

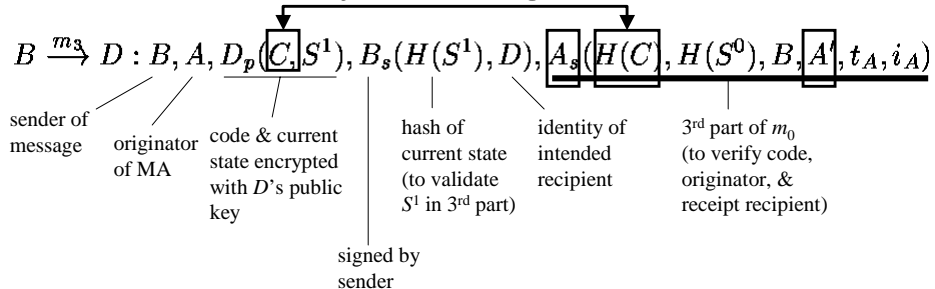
Using Execution Tracing to Detect Tampering of Mobile Agents

version 2.0

Anna Suen

February 24, 2003

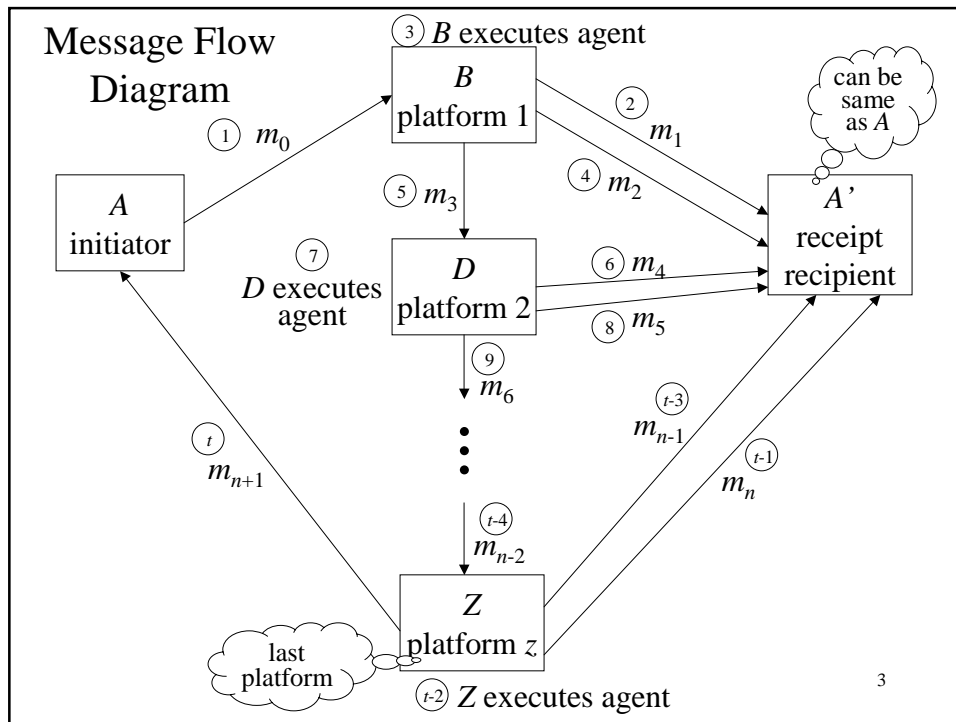
Inefficiency in Original Protocol



From the part of the message signed by A , D can verify that

- the code contained in the third part is correct
- the agent was initially dispatched by A
- A meant to use A' as recipient of the receipt messages

So, it seems like the non-circled parts of the 5th part are useless and can be excluded.



Notation

A_p	public key of A
A_s	secret key of A
$A_x(m)$	message m encrypted with key A_x
$H(m)$	one-way hash of message m
C	the code
S^i	the state at point i
T_C^i	the trace of code C at state S^i

Assumptions

- trusted third party
- PKI
 - for encrypting and signing
- tamper-proof, non-repudiable execution trace file
- code is static

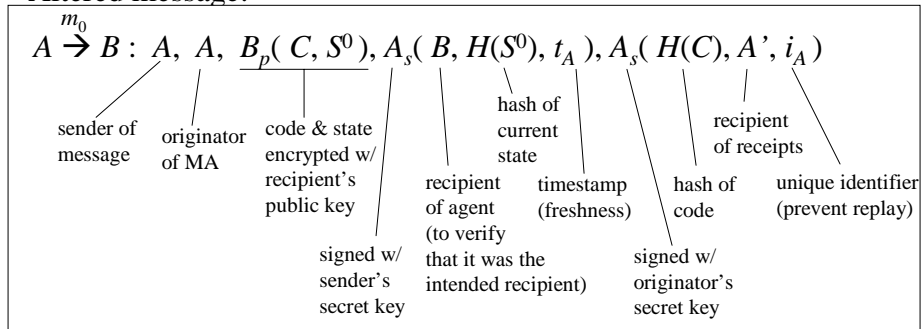
6

1. A sends B the mobile agent.

- Original message:

$$A \xrightarrow{m_0} B : A, B_p(C, S^0), A_s(H(C), H(S^0), B, A', t_A, i_A)$$

- Altered message:



Changes:

- split the 3rd part of original message – this part is used a lot in later messages and not all the contents are needed
- include the originator identity to make the protocol more uniform (so all messages for sending MA to another platform is the same)
- moved timestamp to 4th part (signed by sender)
 - each sender will need to include a new timestamp

7

2. *B* verifies:

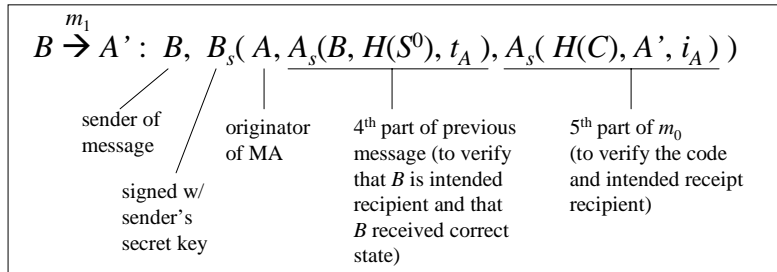
- integrity of *C* by computing $H(C)$ and comparing it with that in the 5th part of the message
- that the message was intended for itself

If everything checks out, *B* sends the designated recipient *A'* a receipt indicating that he received the message.

- Original message:

$$B \xrightarrow{m_1} A' : B, B_s(A, A_s(H(C), H(S^0), B, A', t_A, i_A))$$

- Altered message:



A' can verify that *B* received MA from *A* at time t_A and that *A'* was the intended receipt message recipient. 8

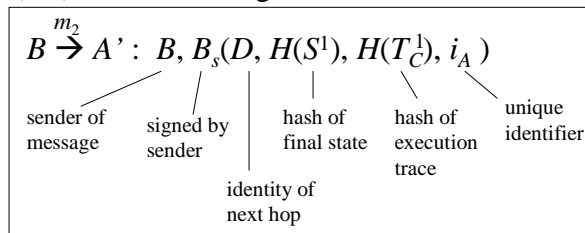
3. *B* executes the agent until it needs to move to the next platform *D*.

4. *B* sends *A'* a hash of the results and the trace.

- Original message:

$$B \xrightarrow{m_2} A' : B, B_s(D, H(S^1), H(T_C^1), i_A)$$

- (Un)Altered message:



- No changes to this message.

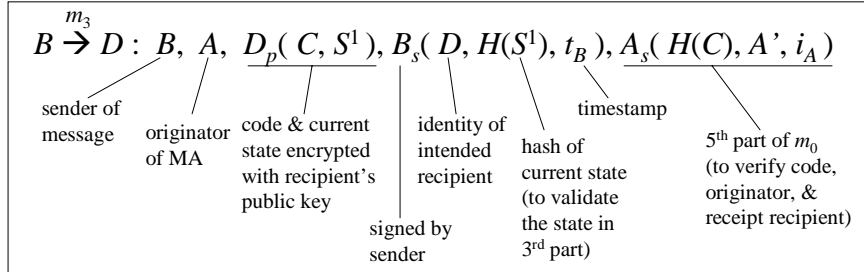
9

5. B sends the MA to D .

- Original message:

$B \xrightarrow{m_3} D : B, A, D_p(C, S^1), B_s(H(S^1), D), A_s(H(C), H(S^0), B, A', t_A, i_A))$

- Altered message:



Changes:

- B includes a new timestamp for freshness.

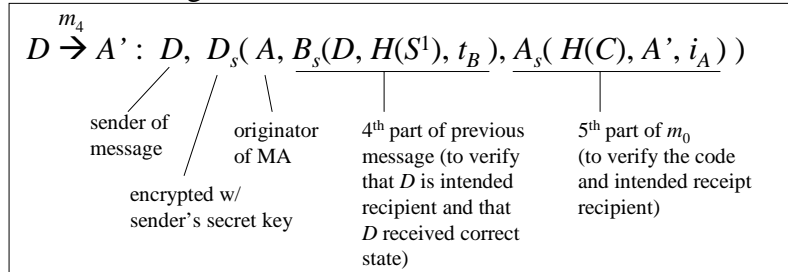
10

6. After verifying the code and recipient, D sends A' a receipt, indicating that he received the agent.

- Original message:

$D \xrightarrow{m_4} A' : D, D_s(B_s(H(S^1), D), A_s(H(C), H(S^0), B, A', t_A, i_A))$

- Altered message:



A' makes sure that:

- state matches that from m_2 – ensures that state wasn't altered in transit
- code matches that from m_1

11

7. D executes the agent.
 8. D sends a receipt of the results to A' .
 9. D sends the agent to the next platform.
- Repeat until the agent decides to terminate.

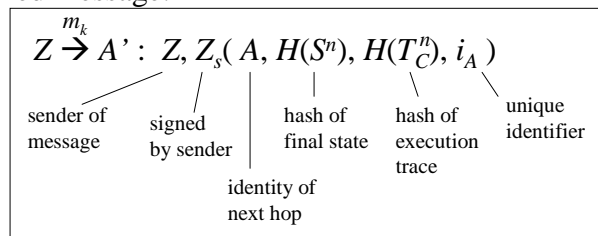
12

$t-1$. The final site Z sends A' a hash of the final results.

- Original message:

$$Z \xrightarrow{m_k} A' : Z, Z_s(H(S^n), H(T_C^n), i_A)$$

- Altered message:



- Identity of next hop is included.
- This format is identical to all other result receipt messages sent to A' .

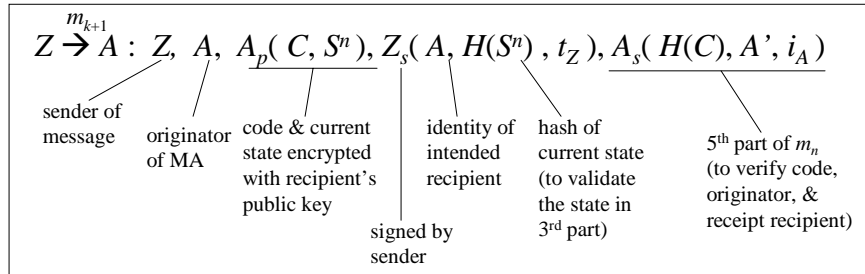
13

t. Z sends the originator A the final results.

- Original message:

$$Z \xrightarrow{m_{n+1}} A : Z, Z_s(A_p(S^n), i_A)$$

- Altered message:



- This message is in the same format as other messages for sending the MA from one platform to the next platform.
- With all the information, originator can verify that code has not been altered, that state wasn't altered in transit, etc.

14

If originator suspects tampering...

- ask A' to provide all receipts
- ask each platform to produce the corresponding execution traces
- simulate the execution to verify the results

15

Protocol Summary

0, ..., j, ..., k+1 messages

0, ..., i, ..., n states

$A \xrightarrow{m_0} B : A, A, B_p(C, S^0), A_s(B, H(S^0), t_A), A_s(H(C), A', i_A)$

$B \xrightarrow{m_1} A' : B, B_s(A, A_s(B, H(S^0), t_A), A_s(H(C), A', i_A))$

B executes agent

$B \xrightarrow{m_2} A' : B, B_s(D, H(S^1), H(T_C^1), i_A)$

⋮

$X \xrightarrow{m_j} Y : X, X, Y_p(C, S^i), X_s(Y, H(S^i), t_X), X_s(H(C), X', i_X)$

$Y \xrightarrow{m_{j+1}} X' : Y, Y_s(X, X_s(Y, H(S^i), t_X), X_s(H(C), X', i_X))$

Y executes agent

$Y \xrightarrow{m_{j+2}} X' : Y, Y_s(D, H(S^{i+1}), H(T_C^{i+1}), i_X)$

⋮

$Z \xrightarrow{m_{k+1}} A : Z, A, A_p(C, S^n), Z_s(A, H(S^n), t_Z), A_s(H(C), A', i_A)$ 16

Currently working on...

- Chaining/encapsulation
- Eliminate trusted third party?
- Evaluate the security

Reference:

Giovanni Vigna. "Protecting Mobile Agents through Tracing." Proceedings of the 3rd ECOOP Workshop on Mobile Object Systems. Jyväskylä, Finland. June 1997.