

Protecting Mobile Agents with Data Encapsulation and Execution Tracing: The Protocol

Anna Suen
suen@cs.fsu.edu
March 24, 2003

Preview

- Layout of protocol steps
- My contribution

Step 1: The Originator

- compute dummy encapsulated offer
- set hash: $H_0 = g$
- dispatch MA

$P_0 \rightarrow P_1:$

$P_0, P_0, ENC_{P_1}(C, S, \{O_0\}, H_0), SIG_{P_0}(P_1, t_{P_0}),$
 $SIG_{P_0}(h(C, S), I)$

3

Step 2: The Intermediate Platforms

- check if I am originator
 - if yes, proceed with processing results
- check if I am intended recipient
 - if not, forward to correct recipient
- check timestamp
 - if expired, halt execution and send notice to originator

$P_1 \rightarrow P_2:$

$P_1, P_0, ENC_{P_2}(C, S, \{O_0, O_1\}, H_1), SIG_{P_1}(P_2, t_{P_1}),$
 $SIG_{P_0}(h(C, S), I)$

4

Step 2: continued

- check integrity of code and state
 - if invalid, halt execution and send notice to originator
- check validity of encapsulated offers against set hash?

$P_1 \rightarrow P_2:$

$P_1, P_0, ENC_{P_2}(C, S, \{O_0, O_1\}, H_1), SIG_{P_1}(P_2, t_{P_1}),$
 $SIG_{P_0}(h(C, S), I)$

5

Step 2: continued

- execute code to produce offer
- compute encapsulated offer & set hash
- log results in execution trace file
- append encapsulated offer
- send mobile agent to next recipient

$P_2 \rightarrow P_3:$

$P_2, P_0, ENC_{P_3}(C, S, \{O_0, O_1, O_2\}, H_2), SIG_{P_2}(P_3, t_{P_2}),$
 $SIG_{P_0}(h(C, S), I)$

6

Step 3: Originator Receives Results

- check timestamp, integrity of code & state against original copy, validity of chain/encapsulated offers
 - if any invalid, last platform was malicious
- if still suspect tampering, retrieve execution trace from respective platform
 - if not match, something's wrong
 - just discard offer?
- if everything checks out, can be confident results are valid

7

Notice Message

- flag indicating message is a notice
- type of notice
 - expired timestamp
 - invalid code
 - invalid state
 - invalid/broken chain
- entire message received, signed

notice message	type of notice	$SIG_{P_3}(P_2, P_0, ENC_{P_3}(C, S, \{O_0, O_1, O_2\}, H_2), SIG_{P_2}(P_3, t_{P_2}), SIG_{P_0}(h(C, S), I))$
----------------	----------------	----------------------------------------------------------------------------------------------------------------

8

My Contribution

$$B \xrightarrow{msg} D : B, A, D_p(C, S^1), B_s(H(S^1), D), A_s(H(C), H(S^0), B, A', t_A, i_A)$$

- Execution Tracing

- eliminated TTP
 - single point of failure
 - dependency
- improve efficiency
 - eliminated extraneous info from last field
 - standardize messages sent from one platform to the next
- timestamp

$$B \rightarrow D : B, A, D_p(C, S^1), B_s(D, H(S^1), t_B), A_s(H(C), A', i_A) \quad 9$$

My Contribution

$$P_i \rightarrow P_{i+1} :$$

$$P_i, P_0, ENC_{P_{i+1}}(C, S), SIG_{P_i}(P_{i+1}, h(S), t_{P_i}), SIG_{P_0}(h(C), P'_0, I_{P_0})$$

- Data Encapsulation

- Vigna's protocol: easy to replace state
- data integrity

- Set Hashing

- verify integrity of data encapsulation
- may be discarded if no updating

- Chaining?

$$P_i \rightarrow P_{i+1} :$$

$$P_i, P_0, ENC_{P_{i+1}}(C, S, \{O_0, \dots, O_i\}, H_i), SIG_{P_i}(P_{i+1}, t_{P_i}), SIG_{P_0}(h(C, S), D)$$

