Towards Payment Systems for Mobile Agents

Christian Anhalt[a] Stefan Kirn[a]

[a] *Universität Hohenheim, 70593 Stuttgart, Germany,*
canhalt|kirn@uni-hohenheim.de,

Abstract

Future areas of application for mobile agents technology are open, distributed and decentralized networks, where Mobile Agents autonomously perform tasks on behalf of their principals . In this paper, we address the conceptual design of payments for and between Mobile Agents, particularly against the background of the Malicious Host Problem. We analyze the requirements on payment systems for Mobile Agents and give a survey on existing technical solutions partially matching these requirements. We then discuss a new concept that does not require Mobile Agents to hold secrets and that allows allocations of digital coins and agents even in an environment that contains Malicious Hosts. The concept is based on the principles of accounting systems.

1 Introduction

Mobile Agents (MoA) are software entities able to migrate autonomously from one host to another (cf. [22]). Within the last decade, the paradigm of MoA was discussed broadly and also controversially (cf. [39], [11]). Many suggestions for future fields of application of MoA have been made. [19] and [66] proposed the application of MoA in distributed databases and digital libraries. [34], [21] and [67] investigated their applicability for intrusion detection; [29], [32], [43] addressed telecommunication networks, while [4] focused on large sensor networks; [35] and [61] proposed software updates and [56], [57] and [41] information retrieval as fields of appli-cation. MoA perform their tasks in open, distributed and decentralized networks. They are executed by hosts owned and operated by people different from the agents' owners. To accomplish their tasks, MoA migrate between hosts and access resources (e.g., computing power, but also services such as yellow book services) provided by hosts as well as other (mobile) agents. Since hosts and MoA are owned by different persons, the need for coordination of (economic) interests between these principals[1] arises.

Setting up payment systems for Mobile Agents (PSMA) is one possible solution to meet these needs. When designing PSMA, the Malicious Host Problem (MHP) has to be taken into account. The MHP refers to the inability of MoA to use as well as hide information from hosts executing them (cf. Section 2). For (secure and tamper-proof) PSMA, this raises questions about their technical feasibility, since existing electronic payment systems are based on the usage of specific secrets, e.g., private keys.

In this paper, we deal with the design of (secure and tamper-proof) PSMA. According to requirements on PSMA not complied by other solutions we develop a PSMA subsystem for allocating[2] MoA and digital coins. Assuming that the MHP is not adequately solvable yet (cf. [59]), we avoid using any secrets held by MoA and address PSMA allocation systems from the perspective of transparency and self-control. Our concept bases on an accounting web distributed among and managed by groups of hosts executing the MoA. It is designed as a decentralized and distributed open system and allows an ex post detection of double-spending and also thievery of coins.

The paper is structured as follows. In Section 2 we discuss PSMA taking into account the MHP. In Section 3 we analyze the requirements for a PSMA by deducing them from the goals of the principals (3.1). We give a survey on existing technical concepts possibly suitable for PSMA and discuss them in view of the requirements that have been identified (3.2). Section 4 describes our allocation concept. Starting with the assumptions we made in 4.1, we give a rough draft of our concept in 4.2. Section 4.3

[1] Within this paper, owners of Mobile Agents or hosts are named principals. Except for the term, there are no relations to the concept of principals used in the principal-agency theory.

[2] With the allocation of MoA and coins, we address the concept of linking at least one coin to one MoA in a way that only this agent can dispose of the coin.

gives a simple example of the double-spent detection mechanism. We discuss unsolved questions, next steps of our research and the evaluation of future results in Section 5.

2 Payment Systems for Mobile Agents and the Malicious Host Problem

A PSMA is a system enabling MoA to balance debt and credit. Debt and credit are generated when agents access resources provided by other agents or hosts, both owned by principals different from theirs. We assume that principals have economic interests, which they link to the agents or hosts administrated and owned by them. Thus, PSMA can be seen as systems supporting the (price/market based) coordination of (economic) interests of principals – MoA pay for access to resources using digital money provided by their principals. The resource-owning principals receive the money and are able to either deposit it in their bank accounts or to use it otherwise (cf. Figure 1). Like any kind of money, the digital information MoA use for payment has to fulfil three functionalities: medium of exchange, storage of value, and measurement of value. (cf. [9] [50]).

Conventional electronic payment systems generally can be grouped into two classes – token/cash-based or account/check-based electronic payment systems (cf. [1]). Account/check-based systems use orders signed by the payer and sent to a central clearing institution in order to transfer the specified amount of money to the payee's account. Since a clearing institution is involved in each financial transaction, one of their characteristics are high minimum transaction costs (considering both the involvement of a third party and high security costs of the clearing server), compared with token/cash-based systems. Thus, they are unsuitable for nano- and micropayments, which will primarily take place in future PSMA. Token/cash-based payment systems incur lower transaction costs, especially when realized as multistage and offline systems (assuming that tampering can be prevented). Thus, they may be better suited for nano- and micropayments between MoA.

For our work, we focus on PSMA defined as systems consisting of token/cash-based electronic money (cf. [17] for definition) that MoA can use for exchange, storage and measurement of value in open, distributed and decentralized networks.

The fact that MoA can be executed by hosts not owned by the same principals leads to the MHP. It refers to the ability of any agent-executing host to attack the MoA, for example by spying out or manipulating code, data or execution state (cp. [13] [7] [30], see [26] for a detailed analysis of the MHP). As a result, it is not possible for MoA to hide and, at the same time, use information without risking manipulation or copying of this information by the executing host[3].

Conventional electronic payment systems use cryptographic methods for generating electronic money. They rely on secrets held by the payer, the payee and other involved actors. These secrets can either be private keys used for signature generation and authentication or digital coins (cp. [46]). Therefore PSMA build on the basis of existing electronic payment technologies cannot address both the exchange functionality and the value storage functionality of money. Trying to implement both functionalities, situations would arise where malicious hosts can copy and/or steal money from MoA, even with significantly reduced chances for traceability. This is true for conventional payment system based PSMA as long as no general solution for the MHP exists. It is also independent from the type of payment system, since both check/account-based and token/cash-based payment systems rely on secrets.

3 Adaptable technology concepts for PSMA

3.1 Requirements on PSMA

Conventional electronic payment systems can be described by means of a generic (financial) transfer and role model (cf. [44]). Within this model, all roles assignable to actors are identified and related to specific financial transfers, e.g., payment, withdrawal or deposit. In order to identify possible roles principals can take within a PSMA, we expanded the original model from [44] to a (financial) transfer and role model describing PSMA (see figure 1). Thus, we first modeled (financial) transactions generated by agents upon MoA roles and integrated them into the original model. Then we deduced additional principal roles involved in the payment circuit.

[3] For our work, we will only focus on problems caused by the MHP. Other security issus related to, e.g., insecure networks or malicious agents will not be considered.

The names of agent roles have been selected to reflect their position within the payment circuit. Additionally, we defined a virtual currency area that represents an arbitrary set of (connected) MoA execution platforms. Agents' payments are restricted to this currency area.

According to the roles and their specific position in the financial circuit, we assigned superordinate goals to each principal's role. We then decomposed and consolidated these goals from a MoA oriented perspective, i.e., the principals' goals were transferred to the MoA as possible. As a result, we obtained high-level requirements on PSMA.

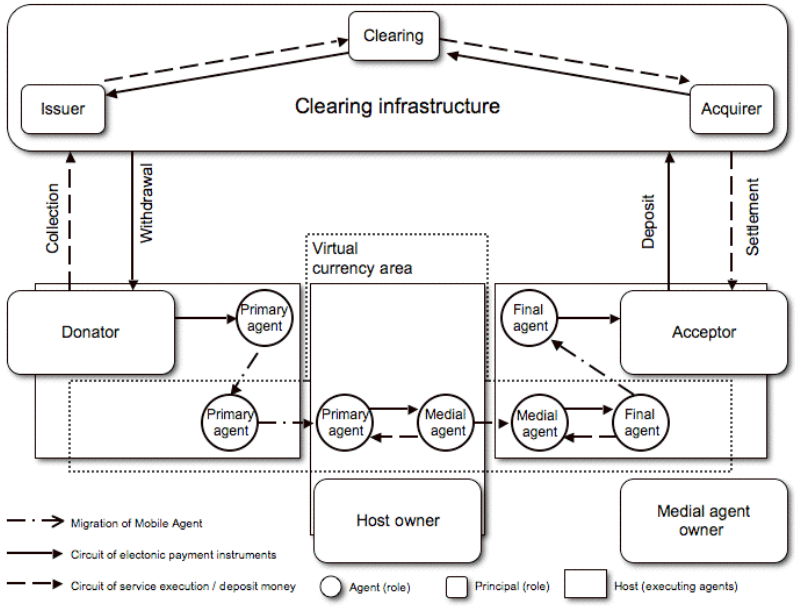

Figure 1: Financial transfer and role model of a PSMA

Among high-level requirements that are, e.g., related to the general functionality of MoA and platforms, we identified two groups of high-level requirements important for our work: requirements related to electronic money per se and requirements related to the allocation of electronic money to MoA.

The requirements related to electronic money are identical with those defined in literature for conventional electronic money: identification of counterfeit and double-spent money, non-linkability of payment and withdrawal, traceability of transactions for the payer and the payee, ex post integrity of payments, adequate transaction costs, etc. (cf. [25], [62]). Important for our works are the requirements related to the second group, the (one-to-one) allocation of electronic money to MoA. In conventional electronic payment systems, similarly to real life, unambiguous allocation of coins is realized through the institute of ownership: someone owns coins and/or specific information and thus has the power of disposition over them. For MoA, exclusive possession is not possible. Thus, an allocation (sub-) system based on a different concept but providing equal functionality is needed (which is, of cause, the first high-level requirement).

The high-level requirements we identified on an allocation system providing one-to-one allocations are:

- consistency of allocations even by migrations of agents,
- verifiability of allocations, i.e., the ability to identify tampered allocations by MoA, hosts and principals,
- interchangeability of allocations, i.e., the transfer of allocations from payers to payees,
- robustness of allocations, e.g., their restorability in case of host or connection breakdown,
- divisibility of allocations, unless the allocations are not related to an atomic finance unit.

These requirements overlap with requirements for electronic money. For instance, while implementing all required allocation functionalities the requirement "identification of double-spent money" could also be met as long as verifiability of one-to-one allocations is granted.

3.2 Research related to PSMA

A final and complete technical solution for the MHP would enable MoA to hide and use secrets. In this case, conventional electronic payment technology could be used for PSMA. Thus, a literature review has to consider both explicit work on payment technologies for MoA and implicit work addressing the MHP directly.

The implicit research on PSMA can be divided into six groups: research on Mobile Cryptography, Code Obfuscation, Environmental Key Generation, Internal Hardware Extensions, Reference States and Signature Delegation techniques for MoA. Explicit research is based upon distributed threshold schemes (which are also used by some implicit solutions).

The conceptual idea of *Mobile Cryptography* is to encipher the code and all data of a MoA in a way so that the agent is still executable in its encoded form. Ideally, the encoded MoA is still able to handle inputs and outputs without loosing its cryptographic protection. [48] originally addressed this idea, ongoing work has been done by [49], [37], [10], [36] and [65]. Whereas these approaches focus only on one host participating in the agents' execution, [2], [68], [55] and [16] addressed the integration of additional (semi-trusted) hosts. Within Mobile Cryptography approaches, an allocation subsystem for PSMA may be build upon the MoA ability to keep secrets. However, there are no prototype implementations known. Additional research is still needed to protect MoA on a higher level (cf. [65]). Also, characteristic of all Mobile Cryptography approaches are very high requirements for additional computation power, bandwidth and complexity handling compared to "non-encoded" MoA. Even when these concepts are implemented, high transaction costs will result when using them for PSMA.

Research on *Code Obfuscation* for MoA has been done by [27], [28] and [42]. Assuming that obfuscation techniques allow only temporary protection, [28] proposed to limit the lifetime of MoA according to the time needed for breaking the obfuscation. [42] suggested hiding secret sub-programs in the remaining MoA code. Both approaches would allow implementing allocation subsystems for PSMA on the basis of temporal secrets. According to the limitation of lifetime, a synchronization concept that can handle lifecycles of agents, digital coins and allocations would be needed. Additionally, the question arises whether it is possible to prevent automatic analysis of obfuscated code (which would lower the lifetime of agents to a few seconds).

[45] proposed and [24] extended the concept of *Environmental Key Generation* for MoA, addressing the idea that a MoA can use, e.g., private keys only when specific environmental information is available. Based on the comparison of doubled hashed information, this concept may be suitable for specific, pre-defined one-time-payments, as it allows allocations based on secrets until the right environment is found. Environmental Key Generation is not suitable for allocations needed in (multistage) PSMA, as this would require ex-ante knowledge of all future transactions (payer and payee) of each MoA.

[63], [64] and [60] suggested approaches based on *Internal Hardware Extensions* and Java Cards. Although these concepts of tamper-proof hosts-in-hosts would allow complying with the requirements on allocation subsystems, their practical applicability is doubtful since the hosts' principals would have to abandon their authority over the hosts. Also, the investment in hardware extension would increase the transaction costs.

The family of *Reference States* address approaches based on the ex-post comparison of MoA execution states. Two subclasses can be identified: approaches that deal with the ex-post comparison by hosts next in the migration route of MoA (cf. [18], [40], [54], [14], [MaTM–2004], [3]) and approaches that address the comparison by principals after the MoA have finished their tasks (cf. [58], [6], [64]). This class of approaches is of limited applicability to PSMA, since reading of data is neither inhibited nor detectable.

The *Signature Delegation* group consists of undetachable signature (cf. [33], [12], [52], [8]), proxy signature (cf. [47], [5]) and blind signature concepts (cf. [20]). These approaches are either insecure against the MHP (proxy and blind signatures) or restricted to a specific, in detail pre-described use (undetachable signatures). Thus, for an allocation subsystem the same applicability problem arises as when using Environmental Key Generation: ex-ante knowledge of all future transactions would be required.

An *explicit approach* to enable MoA to pay for, e.g., service access is presented by [15]. Within his concept, a MoA belongs to a group of n MoA located at different hosts. The MoA cooperate when a payment is needed. Using a *(k,n)* threshold scheme (proposed by [51]), each MoA carries a share of a secret, in this case a private key, and sends its share to a special MoA when payment is needed. No malicious host is able to steal or copy electronic money carried by the MoA without getting at least k

shares from the independently migrating agents. The approach is designed for a one-time payment between a MoA and a host. It requires that the payee host does not execute the MoA. Also, the protocols only protect the money allocation of paying agents. Payments between two MoA are not protected by this concept.

Examining the five high-level requirements related to allocation subsystems, we have to note that, with the exception of *Hardware Extensions*, none of the outlined concepts is suited to serve as a basis for a PSMA yet. Either further development and evaluation is needed or the characteristics of the technology concepts themselves are not suitable for adaptation. For Hardware Extensions, their practical applicability is questionable.

4 An accounting supported PSMA

4.1 General conditions and basic assumptions

The development of an allocation subsystem for PSMA should take into account general conditions of MoAs' environments. By making following assumptions we tried to capture these conditions:

- Within MoAs' environments, two types of actors can be identified: MoA and (executing) hosts. Both are able to join and leave the network.
- When a MoA is located on a host, it is not possible to detect what happens to it (e.g., if it is being copied or analyzed). Hosts have to be interpreted as black boxes, only revealing information about their internal processes when something leaves the box (cf. [28]).
- Within MoAs' environments, the only two ways to (at least partially) control hosts are by involving other hosts[4] (independently from agents executed on them) and, after the task has been accomplished, by the MoAs' principals (cf. [58]).
- MoA cannot keep data, code or execution state secret to the executing hosts and, at the same time, use it when interaction is required (binding existence of the MHP).
- It is possible for MoA to carry data in a way that anyone can read and add data, but later manipulations on once written data can be identified (realized by public verifiable integrity, cf. [23],[53].

As basic requirements we assume the existence of the following infrastructure and hosts' properties:

- Each host joining the environment has a (published) unique identifier (uid).
- A Public Key Infrastructure (PKI) exists, which enables (a) hosts to create and validate digital signatures (related to their uid), (b) principals to sign their MoA and (c) MoA to transport secrets for their principals (without using them).
- An anonymization service exists that decouples MoA from their principals. As a result, everyone can trace MoA and their financial transfers without identifying the principal.
- There is a clearing infrastructure acting as an institution that withdraws and accepts electronic coins. The clearing infrastructure always knows how much money circulates by balancing deposited and issued coins.

As far as the last assumption is concerned, we decided to use digital coins according to the MicroMint micropayment scheme as it has been proposed by [46]. MicroMint coins consist of a fixed number of strings (of equal bit-length) producing the same hash value when operated by a specific hash function. No computing-intensive public-key operations are required. Due to the non-injective and the collision resistance of the hash function, coin generating for a small number of coins is extremely expensive, but gets progressively cheaper (per coin) when generating more coins. Since coin production is economical only with a very large production volume, counterfeiting coins is prevented as it is uneconomical.

Our basic idea for an allocation subsystem is twofold, addressing both the position of the allocation subsystem and its openness. As far as position is concerned, the subsystem has to be managed not at the MoA level, but at the host level. That is, hosts manage the allocations between MoA and digital coins – they change allocations if MoA pay, verify allocations and manage valid transfer between each other if MoA migrate. Hosts, i.e., principals operating hosts, have to be responsible for a valid allocation management. To prevent tampering, hosts have to be real time controlled by other hosts, for instance, a group of hosts that cooperate for allocation management. MoAs are integrated in this allocation subsystem only passively for documentation and control. They carry information about allocations in a way that is not modifiable ex-post. In terms of openness, the idea is to restrict the PSMA and thus the allocation subsystem so that it is open to MoA and hosts, but

[4] This assumtion bases on *proposition 1* given in [2].

"closed" to digital coins and allocations. Thus a currency area has to be created. Digital coins have to be valid only if they are located permanently inside this area. Once they leave the area, coins are not restorable. As a consequence, the overall amount of money in circulation is constant (or decreasing). Detection mechanisms are needed to give alert when the amount rises. Double spent coins and the use of counterfeit coins can thus be avoided.

4.2 Abstract concept of an accounting based PSMA allocation system

Using accounting techniques for a PSMA allocation system, an accounting web for hosts and MoA has to be created. Normally, accounting systems are kept by one central institution. For PSMA, the accounting web needs to be both distributed and decentralized, since a centralized design would lead to higher transaction costs (cf. Section 2). It would also raise the question of (secure) authentication and it does not comply with the offline characteristics of digital money given in Section 2.

Our concept uses files representing accounts. These account files are made forward secure by cryptographic methods, i.e., they are designed to enable writing and reading to everyone, but inhibit (unnoticeable) deletion and manipulation of previously stored information (cf. [64], [23], [53] for forward and publicly verifiable integrity). The accounts are managed by hosts and are related to groups of hosts, (single) hosts and MoA. Each financial transaction is recorded by a double-entry registration in at least two accounts. The accounting web is closed, i.e., no transaction outside of and, even more important, no transaction back into the accounting web is allowed. The clearing infrastructure is part of the accounting web. Also, only MoAs are allowed to pay and receive payments and thus own money. Hosts have to be represented by (non-mobile) agents when they receive or spend money.

Structure of the accounting web: In addition to the account τ^I related to the clearing infrastructure, an accounting web is build on the basis of three kinds of account rows[5]: rows of accounts τ^{MA} related to MoA, rows of accounts τ^H related to hosts and rows of accounts τ^G related to groups of hosts.

Within an accounting web consisting of only one group $G=1$ of n hosts and m MoA, $n,m \in N$, located on these hosts, the following accounts exist:

- τ_1^I , the clearing infrastructure account, where all withdrawals and deposits are recorded,
- τ_1^G , the group account, where all input and output transactions of the group $G=1$ are recorded,
- $\tau_1^H,...,\tau_k^H,...,\tau_n^H$, the hosts' accounts, where all transaction between hosts (i.e., agents located at these hosts) are recorded. Each account τ_k^H is located on and managed by one host. The set of all hosts' accounts τ_k^H that belong to group $G=1$ is named $T_{G=1}^H$,
- $\tau_1^{MA},...,\tau_i^{MA},...,\tau_m^{MA}$, the MoA accounts where all transaction between agents are recorded. Each account τ_i^{MA} is related to and located at a MoA and managed by the hosts executing this agent. The set of all MoA accounts of group $G=1$ is named $T_{G=1}^{MA}$, while the set of all MoA related to host k is $T_{H=k}^{MA}$

Within a group of hosts, the sum of the MoA account row, and the sum the hosts account row and the group account represent the same value: the total amount of money that exists within this group.

(1) $\left|\tau_1^G\right| = \sum_{k=1}^{n} \left|\tau_k^H\right| = \sum_{i=1}^{m} \left|\tau_i^{MA}\right|$, where $|\tau|$ is the sum of all entries of an account τ,

Within the accounting web, the sum of all group accounts has to be equal to the sum of the clearing infrastructure account. For a given one-group accounting web, that is $\left|\tau_1^I\right| = \left|\tau_1^G\right|$. For accounting webs consisting of o groups, $o \in N$, it has the form:

(2) $\left|\tau_1^I\right| = \sum_{p=1}^{o} \left|\tau_p^G\right|$

As mentioned above, the accounting web has to be open for hosts and MoA. MoA can enter the accounting web by migrating to a host that is already member of the accounting web and by receiving an empty or credited account from the clearing infrastructure. When receiving a credited account, the MoA also receives digital coins (of the same value) from the issuer. It is not possible for MoA to bring

[5] An account row is a set of accounts structuring a datum in a specific way. Multiple account rows addressing the same datum allow different perspectives on this datum, e.g. source and disposition of funds.

money with them into the accounting web. Hosts can enter the accounting web when enough new hosts are available for a new group (in our specification: *n* hosts). They are also not allowed to bring money with them and receive their accounts from the clearing infrastructure. Both MoA and hosts are able to leave the web. Since money cannot re-enter the accounting web, they need to be cleared by transferring all money to other hosts/agents staying inside the web. Otherwise the money is lost.

Transactions within the accounting web: Within the accounting web, each financial transaction, i.e., transfers of coins from one MoA to another or MoA migration, generates entries to the accounts. We describe transactions by accounting functions e_f, $f = 1,...,h$, $h \in N$. An accounting function e_f links specific debit and credit accounts and a set of coin IDs to a value. For all e_f applies:

(3) $e_f : T^D \times T^C \times ID \rightarrow V$, where T^D gives all accounts being debited, T^C gives all accounts being credited, *ID* is the set of all coins (coin id) and *V* is the set of all values composable of the elements of *ID*.

Thus, within a specific group of hosts three archetypes of transactions can be identified:

- payments between two MoA located at the same host: $e_f\left(\left\{\tau_i^{MA}\right\},\left\{\tau_j^{MA}\right\},\left\{id_r\right\}\right) = v_{ijr}$,
- payments between MoA located at different hosts: $e_f\left(\left\{\tau_k^{H},\tau_i^{MA}\right\},\left\{\tau_l^{H},\tau_j^{MA}\right\},\left\{id_r\right\}\right) = v_{kiljr}$
- migrations of a MoA between hosts: $e_f\left(\left\{\tau_k^{H}\right\};\left\{\tau_l^{H}\right\},\left\{id_r\right\}\right) = v_{xyr}$

With more than one group, two additional archetypes can be identified:

- migrations of MoA between groups of hosts: $e_f\left(\left\{\tau_p^{G},\tau_k^{H}\right\},\left\{\tau_q^{G},\tau_l^{H}\right\},\left\{id_r\right\}\right) = v_{pkqlr}$
- payments between MoA located in different groups of hosts: $e_f\left(\left\{\tau_p^{G},\tau_k^{H},\tau_i^{MA}\right\},\left\{\tau_q^{G},\tau_l^{H},\tau_j^{MA}\right\},\left\{id_r\right\}\right) = v_{pkiqljr}$

Considering the clearing infrastructure, another two archetypes of financial transactions can be identified:

- withdrawal of digital coins: $e_f\left(\left\{\tau^{I}\right\},\left\{\tau_q^{G},\tau_l^{H},\tau_j^{MA}\right\},\left\{id_r\right\}\right) = v_{Iqljr}$ and
- deposit of digital coins: $e_f\left(\left\{\tau_p^{G},\tau_k^{H},\tau_i^{MA}\right\},\left\{\tau^{I}\right\},\left\{id_r\right\}\right) = v_{pkiIr}$

Each accounting function is related to entries in at least two accounts. These entries include the receiving/spending account, the value *v* that is being transferred, the IDs of the coins used for the transfer, a time stamp, the host recorded the entry and a description of the service/resource paid/sold.

Characteristics: The accounting web described above allows controlling the integrity of the system as it allocates digital coins to MoA. The allocation is realized by recording transfer of ownership to MoA, hosts and group accounts. The presented accounting based allocation concept addresses all five requirements shown in Section 3.1. Fraud allocations can be detected in two ways: by ex post examination of the payments a MoA has made and by checking the account balance.

The ex-post examination of MoA payments is still comparable to conventional logging. Each principal is able to check the income and the payments of its MoA either after the task fulfillment or after a specific time period. He can thus identify payments that do not correspond to the agent's task or, when for instance cryptographic tracing according to [58] is enabled, those that do not match the agent's execution states. Unlike normal logging, each log entry exists both in the agent's account and the other accounts charged. Therefore the MoA is traceable within the accounting web as long as it possesses at least one digital coin. Also, when fixing this coin to the MoA, no host is able to delete the MoA. The accounting trace would end at this host, identifying it as malicious.

With the account balance, we describe the validity of equation (1) for all groups *r* of hosts and the equation (2) for the whole accounting web at any time of a PSMA lifecycle. The validity of equation (1) is checked by a group *r* of *n* hosts each time a MoA enters or leaves the hosts that belong to group *r*. Each host has to cast the MoA accounts located on it and publish the balance for the other group members. All hosts within the group are able to check if the overall balance is equal to the balance of the group account τ_p^G. If

$$\left|\tau_p^G\right| \neq \sum_{i=1}^{m}\left|\tau_i^{MA}\right|$$

an error or fraud is detected and further steps have to be taken, e.g., the check of equation (2) and the concrete analysis of the group's activities.

As long as (1) and (2) are true, the amount of money withdrawn (and not yet deposited) is equal to the amount of money circulating within the accounting web. It is not possible to double spent digital coins, since with the (second) entry this balance is lost and the double-spending is detected by the next balance check. Also, when coins are stolen, e.g., by a malicious host executing a MoA, the stolen money is worthless, as its reintegration into the accounting web would either lead to a discrepancy or, when kept outside the accounting web, cannot be deposited back in the clearing infrastructure. No payee will accept it.

4.3 A simple example

The simple example of an accounting web supported PSMA addresses the prevention of double spending coins by detecting account unbalances. Thus, we make no use of the option to compare coin IDs as they are stored within the agents', hosts' and groups' accounts. Also, we abstain from an explicit integration of the clearing infrastructure, i.e., the amount of circulating money is constant.

The accounting web consists of two groups with *n=2* hosts. At each host, one MoA is located. Each MoA carries two digital coins already recorded in the agents', the hosts' and the groups' accounts. One coin is fixed to the agent, i.e., the agent is not allowed to spend the coin. According to this structure illustrated in figure 2, we have the following accounts:

- τ^I, where τ^I is constant and $|\tau^I| = 8$,
- τ_1^G, τ_2^G, where $|\tau_1^G| = |\tau_2^G| = 4$,
- $\tau_1^H, \tau_2^H, \tau_3^H, \tau_4^H$, where $|\tau_1^H| = |\tau_2^H| = |\tau_3^H| = |\tau_4^H| = 2$
- $\tau_1^{MA}, \tau_2^{MA}, \tau_3^{MA}, \tau_4^{MA}$, where $|\tau_1^{MA}| = |\tau_2^{MA}| = |\tau_3^{MA}| = |\tau_4^{MA}| = 2$

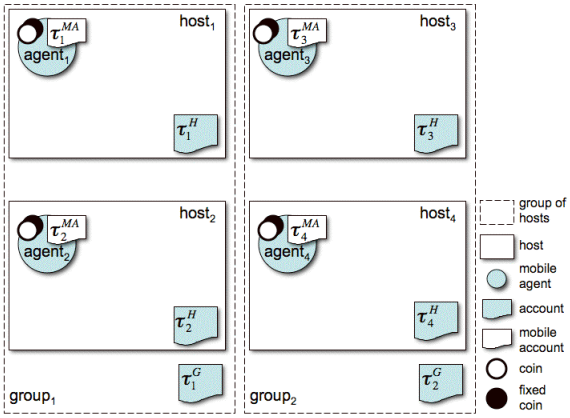

Figure 2: Structure of hosts, MoA and accounts

Assume that $host_1$ is malicious and generates $coin_{1*}$ by copying $coin_1$ of $agent_1$. When $host_1$ is the only malicious or corrupt actor (hosts and MoA) within the web and only MoA are allowed to use money, $host_1$ is able to spend $coin_{1*}$ in two ways only: (1) to generate $agent_{1*}$ that migrates (away from $host_1$) and spends $coin_{1*}$; (2) to disguise itself as $agent_1$ and spend $coin_{1*}$ to e.g. $agent_4$.

Generating an agent$_1$: With the migration of $agent_{1*}$ from $host_1$ to, e.g., $host_2$, the transaction

$$e_1\left(\tau_1^H ; \tau_2^H ; id\left(coin_{1*}\right)\right) = 1$$

is recorded within τ_1^H and τ_2^H. Assume now that $agent_1$ has to migrate too, e.g. to $host_2$, the transaction

$$e_2\left(\tau_1^H ; \tau_2^H ; id\left(coin_1, coin_2\right)\right) = 2$$

would lead to an unbalance within $group_1$, as $|\tau_1^G| = 4$ (nothing has entered or left the group) and $\sum |\tau_i^{MA}| = 5$, $i = 1,1*,2$.

Masquerating as agent$_1$: Using this second option, $host_1$ would initiate a transaction

$$e_3\left(\tau_1^G, \tau_1^H, \tau_1^{MA} \;;\; \tau_2^G, \tau_4^H, \tau_4^{MA} \;;\; id\left(coin_{1*}\right)\right) = 1.$$

Also, when $agent_1$ migrates to, e.g., $host_2$ and thus a transaction

$$e_4\left(\tau_1^H \;;\; \tau_2^H \;;\; id\left(coin_1, coin_2\right)\right) = 2$$

is needed, an unbalance occurs within $group_1$, as $\left|\tau_1^G\right| = 3$ (e_3 was recorded to τ_1^G) and $\sum\left|\tau_i^{MA}\right| = 4$.

For the detection of double spending, a transfer of both $coin_1$ and $coin_{1*}$ is needed. One option to achieve this is by imposing time restrictions on MoA stopovers on hosts or group internal agent rotation directives. There are also additional options for malicious hosts when more than one corrupt actor exists, but at least they all will be detected when reintegrating double spent money into the accounting web (at least when depositing the false coins).

5 Summary and Future Work

In this paper we addressed the design of a PSMA. We identified requirements on PSMA and discussed the suitability of existing technical concepts for an implementation of a coin-MoA allocation system. We presented an account-based solution giving a rough concept of a distributed and decentralized accounting web. The web is managed by all hosts executing MoA. Malicious behavior of one host is controlled by the hosts' community. The accounting web is designed to be open for MoA and hosts, but closed for (electronic) money.

As a result, continuous allocation of (digital) coins and MoA can be achieved. The accounting web also prevents double-spending and theft of coins as it allows an ex-post detection of cheating hosts. The detection is possible both for MoA principals after the agents have fulfilled their task as well as for other hosts according to an unbalance arising when, e.g., double-spent coins are used. Even MoA brainwashing (according to financial transactions) by malicious hosts may possibly be preventable by combining our concept with cryptographic trace methods.

Our research on accounting based allocation systems and PSMA is still in progress. Therefore some aspects/questions related to the application of accounting webs for MoA-coin allocation are not discussed within this paper. These include:

- The need for digital coins when the accounting web is finalized: is it possible to expand the allocation subsystem to a full PSMA without coins?
- The systems' behavior when hosts or groups of hosts crash: is the money also lost when a host disconnects due to a malfunction? What happens with the group if one host disappears?
- Malicious groups of hosts: how can we avoid groups consisting of only malicious hosts? And what happens if a malicious host is detected?
- The economic transaction recorded by the accounting web: actually only financial transactions are recorded. Are there any advantages when, e.g., the service executions are also journalized?

As next steps in the implementation of an account-based allocation system we will address concrete realization of both (1) accounts and (2) protocols for hosts and MoA interaction. While the first step has to take cryptographic methods and the data of the recorded entries into account, the second has to include specifications for hosts' and MoA entrance and exit of the accounting web, protocols ensuring consistent entries to accounts as well as for checking the accounting balance and rules about the consequences of detected unbalance.

In our future research, we will use the axiomatic-based accounting theory as a method to address these aspects and to advance the given rough concept to a more detailed one. For a final evaluation we plan to use a game theoretic analysis of the principals' behaviors affected by our concept.

6 References

[1] Abrazhevich, D.: Classification and Characteristics of Electronic Payment Systems. In *Proceedings of the 2nd International Conference on Electronic Commerce and Web Technologies (EC-Web'01)*. Springer, Heidelberg, LNCS 2115, 2001, 81-90.

[2] Algesheimer, J.; Cachin, C.; Camenisch, J.; Karjoth, G.: Cryptographic Security for Mobile Code. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy (S&P'01)*. IEEE Computer Society, 2001, 2-11.

[3] Al-Jaljouli, R.: Boosting m-Business Using a Truly Secured Protocol for Data Gathering Mobile Agents. In *Proceedings of the International Conference on Mobile Business (ICMoA'05)*. 2005.

[4] Athanassios, B.: Programming Sensor Networks with Mobile Agents. In Chrysanthis, P.; Samaras, G. (eds.): *Proceedings of the 6th International Conference on Mobile data management*. ACM Press, New York, 2005, 252-256.

[5] Bamaska, O.; Zhang, N.: A Secure Method for Signature Delegation to Mobile Agents. In *Proceedings of the 2004 ACM symposium on Applied computing (ACM SAC'04)*. ACM Press, 2004, 813-818.

[6] Biehl, I.; Meyer, B.; Wetzel, S.: Ensuring the Integrity of Agent-Based Computations by Short Proofs. In: Rothermel, K.; Hohl, F. (Eds.): *Mobile Agents*. Springer, Heidelberg, LNCS 1477, 1998, 183- 194.

[7] Bierman, E.; Cloete, E.: Classification of Malicious Host Threats in Mobile Agent Computing. In *Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology (SAICSIT'02)*. ACM International Conference Proceeding Series, 30, 2002, 141–148.

[8] Borselius, N.; Mitchell, C.; Wilson, A.: On Mobile Agent Based Transactions in Moderatly Hostile Environments. In *Proceedings of the 1st International IFIP Working Conference on Network Security (IFIP I-NetSec'01)*. Kluwer Academic Publisher, 2001, 173-186.

[9] Borchert, M.: *Geld und Kredit*. 7th edition, Oldenburg Verlag, Wien, 2001.

[10] Cachin, C.; Camenisch, J.; Kilian, J.; Müller, J.: One-Round Secure Computation and Secure Autonomous Mobile Agents. In *Proceedings of the 27th International Colloquium of Automata, Languages and Programming (ICALP'00)*. Springer, LNCS 1853, Berlin, 2000, 512-523.

[11] Chess, D.; Harrison, C.; Kershenbaum, A.: Mobile Agents: Are They a Good Idea? In: Vitek, J. (Eds.): *Mobile Object Systems - Towards the Programmable Internet*. Lecture Notes in Computer Science, No. 1222, Springer, 1997, 25-45.

[12] Claessens, J.: Analysis and design of an advanced Infrastructure for Secure and Anonymous Electronic Payment Systems on the Internet. Ph.D. Thesis, Katholieke Universiteit Leuven, 2002.

[13] Claessens, J.; Preneel, B.; Vandewalle, J.: (How) can mobile agents do secure electronic transactions on untrusted hosts? A survey of the security issues and the current solutions. *ACM Transactions on Internet Technologie, 3, 1* (2003), 28-48.

[14] Dadon-Elichai, A.: RDS: Remote Distributed Scheme for Protecting Mobile Agents. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents & Multi Agent Systems (AAMAS'04)*. Downloaded from http://www.aamas2004.org, June 20th, 2005.

[15] Das, A.: Payment Agents. In: Kou, W. (Eds.): *Payment Technologies for E-Commerce*. Springer, Berlin, 2003, 149-170.

[16] Endsuleit, R.; Mie, T.: Secure Multi-Agent Computations. In *Proceedings of International Conference on Security and Management (Sam'03)*. 2003, 149-155.

[17] European Central Bank: *Report on Electronic Money 1998*. Downloaded from https://www.ecb.int/pub/pdf/other/emoneyen.pdf, February 8th, 2006.

[18] Farmer, W.; Guttmann, J.; Swarup, V.: Security for Mobile Agents: Authentication and state appraisal. In *Proceedings of the 4th European Symposium on Research in Computer Security (ESORICS'96)*. Springer, LNCS 1146, Berlin, 1996, 118-130.

[19] Falchuk, B.; Karmouch, A.: The Mobile Agent Paradigm Meets Digital Document Technology: Designing for Autonomous Media Collection. In *Multimedia Tools and Application*. Springer, Volume 8, 1, 1999, 137-166.

[20] Ferreira, de Carvalho L.; Dahab, R.: Blinded-Key Signatures: securing private keys in mobile agents. In *Proceedings of the 2002 ACM symposium on Applied computing (ACM SAC'02)*. ACM Press, 2002, 82-86.

[21] Foukia, N.: Idream: Intrusion detection and response executed with agent mobility - the conceptual model based on self-organizing natural systems. In *Proceedings of the Engineering Self-Organising Applications Workshop (ESOA'04)*. AAMoAS conference, 2004.

[22] Fuggetta, A.; Picco, G. P.; Vigna, G.: Understanding Code Mobility. *IEEE Transactions on Software Engineering*, *24, 5* (1998), 342-361.

[23] Gunupudi, V.; Tate, S.: Performance Evaluation of Data Integrity Mechanism for Mobile Agents. In *Proceedings of 2004 IEEE Conference on Information technology: Coding and Computing (ITCC'04)*. IEEE Computer Society, 2004, 62-69.

[24] Hacini, S.: Using Dynamic Adaptability to Protect Mobile Agents Code. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*. IEEE Computer Society, 2005, 49-52.

[25] Henkel, J.: Anforderungen an Zahlungsverfahren im E-Commerce. In: Teichmann, R.; Nonnenmacher, M.; Henkel, J. (Eds.): *E-Commerce und E-Payment*. Gabler, Wiesbaden, 2001, 103-121.

[26] Hohl, F.: A Model of Attacks of Malicious Hosts Against Mobile Agents. In: *Proceedings of the 4th ECOOP Workshop on Mobile Object Systems (MOS'98)*. Springer, LNCS 1543, Berlin, 1998.

[27] Hohl, F.: *A Protocol to Detect Malicious Host Attacks by Using Reference States*. Report Nr. 09/99, Downloaded from ftp://ftp.informatik.uni-stuttgart.de/pub/library/ncstrl.ustuttgart_fi/TR-1999-09/TR-1999-09.pdf, October 10th, 2005.

[28] Hohl, F.: *Sicherheit in Mobile-Agenten-Systemen*. Ph.D. Thesis, Universität Stuttgart, 2001.

[29] Horlait, E., Magedanz, T.; Glitho, R.H. (Eds.): *Mobile Agents for Telecommunication Applications*. 5th International Workshop (MoATA 2003). Springer, Berlin, LNCS 2881, 2003.

[30] Jansen, W.A.: Countermeasures for Mobile Agent Security. In *Computer Communications. Special issue on advanced security techniques for network protection*. 23 (17), Elsevier Science, 2000, 1667-1676.

[31] Jarchow, H.-J.: *Theorie und Politik des Geldes*. 11th edition, Vandenhoeck & Ruprecht, Göttingen, 2003.

[32] Karmouch, A., Magedanz, T., Delgado, J. (Eds.): *Mobile Agents for Telecommunication Applications*. 4th International Workshop (MoATA 2002). Springer, Berlin, LNCS 2521, 2002.

[33] Kotzanikolaou, P.; Burmester, M.; Chrissikopoulos, V.: Secure Transactions with Mobile Agents in Hostile Environments. In *Proceedings of the Australasian Conference on Information Security and Privacy (ACISP'00)*. Springer, LNCS 1841, Berlin, 2000, 289-297.

[34] Krügel, C.; Toth, T.: *Applying Mobile Agent technology to Intrusion Detection*. Arbeitsbericht Nr. TUV-1841-2002-31. Downloaded from http://www.infosys.tuwien.ac.at/reports/bin/abstract.pl?report=TUV-1841-2002-31 , March, 8th 2004.

[35] Kaneda, T.; Tanka, Y.; Enkokido, T.; Takizawa, M.: Transactional Agent Model for Fault-Tolerant Object Systems. In *Proceedings of the 2005 ACM symposium on Applied computing*. ACM Press, New York, 1133-1138.

[36] Lee, H.; Alves-Foss, J.; Harrison, S.: The use of encrypted Functions for Mobile Agent Security. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*. IEEE Computer Society, 2004, 390297b.

[37] Loureiro, S.; Molva, R.: Function Hiding Based on Error Correcting Codes. In *Proceedings of the International Workshop on Cryptographic Techniques and Electronic Commerce (CryptTEC'99)*. 1999, 92-98.

[38] Ma, L.; Tsai, J.P.; Murata, T.: A Secure Mobile Agent System Model Based on Extended Elementary Object System. In *Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC'04)*. IEEE Computer Society, 2004, 218-223.

[39] Milojicic, D.: *Trend Wars: Mobile Agent Applications*. IEEE Concurrency, 6, 1999, 80-90.

[40] Minsky, Y.; Renesse, R.; Schneider, F.; Stoller, S.: Cryptographic Support for Fault-Tolerant Distributed Computing. In *Proceedings of the 7th ACM SIGOPS European Workshop: Systems Support for Worldwide Applications*. ACM Press, 1996, 109-114.

[41] Olougouna, E.; Pierre, S.; Glitho, R.: An Extensible Mobile-Agent-Based Framework for Coordinating Distributed Information Retrival Applications. In: Karmouch, A.; Magedanz, T.; Delgado, J. (Eds.): *Mobile Agents in Telecommunication 2002*, Springer, Berlin, LNCS 2521, 2002, 281-291.

[42] Page, J.; Zaslavsky, A.; Indrawan, M.: Countering Security Vulnerabilities in Agent Execution using a Self Executing Security Examination. In *Proceedings of the 3rd International Joint Conference of Autonomous Agents & Multi Agent Systems (AAMAS'04)*. IEEE Computer Society, 2004, 1486-1487.

[43] Pierre, S., Glitho, R. (Eds.): *Mobile Agents for Telecommunication Applications*. Third International Workshop (MoATA 2001). Springer, Berlin, LNCS 2164, 2001.

[44] Radu, C.: *Analysis and design of Off-line Electronic Payment Systems*. Dissertation, Katholieke Universiteit Leuven, 1997.

[45] Riordan, J.; Schneier, B.: Environmental Key Generation towards Clueless Agents. In Vigna, G. (Ed.): *Mobile Agents and Security*. Springer, LNCS 1419, Berlin, 1998, 15-24.

[46] Rivest, R.; Shamir, A.: PayWord and MicroMint: Two simple micropayment schemes. In *Proceedings of 1996 International Workshop on Security Protocols*. Springer, LNCS 1189, Berlin, 1997, 69-87.

[47] Romao, A.; Silva, M.: Secure Mobile Agent Digital Signatures with Proxy Certificates. In: Ye, Y.; Liu, J. (Eds.): *E-Commerce Agents*. Springer, Heidelberg, 2001, 206-220.
[48] Sander, T.; Tschudin, C.: Towards Mobile Cryptography. In *Proceedings of the 1998 IEEE Symposium on Security and Privacy (S&P'01)*. IEEE Computer Society, 1998, 215-224.
[49] Sander, T.; Young, A.; Yung, M.: Non-interactive CryptoComputing for NC. In: *40th Annual Symposium on Foundations of Computer Science (FOCS'99)*. IEEE Computer Society, 1999, 554-567.
[50] Schaal, P.: *Geldtheorie und Geldpolitik*. 4th edition, Oldenburg Verlag, Wien, 1998.
[51] Shamir, A.: How to share a secret. *Communications of the ACM*, 24, 11 (1979), 612-613.
[52] Shi, Y.; Cao, L.; Wang, X.: A Security Scheme of Electronic Commerce for Mobile Agents Uses Undetachable Digital Signatures. In *Proceedings of the 3rd International Conference on Information Security*. ACM Press, 2004, 242-243.
[53] Songsiri, S.: A New Approach for Computation Result Protection in the Mobile Agent Paradigm. In *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC'05)*. IEEE Computer Society, 2005, 575-581.
[54] Tan, H.K.; Moreau, L.: Certificates for Mobile Code Security. In *Proceedings of the 17th ACM Symposium on Applied Computing (ACM SAC'02)*. ACM Press, 2002, 76-81.
[55] Tate, S.R.; Xu, K.: Mobile Agent Security through Multi-Agent Cryptographic Protocols. In *4th International Conference on Internet Computing (IC'03)*. CSREA Press, 2003, 462-468.
[56] Thai, P.; Chamg, P.; Agha, G.: Crawlets: Agents for High Performance Web Searche Engines. In: Picco, G. (Ed.): *Mobile Agents*. Springer, Berlin, LNCS 2240, 2001, 119-2001.
[57] Velazquez, E; Santoro, N.; Nayak, A.: A Mobile Agent Prototype for Distributed Search. In: Pierre, S.; Glitho, R.(Eds.): *Proceedings of the Third International Workshop on Mobile Agents for Telecommunication Applications*. Springer, Berlin, LNCS 2164, 2001, 245-254.
[58] Vigna, G.: Cryptographic Traces for Mobile Agents. In: Vigna, G. (Eds.): *Mobile Agent Security*. Springer, Heidelberg, LNCS 1419, 1998, 137-153.
[59] Vigna, G.: Mobile Agents: Ten Reasons For Failure. In *Proceedings of the 5th IEEE International Conference on Mobile Data Management (MDM'04)*. IEEE Computer Society, 2004, 298-299.
[60] Wang, R.; Huang, H.; Wang, H.: Multi-Mobile Agents' Seperation Scheme in Java Card Application for Mobile Agent's Security. In *Proceedings of the 2004 IEEE International Conference on Service Computing (SCC'04)*. IEEE Computer Society, 2004, 623-628.
[61] Wagner, A.: Mobile Agent – Based Module Distribution in Heterogenous Networks. In: Stamatis, V. et al: *Proceedings of the first conference on Computing Frontiers*. ACM Press, 288-293.
[62] Weber, B.: *Zahlungsverfahren im Internet – Zahlungen mittels Kreditkarte, Lastschrift und Geldkarte.* Verlag Dr. Otto Schmidt, Köln, 2002.
[63] Wilhelm, U.; Staatmann, S; Buttyan, 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*. Springer, LNCS 1603, New York, 1999, 469-489.
[64] Yee, B.: A Sanctuary for Mobile Agents. In: Vitek, J.; Jensen, C. (Eds.): *Secure Internet Programming: Security Issus for Mobile and Distributed Objects*. Springer, LNCS 1603, New York, 1999, 261-274.
[65] Yang, M.; Huang, S.; Wang, Z.; Cheng, Z.; Mao, D.; Gao, C.: PICC: A Secure Mobile Agent Framework Based on Garbled Circuit. In *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)*. IEEE Computer Society, 2005, 357-362.
[66] Yang, Y.; Rana, O.; Georgousopoulos, C.; Walker, D.; Williams, R.: Mobile Agents and the SARA Digital Library. In: Papazoglou, M.; Sheth, A.: *Seventh IEEE Advances in Digital Libraries 2000 (ADL '00)*. 2000, 71-78.
[67] Zhicai, S.; Zhenzhou, J.; Mingzeng, H.: A Novel Distributed Intrusion Detection Model Based on Mobile Agent. In: White, J.; Sheng, H.: *Proceedings of the 3rd international conference on Information security. ACM Press*, New York, 2004, 155-159.
[68] Zhong, S.; Yang, Y.R.: Verifiable Distributed Oblivious Transfer and Mobile Agent Security. In *Proceedings of the 2003 Joint Workshop on Foundations of Mobile Computing (DIALM-POMC'03)*, ACM Press, New York, 2003, 12-21.