Mobile Object security

Will Harwood

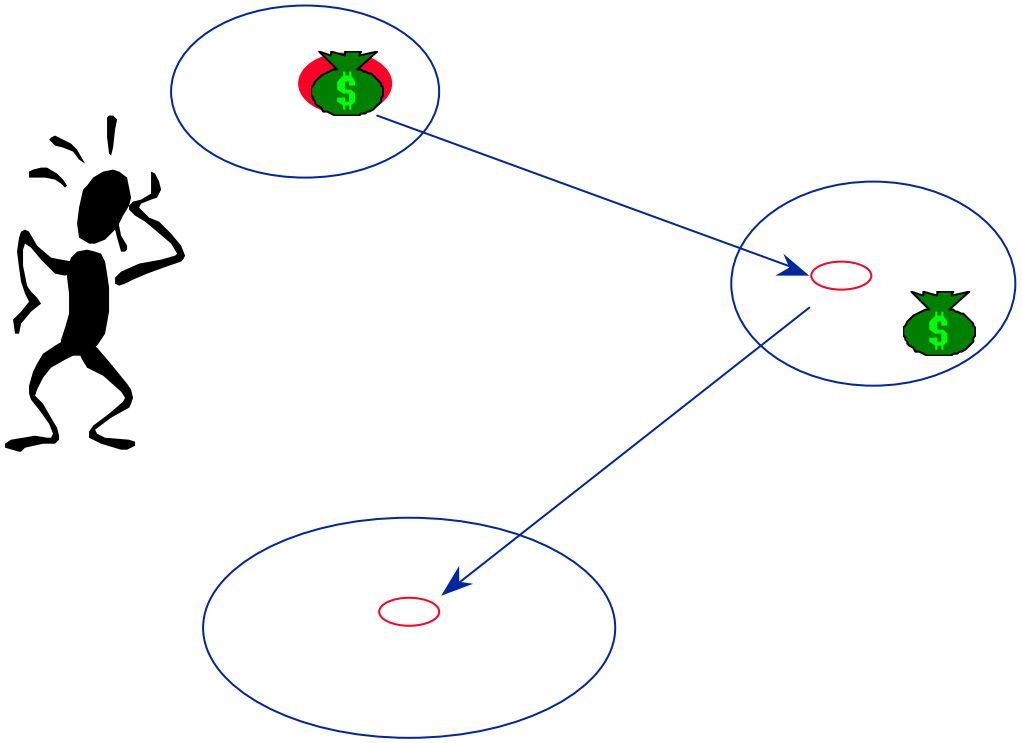
 © 1998 ANSA Consortium

Issues

- Communication Security
  - Good Existing Model c.f. Dave Otways previous presentations
- Mobility Security
  - Mobile objects need to carry secrets
  - Mobile objects act on behalf of users
  - Mobile object code and state may be subverted by hosts

 © 1998 ANSA Consortium

Mobility Security



- Mobile objects need to carry secrets
- Mobile objects act on behalf of users
- Mobile object code and state may be subverted by hosts
- Mobile objects must place some trust in hosts

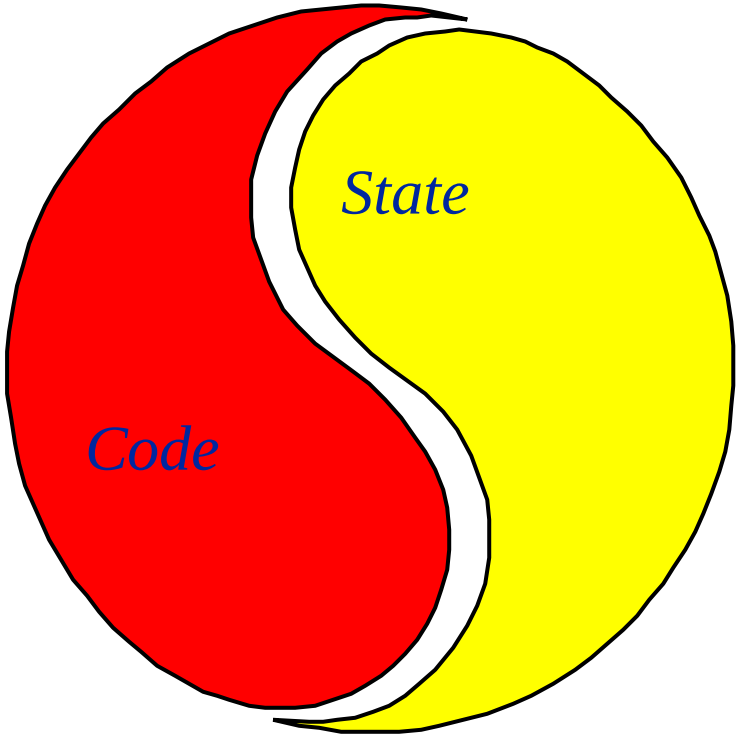
 © 1998 ANSA Consortium

Partial Trust: Basic Ideas

- Mobile objects partially trust hosts
  - trust hosts only with some actions on object
- The nature of the trust is encoded in a security policy for a mobile object
  - A policy states what actions can be performed at which hosts
  - A policy is a part of a mobile objects
  - Changes to policy is an action that can occur at certain hosts

 © 1998 ANSA Consortium

What is an object?



- Abusing an object
  - Use methods that we are not entitled to use
  - Alter an objects code
  - Alter an an object's state without using its methods
  - Read an object's state without using its methods
- Policy needs to address each form of abuse

 © 1998 ANSA Consortium

Approaches to Implementing Policies

- Trust all hosts to respect policy?
  - Unrealistic and limiting assumption
- Create mechanism in mobile objects that enforces adherence to policy?
  - not possible
- Create mechanism that in objects that limits the possibility of policy violation?

 © 1998 ANSA Consortium

Limiting Policy Violation

- Policy is really about:
  - the integrity of the code of an object
  - the integrity of (part of) the state of an object
  - the confidentiality of (part of) the state of an object
- Policy describes which host can read and write which part of the object's state
  - Not all host can read and write all variables of an object

 © 1998 ANSA Consortium

Implementation Mechanisms

- Sign the code of a mobile object so that it cannot be altered unnoticeably
- Sign variable values so that third parties can determine that variables have only been changed at authorised hosts
- Seal variables cryptographically so that their values may only be read at certain hosts

 © 1998 ANSA Consortium

Keys and Key Distribution

- Code signing is performed by “public key” techniques using the mobile objects originator (user or host) key
- State signing is performed by “public key” techniques using either host keys or keys obtained by the host from the object
- Encryption and decryption of variables is performed by either “public” or “secret” key techniques using keys obtained from the object

 © 1998 ANSA Consortium

Key Distribution

- Host support “public key” cryptography
- Keys for variable (values) of an object are distributed in the mobile object to a host encrypted under the public key of the host
  - If a variable may be read at a host the host is given an encrypted read key for the variable
  - If a variable may be written at a host the host is given an encrypted write key for the object
  - If an anonymous integrity check is required the host is given an encrypted signing key for the variable

 © 1998 ANSA Consortium

On Arrival at a New Host

- The host performs a code integrity check
- The host requests the object to perform a state integrity check
- The object hands over any valid keys for the host for variable decryption, encryption and signing. When the object reads or writes protected a variable it interacts with the host to decrypt, encrypt and sign the variables.

 © 1998 ANSA Consortium

Policy Description- Simple Version

- Conceptually an object policy is a (sparse) table of Host Name by Variable Name where the entries are sub sets of :-
  - Read Key - encrypted
  - Write Key - encrypted
  - Private SigningKey or request for host signing - encrypted
  - Public Signing Key - plain

 © 1998 ANSA Consortium

Policy Description- Extended Version

- A variable's integrity may be based on the assumption that other variables have not changed since it was set
  - e.g. X depends on Y,Z
- Implemented by recording a secure hash of X,Y and Z and signing the hash

 © 1998 ANSA Consortium

Implementation: Policy and Variables

- Policies are implemented in the code of an object
- State Integrity Checking is provided as a method of a mobile object that knows about the protected variables
- Variables are objects that support put, get methods and check methods

 © 1998 ANSA Consortium

Scaling

- Cost of policy entry (simplest model) ≤
  - Host name size + variable identifier size + 4 * asymmetric key size
- Easily copes with a small number (100's) of host/variable policies conditions in mobile code
  - Can use more efficient representations for policy entries e.g.
    - P Variable-Name × P (Host-Name × ReadKey × HostKey) × Actions
- Larger numbers can be coped with by mobile policy referring back to "policy servers"

© 1998 ANSA Consortium

Groups of Hosts

- Homogenous groups of hosts (e.g. local networks) can be treated by decrypt on entering network, sign on leaving approach
- Large Non Homogenous groups of hosts can be treated by multiple key approach i.e.
- Hosts have multiple keys, some of these keys are shared by all hosts in a particular group

 © 1998 ANSA Consortium

Costs

- Cost of scheme is better than the worst case of transport between totally trusted hosts on a public network which requires the whole of an object to be encrypted and signed on each transfer between hosts.
- If there is no security requirement there is no cost

 © 1998 ANSA Consortium

Expressing Policy

- Directly by use of implementation classes for policy and SecureObject
- By annotation of state variables of an object e.g.
  - SecureInt y = ...; host1 writes;
  - SecureBool x =...; host1 reads; host2 writes; host3 writes and requires host1 signing of y;
- The annotation is compiled into a policy

 © 1998 ANSA Consortium

- By annotation of methods of an object
  - host1 may invoke method1
  - host2 may invoke method2 provided host1 invoked method1
- The annotations compile into conditions on variables read and written by the methods

 © 1998 ANSA Consortium