Mobile Agent Security

Quote of the Day

"*Always acknowledge a fault. This will throw those in authority off their guard and give you an opportunity to commit more.*"

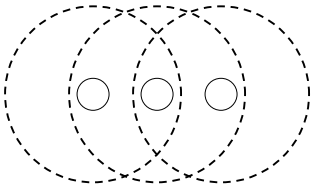
-Mark Twain

Topics of Discussion

- Ad hoc networks
- Application of mobile agents
- Protecting agents

Mobile Ad Hoc NETworks

- MANET – “infrastructure-less”
- Transmission range
- Group coordination – routing



Issues

- Group connectivity
- Routing
- Secure communication

Possible Solution

Mobile Agents

- autonomous
- work on someone's behalf
- well-defined mission
- self-propagating

Trade-offs with security

Taxonomy of Mobile Agent Security

- Protecting agent platform
- Protecting agent

NIST Special Publication 800-19 (August 1999)

Wayne Jansen & Tom Karygiannis

http://csrc.nist.gov/mobileagents/publication/sp800-19.pdf

Protecting Agents

"Cryptographic Traces for Mobile Agents"
Giovanni Vigna
Mobile Agents and Security
Lecture Notes in Computer Science
Springer-Verlag, June 1998

Reasons for Agent Protection

- Mobile code
- Agent decides when & where to go
- Untrusted computing base
- Liability issues

Agent Model

- Code segment p – static
- Execution state S^i – dynamic

Execution environment ***must*** access both

Potential Consequences

- Data & code disclosure
- Data & code tampering
- Incorrect execution

Types of Agent Protection

- Prevention – “hard” to modify
- Detection – flag illegal modification
  mostly for static use (i.e. code)

Traces

- Detection mechanism
- Data collected during execution
- “Post-mortem” analysis of code, execution state, and execution flow

Assumptions (for Alec to refute)

- PKI by means of certificates
- Interpreter implementation certified correct (i.e. “won’t say one thing and do another”)
- Single-threaded agents
- No memory sharing between agents

Notation

X_p : public key of X

X_s : secret key of X

i_X : message identifier

t_X : timestamp

$H(x)$: hash of x

$K_A(x)$: encryption under secret key K_A

Trace T^p of program p with

$$T^p = \langle n,s \rangle$$

for statement identifier n and statement signature s

Remote Execution Protocol

1. $A \rightarrow B : A_S(A,B,i_A,t_A,K_A(p),TTP)$
2. $B \rightarrow A : B_S(B,A,i_A,H(m_1), M)$
3. $A \rightarrow B : A_S(A,B,i_A,B_p(K_A))$
4. $B \rightarrow A : B_S(B,A,i_A,H(m_3))$
5. $B \rightarrow A : B_S(B,A,i_A,K_B(S_B),H(T^P{}_B),t_B)$
6. $A \rightarrow B : A_S(A,B,i_A,H(m_5))$
7. $B \rightarrow A : B_S(B,A,i_A,A_p(K_B))$

Validation

- Performed at code owner's will
- Owner requests full T^p from B
  (already possesses $H(T^p)$)
- Owner validates T^p by **re-executing** p
- Statement-by-statement comparison

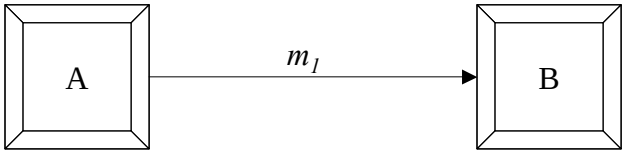
Mobile Agent Protocol

- Extension of remote execution protocol
- Naturally, more than two parties involved
- More interaction between them

Mobile Agent Protocol

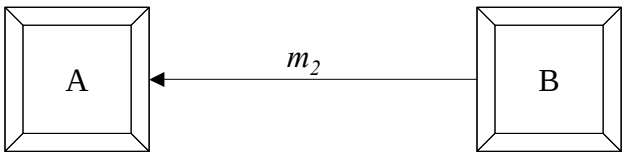
$$m_1 = A_S(A,B,\ K_A(p,S_A),\ A_S\ (A\ ,i_A,t_A,H(p),TTP))$$

$$A_S\ (A\ ,i_A,t_A,H(p),TTP) = agent_A$$



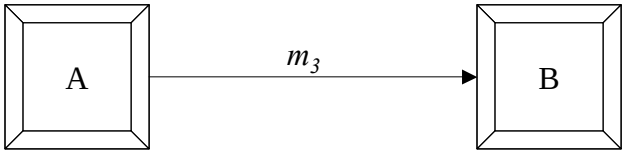
Mobile Agent Protocol

$$m_2 = B_S(B,A,i_A,H(m_1),\ M)$$



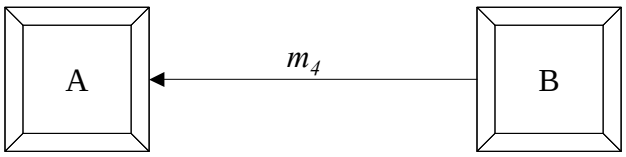
Mobile Agent Protocol

$$m_3 = A_S(A,B,i_A,B_p(K_A))$$



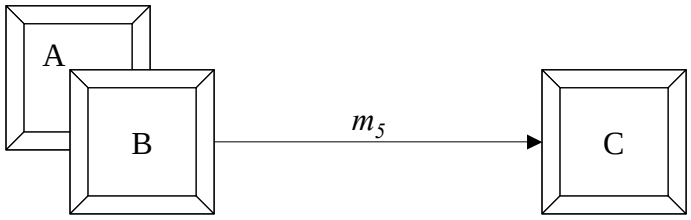
Mobile Agent Protocol

$$m_4 = B_S(B,A,i_A,H(m_3))$$



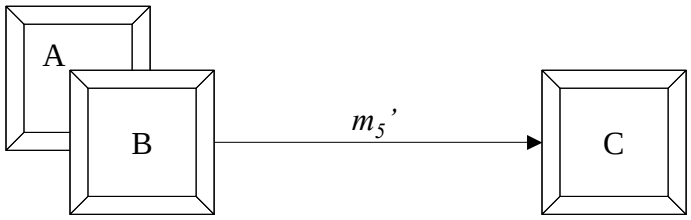
Mobile Agent Protocol

$$m_5 = B_S(B, C, agent_A, H(T^p{}_B), H(S_B), t_B)$$



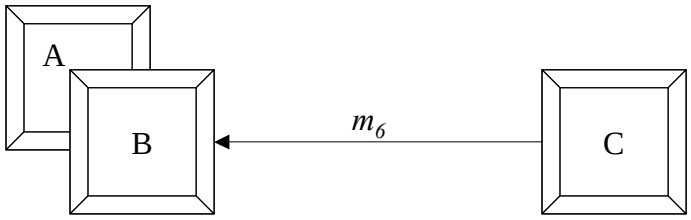
Mobile Agent Protocol

$$m_5' = B_S(K_B(p, S_B), H(m_5))$$



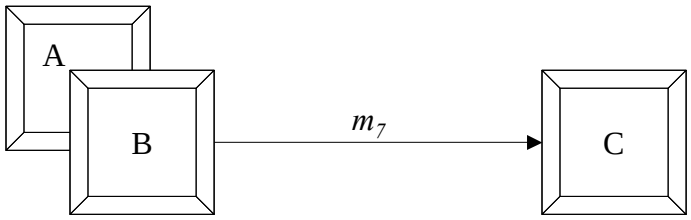
Mobile Agent Protocol

$$m_6 = C_S(C,B,i_A,H(m_5,m_5'),M)$$



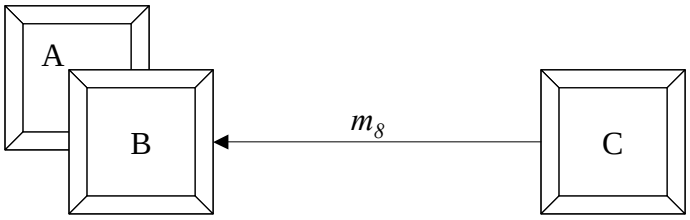
Mobile Agent Protocol

$$m_7 = B_S(B,C,i_A,C_p(K_B))$$

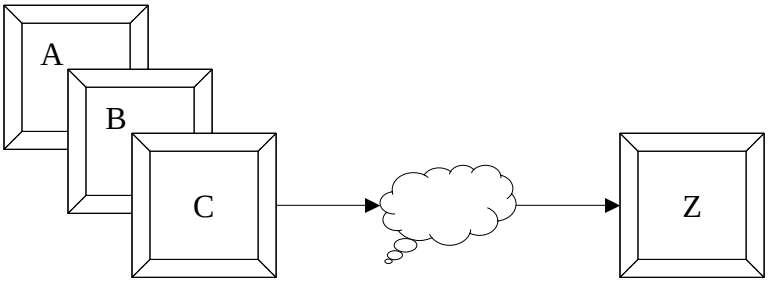


Mobile Agent Protocol

$$m_8 = C_S(C,B,i_A,H(m_7))$$

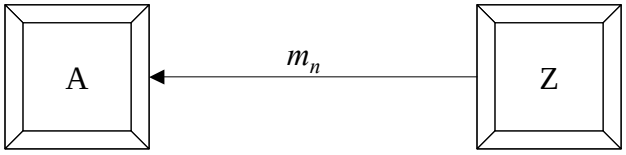


Mobile Agent Protocol



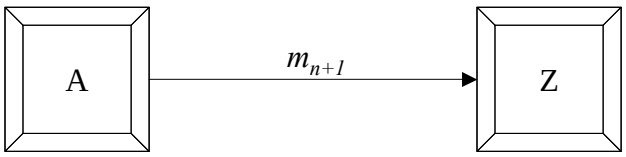
Mobile Agent Protocol

$$m_n = Z_S(Z, A, agent_A, H(T^p{}_Z), K_Z(S_Z), t_Z)$$



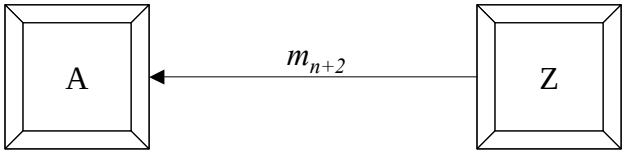
Mobile Agent Protocol

$$M_{n+1} = A_S(A, Z, i_A, H(m_n))$$



Mobile Agent Protocol

$$M_{n+2} = Z_S(Z, A, i_A, Ap(K_Z), H(m_{n+1}))$$



Validation

- Similar to remote execution
- Reference handout

Trace Limitations

- PK crypto & validation EXPENSIVE
- Detection only *after* agent execution
- No means of knowing *a priori* if tampering occurred – must perform validation
- Code & state disclosure