

Macro-level Attention to Mobile Agent Security: Introducing the Mobile Agent Secure Hub Infrastructure Concept

Michelangelo Giansiracusa[‡], Selwyn Russell[‡], Andrew Clark[‡], and Volker Roth^{*}

Information Security Research Centre
Queensland University of Technology
Brisbane, Queensland, Australia[‡]

and

Dept. Security Technology
Fraunhofer Institute for Computer Graphics
Darmstadt, Germany^{*}

{mic, selwyn, aclark}@isrc.qut.edu.au, vroth@igd.fhg.de

Abstract. The autonomous capabilities of Internet mobile agents are one of their great attractions, yet leave them at the mercy of ill-intending agent platforms. We have devised an infrastructural strategy that allows mobile agent users to delegate responsibility to a trusted third party for the safe management of mobile agents they deploy onto the Internet. Our infrastructural approach is based on a Mobile Agent Secure Hub (MASH) which is capable of providing a large number of security services for agent users and their deployed Internet mobile agents. For instance, the MASH can gather statistics on the track record of agent platforms in providing safe and reliable execution of agents. These publishable statistics act as a deterrent against maliciously behaving agent platforms, as some agent users would be hesitant to send their agents to platforms with unsound track records.

Keywords: Mobile agent protection, Trusted Third Party, Mobile Agent Secure Hub, macro-level issues, anonymity, accountability, reputation.

1 Introduction

When first promoted a decade ago, the mobile agent paradigm was marketed as the core future of distributed computing and electronic commerce. However, corporate investment in an infrastructure that allows mobile agents to perform their work autonomously in non-trusted domains - particularly the Internet for e-commerce - has been well below expectations. Security concerns, both to mobile agent users and mobile agent platform service providers, have been a major reason for the slow take-up of mobile agents for this intended purpose.

Before mobile agents can be deemed ready for wide utilisation in security-critical applications, mobile agent precautionary measures and applications must be seen to be as secure as client/server applications. However, this goal would seem as elusive as ever.

We have seen the bulk of research in this field concentrating on micro aspects of mobile agent security. By this, we mean research focused principally on protecting one party or the other - or research solely looking at mobile agent integrity (or confidentiality, etc.), and even then only appropriate to certain mobile agent applications.

We feel taking a step 'back' to take two or more steps forward in the long term is the most sound investment for the mobile agent security field one could make. Thus, our research

looks at methods to bring distrusting mobile agent owners and mobile agent platforms to the bargaining table and in turn form a working relationship - the great stumbling block thus far in this field.

There will never be an Internet community of thousands or millions of Internet mobile agent platforms for agents to do meaningful work on without such a higher-level, forward thinking, infrastructural approach to mobile agent security. Just as an economy cannot thrive without due care being shown to micro and macro issues, we believe mobile agent security research (and ultimately Internet mobile agent prospects) cannot prosper without diligent research on both levels.

1.1 Background

We define a mobile agent as: *A software coded abstraction of a number of tasks assigned on behalf of its user. The mobile agent is capable of autonomous migration to networked mobile agent platforms where it performs a subset of its work. The mobile agent's execution state is maintained as it hops between mobile agent platforms in its itinerary. A mobile agent's work at each agent platform provides partial results which are accumulated and analysed in achieving its high-level purpose - that is, the mobile agent's mission goal.*

In their purest form, mobile agents must be capable of performing their assigned tasks autonomously and securely. This means they must perform their mission without the need for premature or intermittent return to their user's home platform to assure their data and logic integrity or to complete a task securely. Hence, to use an example, a "pure" mobile agent would be capable of autonomously buying an airline ticket that it has found to offer the cheapest price satisfying some search criteria, without improper disclosure or use of the mobile agent user's personal and payment details. Within a non-trusted environment like the Internet, the pursuit for autonomous and secure mobile agents remains largely an unsolved problem.

Internet mobile agents face many security threats, many of which are non-trivial to foil. Whilst some countermeasures have been proposed, the underlying framework of the mobile agent paradigm, its intended distributed flexibility, and associated implications raise many more problems than viable safe solutions. For a good appraisal of the security risks and associated issues pertaining to Internet mobile agents we guide the interested reader to [1–5].

The security threats, especially from malicious mobile agent platforms, to mobile agents have been widely reported [6–8]. The associated risks are real and often non-trivial to counter, so much so that they have limited investment in the adoption of mobile agents as a paradigm and platform for global electronic services. While some countermeasures have been proposed and reviewed [1, 3], their lack of robustness is of some concern.

Research to date in countering the malicious host platform problem has focused broadly on either: (i) using trusted hardware [9, 10], (ii) rendering the agents code and data immediately unintelligible (via obfuscation [8, 11], clueless agents [12, 13], or partitioned co-operating agents [11, 14]) to agent platforms, or (iii) utilising trusted mobile agent platforms in the mobile agent's itinerary, if available, for raising confidence in a mobile agent's mission results [15, 16].

The first set of approaches (trusted hardware) suffer from limitations in their practicability to widespread mobile agent platform adoption. Trusted hardware is expensive to acquire and

to certify high security-grade compliant, and potential agent platform service providers would be hesitant to invest in such hardware without higher-level trust mechanisms to encourage mobile agent interaction - thereby offering a credible basis for return on investment.

The strategies which render agent code and data unintelligible are only useful for a certain time period - as all obfuscated code or code distributed over multiple agents can, given sufficient time, be reverse engineered or analysed by agent platforms co-operating maliciously.

Lastly, the set of strategies utilising trusted platforms in the mobile agent's itinerary break down in the case where no mobile agent platforms in the agent's itinerary are reputedly trustworthy - we expect a common scenario for mobile agents geared to work in the Internet. In such a situation, the safety checks must then be performed by the mobile agent user (who may be offline), thus breaking our definition of autonomous work and mission completion solely by the mobile agent remotely.

Returning to our previous hypothetical example of a mobile agent's mission; a mobile agent may be required to purchase the cheapest available around-the-world ticket matching certain criteria - such as a prescribed set of countries and a range list of departure and arrival dates. How can such a mobile agent mission be met and completed both autonomously and securely?

In all likelihood, if the mass adoption of mobile agents were ever to become a commercial reality, the average mobile agent user (launching a mobile agent onto the Internet from possibly a mobile phone) would be a novice to the intricacies of the dangers posed by malicious platforms and other entities. The responsibility must not fall to the mobile agent user for guaranteeing the mobile agent's mission is completed securely.

Trusted third parties (TTPs) can play an effective bridging role in bringing higher security guarantees to both mobile agent users and mobile agent platform service providers in an Internet mobile agent context [17]. As mobile agent platform service providers and mobile agent owners are mutually suspicious of the other, strategically involving a TTP entity into their business engagement can downsize concern for both sets of interested parties. Our approach builds on this observation, and the remainder of the paper spells out in some detail our TTP-based conceptual contribution and work to date, specifically the Mobile Agent Secure Hub Infrastructure (MASHIn).

1.2 Outline of Paper

In Section 2, our TTP modeled Mobile Agent Secure Hub Infrastructure (MASHIn) concept is introduced, and we discuss the major security services offered to Internet mobile agents by MASHs.

We recognise, in Section 3, that the MASHIn concept is not a panacea solution to the malicious host platform problem for Internet mobile agents. Careful agent design, along with careful itinerary selection and management decisions are seen to be important inputs to the delegated MASH protection of Internet mobile agents process. Only with this cohesion can MASHs responsibly offer Internet mobile agents a significant reduction in risks to the aforementioned problem.

Related research work is discussed in Section 4, including how that work differs to our MASHIn work.

In Section 5, we review our contribution and conclusions, and state our focus for future work in this research undertaking.

2 MASHIn Concept

For Internet mobile agent usage to reach anywhere near the growth their original marketing promised, mobile agent users need much stronger guarantees for mobile agent protection. This assurance is needed so mobile agent users can confidently assume the results mobile agents return are not only believable and legitimately correct, but the danger for misuse of their mobile agents is kept to an absolute minimum.

We commence overviews of our MASHIn concept in Section 2.1. In Section 2.2, we list some of the major threats to Internet mobile agents by malicious platforms and our MASHIn countermeasures. Section 2.3 delves further into the security services offered by MASHs in helping to mitigate these threats. The core software modules we foresee in the MASHIn are discussed in Section 2.4, including their purpose and core interdependencies. In Section 2.5, we discuss deployment considerations for the MASHIn.

2.1 Overview

It is impractical to investigate either protecting agents or protecting agent platforms in isolation of the other. Both sets of parties (i.e. agent owners and agent platform owners respectively) must have compatible, or at least accommodating, security policies and precautionary measures to ensure a long and fruitful business relationship.¹

The Mobile Agent Secure Hub Infrastructure (MASHIn) is our high level conceptual community of Internet mobile agent platforms and Internet mobile agent users who are dually interested in safe interaction and accountability for breaches of security policy. The central abstraction in the MASHIn are Mobile Agent Secure Hubs (MASHs). MASHs act as unbiased mediators and hold both parties accountable for their actions.

The MASHIn is a novel conceptual offering for raising the level of confidence in a mobile agent's results and safety. A mobile agent user sends the Mobile Agent Secure Hub (MASH) a message asking for mission protection of its mobile agent. During the lifetime of the agent's mission, the MASH is delegated the responsibility of managing and monitoring the itinerary and safe execution of the mobile agent.

As a trusted third party, the MASH reduces the mutual distrust lingering between mobile agent users and mobile agent platform service providers and necessitates accommodating security measures for both sets of parties.

It is important to note, that the MASH may well use TTP services offered by other specialist mobile agent security providers (for more discussion see Section 2.5) in fulfilling its responsibilities, but to keep things conceptually easy to follow we refer to the TTP entity providing mobile agent security services simply as the MASH.

¹ Whilst this paper's focus is primarily on precautionary measures for Internet mobile agents, we have discussed the benefits of employing TTP security services for both mobile agent users and agent platforms in [17].

2.2 Threats and Countermeasures

Table 1 presents some threats to Internet mobile agents from malicious agent platforms that can be (by the countermeasures listed) strongly mitigated in the MASH infrastructure.

Table 1: Threats to Internet mobile agents from attacking agent platforms, and MASHIn countermeasures.

THREAT	COUNTERMEASURE/S
Service Overcharging	Standard <i>a priori</i> understood service charges, or charge negotiated on the MASH - either way, charged by the MASH.
Illegal Purchases	Mobile agent user policy can stipulate purchases are initiated only on the MASH, and agent's electronic money remains solely on the MASH.
Reading Sensitive Data	Data stored in <i>real</i> secure event callback class (see Section 3.1), and/or group enveloping of agent folders for selected recipients only.
Cut and Paste Attacks [18, 19]	SeMoA's [20] agent kernel signature authentication, and recording at MASH.
Manipulative Modification of Agent Code	Our adaptation of Hohl's reference states approach (see Section 3.1); and mutual responsibility - careful agent programming!
Data Mining Agent User Preferences	Anonimisation via generically MASH authorised role, agent owner signature stripped at MASH, MASH signature added to agent, and agent deployed from MASH.
Agent Hijacking	MASH monitored agent Time-To-Lives (TTLs).
Non-accountability for Abuses	Percentage of calculated agent reference state failures on an agent platform is calculated and published. Unreliable processing times of agents can also be published.
Improper Agent Routing	MASH recorded pre-mission route, and dynamic routing decisions only performed on the MASH.
Inter-Agent Messaging Impersonation	Secure inter-agent messaging can only be performed from MASHs.
Impersonation as MASH	Agent has a list of MASHs it trusts (stored in its non-mutable section) before its mission is started, and a pre-determined static itinerary list is supplied to the MASH (the MASH monitors the agent's current location).
False Item Advertising Price and/or Availability Claim	Signed advertised price appended to mutable part of agent, as well as a no obligation expiration-limited purchase offer type hold.

2.3 MASHIn Security Services

The MASHIn can offer a number of security services for mobile agent users and their MASHIn-deployed agents including, but not limited to:

Agent user anonymity User anonimisation is achieved via role assignment, with the MASH authenticating and authorising the user/role mapping for the agent.

Agent itinerary anonymity The agent's travel history can be kept secret via MASH secure itinerary management [17].

Periodic mobile agent data integrity checking Reference state checks are performed on the MASH (or MASH-delegated *checking hosts* - see Section 3.1), at user/MASH defined itinerary breakpoints.

- A trusted base for performing sensitive transactions** For example, the MASH is a trusted environment enabling secure purchasing agents (in the traditional client-server manner), secure inter-agent messaging, and non-repudiation of digitally brokered contracts.
- Equitable, safe charging for mobile agent use of target agent platform services** Debit charging for agent use of target agent platform services is done safely on the MASH. No agent user payment details are disclosed to agent platforms.
- Per agent platform time-to-live monitoring** The MASH monitors user-defined minimum and maximum stay times for mobile agents at target agent platforms.
- Location tracking of agents** The MASH is a secure tracking service for monitoring the location of agents, and answering agent location queries for authenticated entities.
- Safe dynamic routing for mobile agents** Agent user policy can stipulate dynamic routing decisions (i.e. changes to the static, pre-determined mission itinerary) are permissible only on the MASH.
- Protected mobile agent user offline management, when mission is completed** The MASH is a secure repository for the case when the agent user is disconnected from the Internet and its agent has completed its mission. Agent user policy can stipulate holding the agent for a certain time at the MASH, securely emailing the agent to the agent's user, or some other user-defined alternative.
- A reputation service issuing agent platform credibility and reliability ratings** MASHs calculate the percentage of failed agent reference checks per agent platform. These statistics are publishable, as well as agent platform response times for processing agents.
- Trust facilitator** MASH can store user-defined trust statements in assisting agents to make safe dynamic routing decisions, and/or recall agents once a reference check failure rate for an agent platform reaches some user-defined threshold.

From the perspective of agent platforms, the MASH takes responsibility for authentication and (role) authorisation of agents, meaning the agent platform can concentrate further on delivering high quality services. The MASH can also scan agents for viruses or common logic attacks - such as denial of service attack code structures.

MASHs inherently possess reliable TTP auditing and conflict resolution capabilities since they retain a copy of all policies delegated to them by agent users, and policies they delegate to agent platforms. Thus, if there is a security breach by an agent platform (or agent), or a discrepancy in a negotiated mobile agent contract, the MASH has both the technical and authoritative capabilities to handle such matters judiciously. This helps to address the accountability stumbling block for the Internet mobile agent paradigm.

We anticipate that mobile agent platforms are much less likely to attack agents if their reckless actions are held accountable, and to be seen by all, jeopardising their business future. The co-verse (malicious agent owner) deterrent is also possible via the MASHIn concept.

2.4 MASHIn Core Components

Figure 1 on page 7 depicts the major software modules we foresee for MASHs. The vertical layering gives some insight into the dependencies of modules, with higher stacked modules building on the services delivered by lower modules.

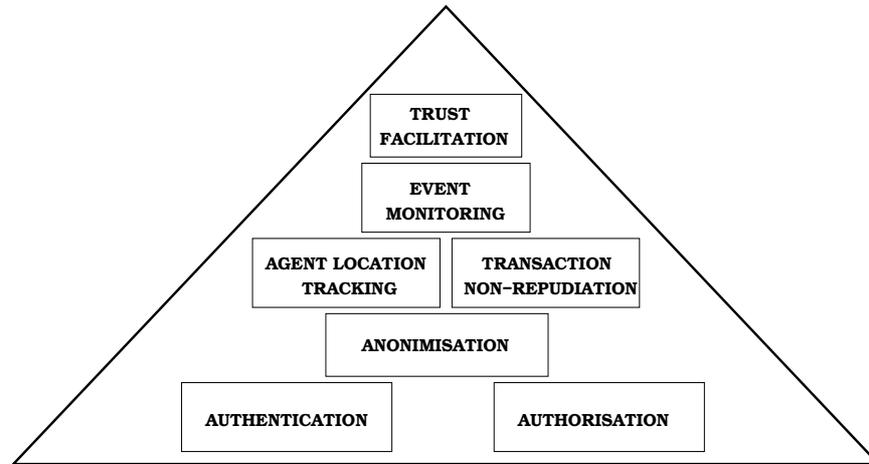


Fig. 1: Major MASHIn software modules.

The **AUTHENTICATION MODULE** is responsible for authenticating the owners of incoming agents requesting mission protection, and passing authenticated agents onto the authorisation module. Authentication of agent platforms and agents co-operating via secure inter-agent messaging is also the responsibility of the authentication module.

The **AUTHORISATION MODULE** is responsible for verifying an agent owner has been granted a claimed role, or role set for its agent's work on target agent platforms. This module also acts as a helper to other modules needing access control determinations - for example, it may assist the agent location tracking module in determining whether a requesting entity is allowed to know an agent's location. Considering another example, the authorisation module will also assist the transaction non-repudiation module in determining which agents can participate in business transactions, and under what conditions.

The **ANONIMISATION MODULE** is responsible for generating and keeping specific instances of authorised user/role anonymisation mappings secret from agent platforms. Higher level modules will interface with and through the anonymisation module to determine access control rights and record non-repudiable transactions involving mobile agents.

The **AGENT LOCATION TRACKING MODULE** is responsible for receiving mobile agent location updates, and also serves as a naming service for entities within the MASH infrastructure.

The **TRANSACTION NON-REPUDIATION MODULE** is responsible for being a non-repudiable record of mobile agent e-commerce related transactions, and charging agents for utilisation of agent platform services. It has the added responsibility to ensure non-disclosure of agent-user payment details.

The **EVENT MONITORING MODULE** monitors a wide variety of events within the MASHIn - for example agent migrations, agent state checking failures, and bottlenecked agent platforms.

The **TRUST FACILITATION MODULE** heads the tiered group of software models for MASHs, afterall the MASH's core function is acting as a trusted third party between mobile agent users (more specifically their mobile agents) and mobile agent platform service providers (more specifically their agent platforms). Trust relationships can be statically defined, based

on transitive entity relationships, or dynamically updated in response to events like an agent platform reaching some unacceptable threshold of mobile agent reference state failures.

2.5 MASHIn Deployment Considerations

We stipulate, once more, that we do not expect a single server to be responsible for offering all expected MASHIn said services. Deployment of MASHIn security services should be distributed and offered by many providers, for a number of reasons including:

- No MASHIn TTP should be the equivalent of *God*. That is, relying solely on any one entity gives that entity too much power. If that entity is attacked (for example, via a denial of service attack), or it is compromised (either via coercive ‘*soft*’ tactics or crafty hacking), the TTP security concept is broken.
- De-centralised TTP location and management is important so the transport and processing penalties of TTP utilisation are minimised wherever possible.
- The MASHIn should be capable of serving a large number of mobile agents (possibly millions at any one time).
- TTP data backup and data replication procedures would be required - both necessitate a distributed system.
- There will be entities who only trust certain TTP companies or hardware configurations to capably perform some TTP services, whilst dismissing their value at offering other TTP services.
- We anticipate that some TTPs would serve particular jurisdictional and/or locale boundaries.

Considering the major software security services depicted in Figure 1 on page 7, we anticipate close coupling (perhaps same in-organisation housing) between the Authentication, Authorisation, and Anonimisation modules. The Agent Location Tracking responsibility would be processor intensive, and we anticipate be the function of servers dedicated solely to that purpose. The Event Monitoring module would involve stationary agents near the source of events communicating with appropriately distributed handlers (case specific to the event/action). Dedicated servers should handle the Transaction Non-Repudiation module functions. And, yet again, we envisage dedicated entities to handle the responsibilities handled by the Trust Facilitation module.

Other deployment variations could be seen. We do not wish to be over-committed or narrow-minded in stipulating required infrastructural entities or inter-entity relationships at this stage of our MASHIn conceptual knowledge offering.

3 Considerations for the Design of MASHIn Deployed Mobile Agents

This section continues on from our conceptual overview of the MASHIn and MASH components in the previous section by noting the mutual obligation of agent designers and those delegating Internet mobile agents to MASHs for their mission’s protection.

In Section 3.1 we list some key design rules for best practice utilisation of MASHs. Discussion on the strategic employment of our minimal, but extensible, MASHIn collection of

secure mobile agent callback events is given in Section 3.2. Finally, in Section 3.3 we briefly overview the importance of the MASH as a pivotal entity in the forming of new trust (and business) relationships within the MASHIn.

3.1 Secure Agent MASHIn Design Recommendations

Malicious agent platform reading or lying attacks are, in theory, virtually impossible to prevent because once an agent is on an agent platform it is at the mercy of that agent platform's intentions, be they good or bad.

We introduced three secure itinerary routing patterns (strict routing, directed routing, and loose routing) for TTP-monitored secure agent itinerary management in [17]. By combining sound agent itinerary management with careful agent design, the usefulness of these otherwise stealthy malicious agent platform attacks can be reduced significantly. We suggest the following design rules which decrease the susceptibility of an agent's data to compromise and the damage severity impacted when attacked:

1. No agent should initiate any form of electronic payment, except on a MASH.
2. No agent should carry any payment details (e-cash, credit card details, etc.) to any target agent platform.
3. No agent should digitally sign a transaction for repudiation purposes (be that an e-commerce purchase, or a secure inter-agent message, etc.), except on a MASH.
4. Agents should not carry any personal information identifying its agent user to target agent platforms.
5. Agents should not carry any personal information identifying its agent home platform to target agent platforms.
6. Agents should have time-to-live (TTL) limits, and their location tracked by a MASH.
7. No dynamic routing decisions should be made by agents, except on a MASH.
8. Agents should be constructed to do a small, well-defined task.
9. Agents should attempt to complete their mission on as few as possible agent platforms.
10. Agent cloning should only be performed on MASHs.
11. Before migration to target agent platforms, agents should be sure to leave only non-sensitive code and data parts visible to non-trusted agent platforms.

It is pertinent to note that regardless of which combination of MASH agent precautionary measures are requested by an agent user, a poorly programmed agent remains vulnerable to blatant misuse by malicious platforms. Extravagantly programmed mobile agents should be avoided, and great care should be invested into what agent code and data (as little as needed) is left in plaintext on target agent platforms. Employment of our secure agent callback mobile agent privacy mechanism [21] is a highly effective strategy for achieving this.

The greater the flexibility in an agent's structure and capabilities, the more loopholes maliciously-behaving agent platforms are likely to be able to find and exploit. Agent reference state checking algorithms cannot detect clever exploits targeted at insecure agent programmer code [22, 23]. However, with proper employment of our secure callback mobile agent privacy mechanism, read attacks by non-trusted agent platforms do not reveal damaging code or data. This is because the sensitive code and data remains privy only to the MASH via the

real callback class, whereas the non-trusted platforms only see a *dummy* callback class (which could be empty but) having the same method and property forms.

In [21], we also discussed an adaptation of Hohl's [16, 22] reference states checking protocol for detecting attacks by malicious platforms against an agent's integrity. The adaptation overcame the two major weaknesses readily identified by Hohl - they being (i) a collaboration attack involving two consecutive hosts cannot be detected by future hosts, and (ii) input to an original execution cannot be held secret from the checking - possibly competitive - hosts.

We insert a *checking host* (appropriately chosen by the MASH) between non-trusting agent platforms in the original itinerary. The checking host's main purpose is to verify the reference states claims from the previous host, and if checked correctly forward the agent onto the next agent in its itinerary. If a discrepancy is detected, they report this to the MASH and also return the agent directly to the MASH. The MASH can then use this reference states failure in future reputation rating calculations of agent platforms. We introduce this reputation concept briefly in Section 3.3.

The second problem in Hohl's original reference states protocol is no longer an issue (in our adapted approach) because the inserted TTP delegate checking host has no stakeholder application interest in the agent's data or previous host's input.

3.2 Event Callback Methods

Supplied logic, privy only to the MASH, triggered on a mobile agents' callback events can be used strategically to support the autonomy and security goals for mobile agents in the MASHIn.

An extensible base set of interface class implementation methods is sent with the agent on the mobile agent user's mission protection request to the MASH. The base set of methods in this interface (and thus also in the concrete class as well) include:

- ▶ **afterEveryHost()**: Triggered on the MASH after every route to an agent platform (i.e. only applicable in the *strict routing* scenario).
- ▶ **afterNthHost(int[] nthHostList)**: Triggered on the MASH after a route to a specific agent platform in a pre-determined itinerary list. The list of pertinent hosts to invoke this method on after visitation is passed in as an integer array.
- ▶ **beforeNonTrustedHost()**: Triggered on the MASH before the agent is routed to a host platform specified as non-trusted.
- ▶ **afterNonTrustedHost()**: Triggered on the MASH after the agent is finished working at a host platform specified as non-trusted.
- ▶ **afterMission()**: Triggered on the MASH after the agent's mission is finished and at the MASH for the last time before being processed for shipping back (or alternative specified action) to the mobile agent user.

The agent user sends two versions of these implementations to the MASH - a *real* callback class (privy only to the MASH), and a *dummy* callback class which is inserted into the agent's classes (instead of the real callback class) when it is sent to non-trusted agent platforms.

A proof-of-concept application example, a purchase agent, was constructed to demonstrate the feasibility of our original mobile agent privacy approach. The proof-of-concept also provided an example of strategic utilisation of our callback methods. The purchase agent

would immediately initiate a secure purchase on the TTP if it found a *bargain* price (via the `afterEveryHost()` method). If no bargain offer was made, but the best offer was below an *acceptable* price, the agent would make a secure purchase on the TTP (via the `aferMission()` method). If, however, no acceptable offer was made, the agent would make no purchase. At no time, were the pre-determined bargain and acceptable prices revealed to the non-trusted hosts (i.e. data privacy was achieved), and at no time was the code for purchasing revealed to the non-trusted hosts (i.e. code privacy was achieved). Further details on the demo are provided in [21].

3.3 Reputation and Trust Relationships

The value of strategically incorporating TTPs to downsize the concerns and conflicting interests of mutually distrusting stakeholders in an emerging Internet mobile agent community can not be underestimated [17].

Mobile agent users and mobile agent platform owners may choose to do business instinctively via the TTP with stakeholders it would not have ordinarily trusted had there not been a middle party.

Reputation ratings may play an important role in this dynamic discovery of new business partners and/or customers. Inputs to the calculation of reputation can be wide and varied, for example efficiency of service and dependability for producing correct (non-attacking) results. The input cohorts in determining a reputation can be entity-defined, along with entity-defined importance weightings for the various chosen cohorts.

More details on our ideas for the application of reputation and trust management for realising, via MASHs, a dynamic and safer Internet mobile agent community in the MASHIn will be provided in [24].

4 Related Work

The importance of formalising a trust model for the stipulation of initial mobile agent system stakeholder relationships, and the inference of new trust relationships in a mobile agent infrastructure were highlighted by Tan and Moreau in [25]. Incorporation and extension, within the context of the MASHIn, of parts of Tan and Moreau's seminal work is likely.

Whilst delving deeper into our research, we came across an early use of social control in decreasing attacks involving agents [26]. This work looked into discouraging dishonest or irrational agents in open markets. Whilst we believe social control is a powerful mechanism, we feel it should not be the only means of achieving a more fair and safe system. Reputation ratings in the MASHIn are just one of many security service possibilities for mobile agent users wishing delegated TTP mission protection of their mobile agents via MASHs.

A draft architecture of an electronic market with secure mobile agents guaranteeing anonymity is presented in [27]. The architecture has a brokering scheme for forwarding on users' agents to market servers via a TTP agent server. We add role-based access control to our anonymity approach, secure mobile agent event callbacks (for achieving mobile agent privacy), different secure itinerary management patterns, user offline management, and a trust/reliability rating service via our MASHIn approach.

Algesheimer, Cachin, Camenisch and Karjoth [28] use a trusted third party for the submission of sensitive calculations to a *secure computation service* (the TTP) without learning anything about the submitted (encrypted) computation. This is an elegant approach for securing sensitive computations, but serves a distinctly different purpose (not deterrence, detection, and accountability for attacks - for which the MASHIn approach is intended for).

5 Conclusions and Future Work

There are no definitive answers or easy solutions for the protection of mobile agents deployed onto the Internet. The nature of Internet mobile agents, their flexibility to roam and work on untrusted hosts, poses great dangers for their misuse - especially from malicious hosts.

However, in our research we have made a conscious decision to look more at macro-level security issues concerning Internet mobile agents. Without methods to bring distrusting parties (namely agent owners and agent platform owners) to the bargaining table, much of the micro-level research performed on agent security to date in isolation is counterintuitive.

Our MASHIn conception, introduced for the first time in some depth in this paper, is a novel and extensible trusted third party-based strategy offering delegated secure management of mobile agents. MASHs are directed in this protector role by delegated policies from the agent's user.

MASHs can anonymise user agents via the role they are authorised to work under. Agent platforms, therefore, do not know which users they are attacking if they have malicious intentions. Moreover, (our adaptation of) reference state checking is performed at user-defined breakpoints in the mobile agent's itinerary on MASH-delegated TTP checking hosts. From those attacks which can be identified from the reference state-checking algorithms, statistics can be gathered on attacking hosts such as their frequency. These statistics can be published, offering the threat of diminished reputation as a strong deterrent against attacks.

For those attacks not readily identifiable from reference state-checking algorithms, we have offered a number of secure agent design guidelines which should be considered when manufacturing agents and employing MASHs for the protection of deployed Internet mobile agents. These measures can reduce the attractiveness of attacking and mitigate the severity when indeed attacked.

Our future work includes a more concrete investigation into reputation and trust management dynamics within the MASHIn. Detailed protocols for our suggested adapted reference states (with MASH-feedback) approach should also be formed.

References

1. Claessens, J., Preneel, B., Vandewalle, J.: (How) can mobile agents do secure electronic transactions on untrusted hosts? - A survey of the security issues and current solutions (2003) ACM TOIT, February 2003.
2. Hohl, F.: An Approach to Solve the Problem of Malicious Hosts. Technical Report 1997/03, Universit Stuttgart (1997)
3. Jansen, W.: Countermeasures for Mobile Agent Security. Computer Communications **Special Issue on Advanced Security Techniques for Network Protection** (2000)

4. Jansen, W., Karygiannis, T.: Mobile Agent Security. NIST Technical Report. Technical Report, National Institute of Standards and Technology (1999)
5. Posegga, J., Karjoth, G.: Mobile Agents and Telcos' Nightmares. *Annales des Telecommunication*, Special issue on communications security (2000)
6. Chan, A.H., Lyu, M.R.: The mobile code paradigm and its security issues (1999) <http://www.cse.cuhk.edu.hk/~lyu/student/mphil/anthony/gm99.fall.ppt>.
7. Farmer, W.M., Guttman, J.D., Swarup, V.: Security for Mobile Agents: Issues and Requirements (1996) Presented at the 1996 National Information Systems Security Conference, Baltimore, MD, USA. <http://csrc.nist.gov/nissc/1996/papers/NISSC96/paper033/SWARUP96.PDF>.
8. Hohl, F.: Time Limited Blackbox Security: Protecting Mobile Agents From Malicious Hosts. *Lecture Notes in Computer Science* **1419** (1998) 92–113
9. Uwe Wilhelm: Cryptographically Protected Objects. Technical report, Ecole Polytechnique Federale de Lausanne, Switzerland (1997)
10. Wilhelm, U.G., Staamann, S., Buttyán, L.: Introducing trusted third parties to the mobile agent paradigm. In Vitek, J., Jensen, C., eds.: *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*. Volume 1603. Springer-Verlag, New York, NY, USA (1999) 471–491
11. NAI Labs: Secure Execution Environments: Self-Protecting Mobile Agents (2002) <http://www.pgp.com/research/nailabs/secure-execution/self-protecting.asp>.
12. Riordan, J., Schneier, B.: Environmental Key Generation towards Clueless Agents. *Lecture Notes in Computer Science* **1419** (1998) 15–24
13. Sander, T., Tschudin, C.F.: Protecting Mobile Agents Against Malicious Hosts. In Vigna, G., ed.: *Mobile Agents and Security*, LNCS, Heidelberg, Germany, Springer-Verlag (1998) 44–60
14. Roth, V.: Mutual Protection of Co-operating Agents. In: *Secure Internet Programming*. (1999) 275–285
15. Fischmeister, S.: Building Secure Mobile Agents: The Supervisor-Worker Framework. Master's thesis, Technical University of Vienna (2000)
16. Hohl, F.: A framework to protect mobile agents by using reference states. In: *International Conference on Distributed Computing Systems*. (2000) 410–417
17. Giansiracusa, M., Russell, S., Clark, A.: Clever Use of Trusted Third Parties for Mobile Agent Security. In: *Applied Cryptography and Network Security - Technical Track*, ICISA Press (2004) 398–407
18. Roth, V.: On the robustness of some cryptographic protocols for mobile agent protection. *Lecture Notes in Computer Science* **2240** (2001) 1–??
19. Roth, V.: Empowering mobile software agents. *Lecture Notes in Computer Science* **2535** (2002) 47–63
20. Roth, V., Jalali-Sohi, M.: Concepts and architecture of a security-centric mobile agent server. In: *Fifth International Symposium on Autonomous Decentralized Systems (ISADS 2001)*, IEEE Computer Society (2001) 435–442
21. Giansiracusa, M., Russell, S., Clark, A., Hynd, J.: A Step Closer to a Secure Internet Mobile Agent Community (2004) Submitted to The Fifth Asia-Pacific Industrial Engineering and Management Systems Conference (APIEMS 2004).
22. Hohl, F.: A Protocol to Detect Malicious Hosts Attacks by Using Reference States. Technical report, Universit Stuttgart, Fakult Informatik (1999)
23. Farmer, W.M., Guttman, J.D., Swarup, V.: Security for Mobile Agents: Authentication and State Appraisal. In: *Proceedings of the Fourth European Symposium on Research in Computer Security*, Rome, Italy (1996) 118–130
24. Giansiracusa, M., Russell, S., Clark, A., Hynd, J.: MASHIn Reputation Ratings as a Deterrent Against Poor Behaviour (2004) To be submitted to The 3rd Workshop on the Internet, Telecommunications and Signal Processing (WITSP 2004).
25. Tan, H.K., Moreau, L.: Trust Relationships in a Mobile Agent System. *Lecture Notes in Computer Science* **2240** (2001) 15–30
26. Rasmusson, L., Jansson, S.: Simulated social control for secure Internet commerce. (1996) 18–26
27. Mandry, T., Pernul, G., Röhm, A.W.: Mobile agents on electronic markets – opportunities, risks and agent protection. In Klein, S., Gricar, J., Pucihar, A., eds.: *12th Bled Electronic Commerce Conference*, Moderna Organizacija (1999)
28. Algesheimer, J., Cachin, C., Camenisch, J., Karjoth, G.: Cryptographic Security for Mobile Code. In: *Proc. IEEE Symposium on Security and Privacy*, IEEE (2000)