_{\overline{y_n}}(ENG_{v_0}(o_n, r_n), h_n, y_{n+1})$$

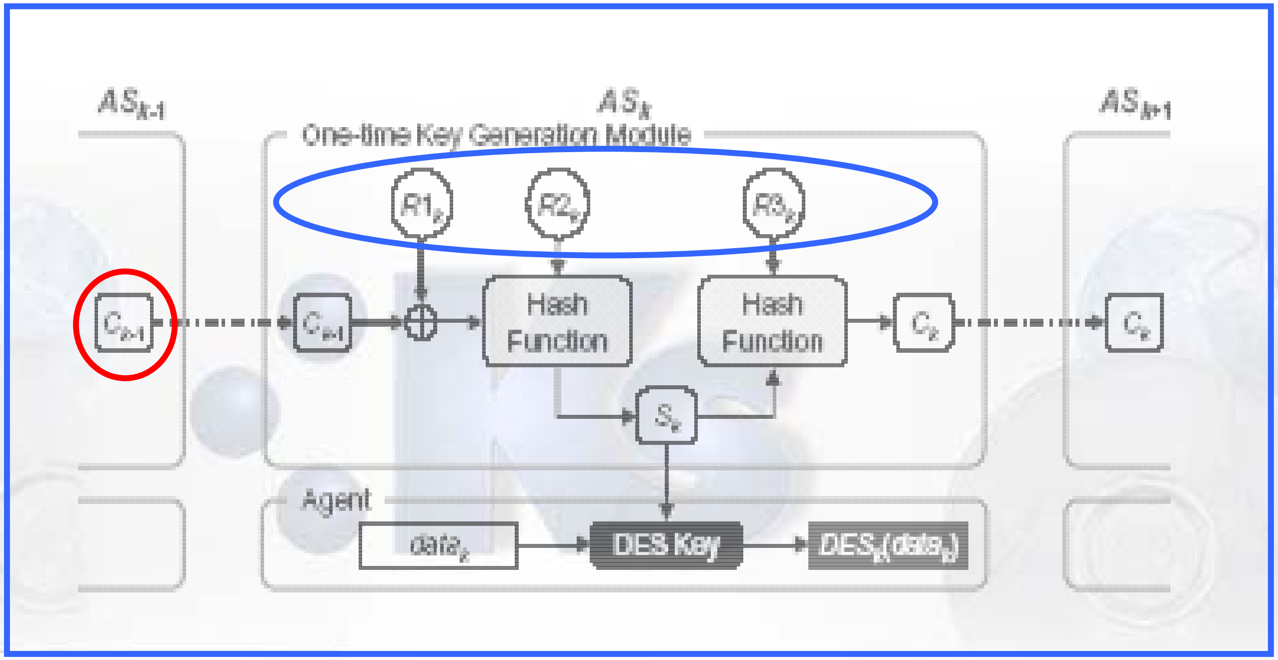
$$h_n = H(O_{n-1}, S_{n+1})$$

1. One time key pair generation problem



Related Work

Protocol OKGS [PARK01]



Solutions

Protocol T1

Scenario "*Free Roaming Mobile Agent*"

Assumption:

1. There are 3 entites in this protocol
2. There is no PKI in this scenario.
3. Every visited host knows public key of originator and ***TTP***
4. ***TTP*** is always online.
5. Both of originator and ***TTP*** know each other public keys.

Protocol T1

Briefly description of the protocol T1:

- S_i obtains the one time private key $\overline{\mu_i}$ for signing its offer o_i under the constraint that only S_{i+1} will be its successor.
- S_{i+1} receives the one time public key μ_i for verification of the validity of S_i's signature on its offer.
- ***TTP*** issues the one time public/private key pair and maintains the key list.

Protocol T1 at S_0

1. Mutual Authentication:

It is is adapted from the ISO/IEC 9798-3 three pass mutual authentication.

$$S_0 \rightarrow TTP : ENC_{\upsilon_{TTP}}(r_0, \upsilon_0, S_0)$$

$$TTP \rightarrow S_0 : ENC_{\upsilon_0}(SiG_{\overline{\upsilon_{TTP}}}(r_0, r_{TTP}, S_o, TTP))$$

$$S_0 \rightarrow TTP : ENC_{\upsilon_{TTP}}(SIG_{\overline{\upsilon_0}}(r_{TTP}, r_o, TTP))$$

Figure 1. Mutual Authentication Protocol

Protocol T1 at S_0

2. Key transportation protocol and Key list

Session 1:
between S_0 and TTP

$$S_o \rightarrow TTP : ENC_{\upsilon_{TTP}}(S_0, S_1, T_{S_0})$$

$$TTP \rightarrow S_0 : ENC_{\upsilon_0}(SIG_{\overline{\upsilon_{TTP}}}(S_0, S_1, \overline{\mu_0}, T_{S_0}, T_{TTP_{0,1}}))$$

$$S_o \rightarrow TTP : ENC_{\upsilon_{TTP}}(SIG_{\overline{\mu_0}}(S_0, S_1, T_{S_0}, T_{TTP_{0,1}}))$$

Session 2:
between S_1 and TTP

$$TTP \rightarrow S_1 : SIG_{\overline{\upsilon_{TTP}}}(r_{TTP}, TTP)$$

$$S_1 \rightarrow TTP : ENC_{\upsilon_{TTP}}(SIG_{\overline{\upsilon_1}}(S_1, TTP, r_1, r_{TTP}), \upsilon_1)$$

$$TTP \rightarrow S_1 : ENC_{\upsilon_1}(SIG_{\overline{\upsilon_{TTP}}}(S_1, TTP, r_1, T_{TTP_{1,0}}, \mu_0, S_0))$$

$$S_1 \rightarrow TTP : ENC_{\upsilon_{TTP}}(ENC_{\mu_0}(S_1, TTP, r_1, T_{TTP_{1,0}}, S_0))$$

Figure 2. Key Transportation Protocol

Protocol T1 at S_0

3. Key List generated by TTP

Signer	Next host	Time of issue	Key pairs
S_0, υ_o	S_1, υ_1	$T_{TTP_{1,0}}, T_{TTP_{0,1}}$	$(\mu_0, \overline{\mu_0})$

Figure 3. Key List

4. Offer Encapsulation

$$O_0 = SIG_{\overline{\mu_o}}(ENC_{\upsilon_0}(o_0, r_0), h_o)$$

$$h_0 = H(r_o, S_1)$$

Figure 4. Encapsulated Offer

Protocol T1 at S_0

5. List of Visited Host

$$\Psi_o = ENC_{v_0}(SiG_{\overline{\mu_0}}(S_0, S_1))$$

Figure 5. List of visited host

6. Agent Transmission

$$S_0 \rightarrow S_1 : SiG_{\overline{v_0}}(\Pi, T_{S_0}), O_0, \Psi_0$$

Figure 6. Agent Transmission

Protocol T1 at S_1

- ***At host S_1***

  1. Verification of agent's code and encapsulated offer O_0.

$SiG_{\bar{v}_0}(\Pi, T_{S_0})$ → By using the originator's public key.

O_0 → By using one time public key μ_0.

Protocol T1 at S_1

1. One time private key retreiving

$$S_1 \rightarrow TTP : ENC_{\upsilon_{TTP}}(S_1, S_2, T_{S_1})$$

$$TTP \rightarrow S_1 : ENC_{\upsilon_1}(SIG_{\overline{\upsilon_{TTP}}}(S_1, S_2, \overline{\mu_1}, T_{S_1}, T_{TTP_{1,2}}))$$

$$S_1 \rightarrow TTP : ENC_{\upsilon_{TTP}}(SIG_{\overline{\mu_1}}(S_1, S_2, T_{S_1}, T_{TTP_{1,2}}))$$

$$TTP \rightarrow S_2 : SIG_{\overline{\upsilon_{TTP}}}(r_{TTP}, TTP)$$

$$S_2 \rightarrow TTP : ENC_{\upsilon_{TTP}}(SIG_{\overline{\upsilon_2}}(S_2, TTP, r_2, r_{TTP}), \upsilon_2)$$

$$TTP \rightarrow S_2 : ENC_{\upsilon_2}(SIG_{\overline{\upsilon_{TTP}}}(S_2, TTP, r_2, T_{TTP_{2,1}}, \mu_1, S_1))$$

$$S_2 \rightarrow TTP : ENC_{\upsilon_{TTP}}(ENC_{\mu_1}(S_2, TTP, r_2, T_{TTP_{2,1}}, S_1))$$

Figure 7. Key Retreiving

Protocol T1 at S_1

2. *Offer Encapsulation*

$$O_1 = SIG_{\overline{\mu_1}}(ENG_{v_0}(o_0, r_0), h_1, S_0, \mu_0)$$

$$h_1 = H(O_0, S_2)$$

Figure 8. Encapsulated Offer

3. *List of Visited host*

$$\Psi_1 = ENC_{v_0}(SiG_{\overline{\mu_1}}(S_0, S_1, S_2), \Psi_0)$$

Figure 9. List of visited host

Protocol T1 at S_1

4. Key List Update at TTP

Signer	Next host (Verifier)	Time of issue	Key pairs
S_0, v_o	S_1, υ_1	$T_{TTP_{1,0}}, T_{TTP_{0,1}}$	$(\mu_0, \overline{\mu_0})$
S_1	S_2, υ_2	$T_{TTP_{2,1}}, T_{TTP_{1,2}}$	$(\mu_1, \overline{\mu_1})$

Figure 10. Key List Update

5. Agent Transmission

$$S_{1} \rightarrow S_2 : SiG_{\overline{\upsilon_0}}(\Pi, T_{S_0}), \{O_0, O_1\}, \Psi_1$$

Figure 11. Agent Transmission

Protocol T1 (verification of the chain)

Assume: The mobile agent arrives at host S_5. S_5 receives $\{O_o, O_1, O_2, O_3, O_4\}$. The host performs as follows

$Ver(O_4, \mu_4)$	Recovers	h_4, μ_3, S_3	Checks	$h_4 \overset{?}{=} H(O_3, S_5)$
$Ver(O_3, \mu_3)$	Recovers	h_3, μ_2, S_2	Checks	$h_3 \overset{?}{=} H(O_2, S_4)$
$Ver(O_2, \mu_2)$	Recovers	h_2, μ_1, S_1	Checks	$h_2 \overset{?}{=} H(O_1, S_3)$
$Ver(O_1, \mu_1)$	Recovers	h_1, μ_0, S_0	Checks	$h_1 \overset{?}{=} H(O_0, S_2)$
$Ver(O_0, \mu_0)$	Recovers	h_0		

Protocol T1 at S_i

- ***Encapsulated offer:***

$$O_i = SIG_{\bar{\mu}_i}(ENG_{v_0}(o_i, r_i), h_i, S_{i-1}, \mu_{i-1})$$

$$h_i = H(O_{i-1}, S_{i+1})$$

- ***The list of visited host:***

$$\Psi_i = ENC_{v_0}(SiG_{\bar{\mu}_i}(S_{i-1}, S_i, S_{i+1}), \Psi_{i-1})$$

- ***Agent transmission:***

$$S_i \rightarrow S_{i+1} : SiG_{\bar{v}_0}(\Pi, T_{S_0}), \{O_0, O_1, \ldots, O_i\}, \Psi_i$$

Protocol T2

Scenario:

The protocol T2 is focused mainly on providing flexibility when dealing with a more realistic situation. For example, ***TTP*** is inactive when host requires for the key for signing an offer.

" *Each host will be granted a temporary authority to generate the one time key pair on ist own, only when the absence of TTP has been confirmed by its successor.*"

Protocol T2

Assumptions:

1. There is no shared key between S_i and S_{i+1}.
2. They have no knowledge of each other's public keys.

Protocol description S_i should do as follows:

- ***Offline status of TTP generation*** (OST_{S_i})

Protocol T2

- ***Key agreement protocol and Mutual authentication***

Step 1. Key agreement protocol based on Diffie-Hellman

$$S_i \rightarrow S_{i+1} : S_i , LIST , r_i , t_{S_i}$$

$$S_{i+1} \rightarrow S_i : S_{i+1} , t_{S_{i+1}} , r_{i+1}$$

Figure 12. Key agreement protocol

$Z_{S_i S_{i+1}}$ at $S_i = t_{S_{i+1}}^{\alpha_{S_0}}$ at $S_{i+1} = t_{S_i}^{\alpha_{S_{i+1}}}$

$LIST = (p, q, g, G, key\ derivation\ function)$

$$K_{i,i+1} = MAC_{r_i, r_{i+1}}(Z_{S_i S_{i+1}})$$

Protocol T2

- ***Mutual authentication***

At S_i, it generates the message which contians its signature on MAC and Offline status of TTP

$$S_i \rightarrow S_{i+1} : SIG_{\bar{v}_i}(MAC_{K_{i,i+1}}(t_{S_i}, t_{S_{i+1}}, r_i, r_{i+1}, List, S_i), OST_{S_i}), ENC_{K_{i,i+1}}(v_i)$$

Figure 13. Mutual Authentication performed by S_i

Protocol T2

At S_{i+1}, it generates the message which contians its signature on MAC and acknowledgement of TTP'offline status.

$$S_{i+1} \rightarrow S_i : SIG_{\overline{v_{i+1}}}(MAC_{K_{i,i+1}}(t_{S_i}, t_{S_{i+1}}, r_i, r_{i+1}, List, S_{i+1}), ack^T_{S_{i+1}}), ENC_{K_{i,i+1}}(v_{i+1})$$

Figure 14. Mutual Authentication performed by S_{i+1}

$ack^T_{S_{i+1}}$ ***= TTP is inactive***

$ack^F_{S_{i+1}}$ ***= TTP is active***

Protocol T2

- *Offer Encapsulation*

$$O_i = SIG_{\bar{\sigma}_i}(ENC_{\upsilon_0}(o_i, r_i, ack^T_{S_{i+1}}, OST_{S_i}), h_i, S_{i-1}, \mu_{i-1}), ENC_{K_{i,i+1}}(ENC_{v_{i+1}}(\sigma_i))$$

$$h_i = H(O_{i-1}, S_{i+1})$$

- *The list of visited host*

$$\Psi_i = ENC_{\upsilon_0}(SiG_{\bar{\sigma}_i}(S_{i-1}, S_i, S_{i+1}), \Psi_{i-1})$$

- *Agent Transmission*

$$S_i \rightarrow S_{i+1} : SiG_{\bar{\upsilon}_0}(\Pi, T_{S_0}), \{O_0, O_1, ..., O_i\}, \Psi_i$$

Security Analysis

1. *Data Confidentiality* ☑
2. *Non-Repudiability* ☑
3. *Forward Privacy*
4. *Strong Forward Integrity* ☑
5. *Insert Resilience* ☑
6. *Truncation Resilience* ☑

Strength and Vulnerability of T1

- *Strenghts*
  1. *One time private key remains a secret between the host and the TTP.*
  2. *The collusion attack can be detected and prevented by using the key list and the list of visited host.*
  3. *Authentication procedure and key transportation scheme are secure againt the impersonation attack.*
- *Weaknesses*
  1. *The protocol introduces a high number of challenge response activities which lead to intervention of communication between the host and the TTP.*
  2. *During the signing period, if there is an absence of TTP then the protocol cannot work.*

Strength and Vulnerability of T2

- *Strengths*
  1. *The execution of mobile agent can be proceeded during the absence of the TTP.*
  2. *This protocol provides more flexibility than T1.*

- *Weaknesses*
  1. *The protocol cannot efficiently defend against truncation attack.*
  2. *The cost of computation is high due to Diffie-Hellman key exchange.*

Improvement of the protocol

- *Provide the system with more than one TTP which can back up each other.*
- *Provide a set of legitimate hosts to be visited in case the TTP is unreachable.*

Conclusion and Future work

- *Conclusion*

1. *The protocol T1 can detect and prevent collusion attack.*
2. *The protocol T2 cannot efficiently defend collusion attack but provide more flexibility than T1.*
3. *We use the combination of T1 and T2 increases the ability to detect and prevent collusion attack.*

- *Future work*

1. *Reduce a number of challenge responses in the protocol T1*
2. *Give the concrete solution of how to generate the evidence and the acknowledgement of TTP's inactiveness*
3. *Add on the function of updated result.*

References

[1]. Yee, B. S.: A Sanctuary for Mobile Agents. Secure Internet Programming. *Lecture Notes in Computer Science,* Vol. 1603. Springer-Verlag, Berlin Heidelberg (1999) 261–273

[2]. Karjoth, G., Asokan, N., G¨ulc¨u, C.: Protecting the Computation Results of Free-Roaming Agents. In: Rothermel, K., Hohl, F.. (eds.): Proceedings of the 2nd International Workshop on Mobile Agents (MA '98). *Lecture Notes in ComputerScience,* Vol. 1477. Springer-Verlag, Berlin Heidelberg New York (1998) 195–207

[3]. Cheng, J. S. L., Wei, V. K.: Defenses against the Truncation of Computation Results of Free-Roaming Agents. In: Deng, R. H., Qing, S., Bao, F., Zhou, J.(ed.): *Information and Communications Security, 4th International Conference, ICICS2002, Singapore. Lecture Notes in Computer Science,* Vol. 2513. Springer-Verlag, Berlin Heidelberg (2002) 1–12

[4]. Ming Yao, Ernest Foo, Kun Peng, and Ed Dawson.: An Improved Forward Integrity Protocol for Mobile Agents. *Lecture Notes in Computer Science*, Vol. 2908. Springer-Verlag, Berlin Heidelberg (2004) 272–285

[5]. Maggi, P., Sisto, R.: A Configurable Mobile Agent Data Protection Protocol *Proceedings of the 2nd International Conference on Autonomous Agents and Multi agent Systems (AAMAS'03)*, Melbourne, Australia. ACM Press, New York, USA (2003) 851–858

[6]. Menezes, A., Oorschot, P. van, Vanstone, S.: Handbook of Applied Cryptography. CRC Press Inc. (1996)

[7]. Roth, V.: Programming Satan's agents. *In: Fischer, K., Hutter, D.. (eds.): Proceedings of 1st International Workshop on Secure Mobile Multi-Agent Systems (SEMAS 2001)*. Electronic Notes in Theoretical Computer Science, Vol. 63. ElsevierScience Publishers (2002).

[8]. Roth, V.: On the Robustness of some Cryptographic Protocols for Mobile AgentProtection. Proceedings Mobile Agents 2001. *Lecture Notes in Computer Science,*Vol. 2240. Springer-Verlag, Berlin Heidelberg (2001) 1–14

References

[9]. Roth, V.: Empowering Mobile Software Agents. *Proceedings 6th IEEE Mobile Agents Conference. Lecture Notes in Computer Science*, Vol. 2535. Springer-Verlag, Berlin Heidelberg (2002) 47–63

[10]. N. M. Karnik and A. R. Tripathi. Security in the Ajanta Mobile Agent System. *Technical Report TR-5-99*, University of Minnesota, Minneapolis, MN 55455, U. S. A. , May 1999.

[11]. A. Corradi, R. Montanari, and C. Stefanelli. Mobile agents Protection in the Internet Environment. In *The 23rd Annual International Computer Software and Applications Conference (COMPSAC '99)*,pages pp. 80–85, 1999.

[12]. Jong-Youl Park, Dong-Ik Lee and Hyung-Hyo Lee. Data Protection in Mobile Agents; one-time key based approach. *IEEE ISADS 01*, pp.411-418, March 2001

[13]. Colin Boyd and Anish Mathria. Protocols for Authentication and Key Establishment. *Springer-Verlag* ISBN 3-540-43107-1

[14]. ISO. I*nformation technology-Security techniques- Entity authentication mechanisms-part 3: Entity authentication Using a Public key Algorithm ISO/IEC 9798-3.* 2nd Edition , 1998. International standard.

[15]. Simon Blake-Wilson and Alfred Menezes. Authenticated Diffie-Hellman Key Agreement Protocols. *SAC'98 LNCS 1556*, pp. 339-361. 1999

[16]. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE transaction on Information Theory*, November 1976.

[17]. Whitfield Diffie, Paul C. van Oorschot and Michael J. Wiener. Authentication and Authenticated Key exchange. *Designs, Codes and Cryptography*, March 1992

[18]. H. Orman. The OAKLEY Key Determination Protocol. *The Internet Society, November 1998, RFC 2412*

[19]. Hugo Krawczyk. SKEME: A versatile secure key exchange mechanism for Internet. In *Symposium on Network and Distributed System Security*, pages 114-127. IEEE Computer Security Press, 1996