# Spy Agents: Evaluating Trust in Remote Environments

Georgios Kalogridis*, Chris J. Mitchell†, and Gary Clemo*
*Toshiba Research Europe Limited
Telecommunications Research Laboratory, 32 Queen Square, Bristol, BS1 4ND, UK
E-mail: {georgios.kalogridis, gary.clemo}@toshiba-trel.com
†Information Security Group
Royal Holloway, University of London, Egham, Surrey, TW20 0EX, UK
E-mail: c.mitchell@rhul.ac.uk

*Abstract— We introduce the notion of spy agents and describe how they can be deployed within diverse network protocol architectures in order to perform high fidelity trust assessments in remote environments. The spy agent framework developed here consists of: a spy agent structural architecture that instruments and instantiates spy agents with appropriate content; a spy agent routing framework that fabricates and deploys the overall spying scenario with specialised spying routing protocols; and an evaluation entity that implements all the necessary security analysis mechanisms.*

*Keywords—Spy agents, trust evaluation, surveillance*

## 1. Introduction

Mobile agents are the basis of a distributed programming infrastructure with numerous inherent potentially beneficial characteristics such as autonomy, flexibility and intelligence [1]. One typical example of an application is a price comparison agent which "visits" a number of on-line retailer sites or nodes and requests a price for a particular item. The agent could retrieve and process information, including for example prices, from a number of different retailers.

However, there are still challenges that need to be addressed, including mobile agent security [2]. As is widely discussed (see, for example, [2], [3]) there are two parallel sets of security issues associated with mobile agents, namely protecting hosts (and other agents) against malicious agents, and protecting agents against malicious hosts.

Legitimate mobile agents will interact with hosts in a defined way, and hosts will be built to deal with expected agent behaviour. This contrasts with viruses and other "illegitimate" agents, which may attempt to access the host itself rather than remain in the execution environment reserved for agents (e.g. a sandbox). Such malicious agents can then steal sensitive information from the host, for example personal financial details, cause the host to act in an unintended way, for example send spam emails, or simply corrupt the host so that it no longer functions properly.

The parallel security problem, and the one that is the main focus of this paper, is that agents are at the mercy of the host which executes them, as ultimately the host may either carry out the functions requested by the agent as expected, or it may manipulate the agent. Such manipulation might include reading data contained within the agent which is intended to remain private, e.g. the source address or the identity of the agent originator. This information can then be misused for a variety of purposes, including forwarding spam to the originator's email address. Other examples of inappropriate behaviour might include reading quotes from competitor on-line retailers and providing a more attractive quote, or changing the other quotes to make them less attractive.

Autonomous mobile agents, apart from obtaining price quotes or retrieving other information for further analysis, might also be able to perform a transaction remotely and completely autonomously based on the client's instructions. For

example, to get a cheap air ticket automatically, an agent might be instructed to visit several online stores in order to make a purchase. The client may wish to give third parties certain personal information, embedded in the agent, only if there is a considerable discount on the final price. The host should never read or modify the agent's transaction logic, or the private data that the agent will carry; however there is clearly a possibility for such abuse.

In this paper we address this latter security issue, i.e. the threat posed by malicious hosts. We introduce a new kind of security service designed to indirectly improve the overall security level by means of accurate and powerful evaluation mechanisms that pre-emptively assess the security of remote hosts, before they are sent vulnerable mobile agents. We introduce the notion of spy agents as "legitimate" mobile agents which are able to interact with remote, potentially hostile, mobile agent platforms in a manner that facilitates trust assessment.

We further introduce a spy agent framework, and we analyse the importance and the value that this could have in a number of situations.

## 2. Background

Surveys of mobile agent security issues can be found in [2], [3]. The two main actors in a mobile agent system are the following:

- Agent: An instance of mobile code
- Host: A platform that can execute agents

As mentioned in the introduction, in this paper we consider the threat posed to agents by malicious hosts; this is an important research area and a plethora of solutions have been proposed (tamper-proof hardware [4], tamper-proof execution environment [5], code obfuscation [6], encrypted functions [7], [8], strategic division of functionality across multiple agents [9], etc). Nevertheless, none of the existing solutions is able to address the problem in both a practical and robust manner, because they either depend on hardware modules, or have unresolved technical problems, and/or depend too much on trust assumptions and implied policies.

Instead of trying to overcome these issues with some direct security mechanisms, we propose a robust trust evaluation mechanism that can indirectly provide mobile agents with the required security. This can be leveraged by special mobile agents that will bear security services enabling them to retrieve and process security related information from target hosts. This concept is introduced in [10], which suggests that a security assessment can be facilitated simply when an agent migrates from a trusted platform to a target platform, where it gets certain information and then returns to the trusted host for further analysis.

Ideas relevant to the notion of spy agents can be found in the field of internet security monitoring services. Strategically placed control mechanisms can perform service monitoring, [11], [12], including monitoring of enhanced IP services and virtual private network (VPN) packets. Similarly, in distributed intrusion detection systems, [13], [14], control agents may protect a domain by "interrogating" suspicious agents.

## 3. The challenge

The idea of a distributed security system comprising security agents patrolling target entities, modelling and monitoring behaviours and potentially taking steps in advance, is not novel. However, in previous approaches [11]–[13], [15] the idea is that such security mechanisms or agents operate in trusted environments and cannot be applied in remote, potentially hostile, areas.

In [10] the authors attempt to approach the challenge of remote trust evaluation; however there are major limitations in their approach. The main problem is that it is assumed that the target hosts will adhere to their policies and will provide the agents with all the security assessment information they request. Hence, a potentially corrupted remote host could serve these security agents by providing them with apparently proper information. The host could subsequently not behave in an appropriate manner when it has the opportunity to cheat without being detected and, vice versa, in order to escape detection, it may modify its behaviour if it knows it is being monitored.

By contrast with previous work, our proposed spy agents are designed to be able to fetch security information that truly reflects the genuine character of any remote host.

## 4. The spy agent concept

The main idea behind a spy agent is that it provides the means to evaluate trust in remote en-

vironments without the target hosts knowing that they are being assessed. In this way, spy agents have the ability to determine the target hosts' genuine behaviour, i.e. the degree to which a host complies with its policies or, more specifically, with its responsibility to respect client security requirements.

A significant part of a spy agent's task is the extraction of security-related information from remote hosts, without violating the remote host's policy or security protection mechanisms. This means that a spy agent should retrieve the information it needs from remote hosts in a legitimate way and with the explicit permission of the host concerned; at the same time this must happen in a way that does not reveal the spy agent's true objectives. The information retrieved by a spy agent will not necessarily be directly security-related; it simply needs to be information that can be used in some way to assess host behaviour, possibly when combined with information retrieved by one or more other such agents.

In all cases target hosts should be unable to exhibit special behaviour in order to give a good impression. In brief, the following requirements spying requirements are identified:

- Target hosts should be incapable of deciding whether they are dealing with a spying scenario or not;
- Spy agents should appear as "normal" m-commerce ones;
- Target hosts should be given motives to misbehave by using spy agents as "baits";
- Feedback should be analysed in a safe environment.

# 5. Spy agent system architecture

In general terms a spy agent system disseminates a number of assessment spy agents to a target network node in an insecure network, and retrieves these agents following interaction with the node. As it will be further discussed in section 6, a large number of spy agents can facilitate more enhanced cross-referenced analyses.

One of the most basic requirements is to provide spy agents with anonymity. This is achieved by arranging the agents so that they can be associated with different sources or transmitters. That is, spy agents can be forwarded to a plurality of trusted nodes in the network, which each modify the received agent's code in order to show the trusted node as the source of the agent, before forwarding the agent towards the target node. This is shown in figure 1.

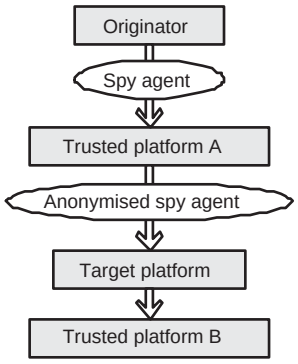


*Figure 1: Fundamental spy agent system components*

For reasons that will be analysed in more detail below, a spy agent should ideally give away as little information as possible. In this context it is proposed that the modified spy agent should ultimately be destined for a trusted node different to the node that sends it to the target host. This second trusted node will be notified by the first trusted node to expect a particular spy agent, and on receiving the agent will be able to forward it back to its origin.

When all spy agents return to their original source, the spy agent system will then proceed to analyse their interactions with the target node, in order to determine a trust level for it.

## 5.1 Spy agent content framework

A schematic of a software spy agent is shown in figure 2. The agent includes an agent ID as well as an origin or source ID field, a final destination ID field, a number of intermediate node IDs, and a payload. The payload includes personal data such as a name, address, email address, various certificates, financial information, and other information associated with a person or client. Finally, the agent includes executable code.

The spy agent's private data, such as ID information, email address, public key certificates, and so on, should preferably correspond to a temporary entity that a mobile platform sets up in a legitimate manner. Hence they should represent a virtual client that appears to be a normal client,

and its relationship with the real client are hidden by the system. In this way it can be ensured that the target hosts behave in exactly the same way as they would behave to a real e-commerce agent scenario. In order to be able to disguise a spy agent in this way, the spying system needs to interact and cooperate in a secure manner with online services such as e-mail providers, certification authorities and banks, who need to be aware of the purposes of the spy agent and be prepared to support them. These relationships are shown in Figure 3.

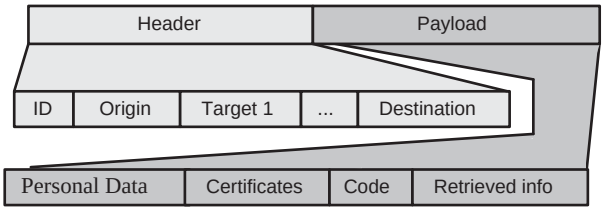


*Figure 2: Spy agent internal structure*

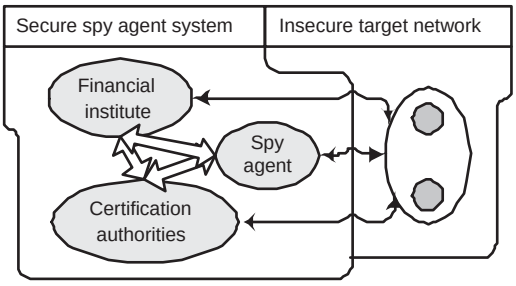


*Figure 3: Spy agent system architecture for collaboration with other trusted parties*

For example, the creator of an assessment agent might want to set up a temporary email address, or request a certificate from a certification authority for temporary use in assessing a host. This certificate need not allow an agent to perform any transaction automatically since it will be temporary. However the target platforms will not be aware of this, and should believe that the agent will be equipped with all the "normal" functions of an m-commerce agent. That is, it must appear to be just another commerce agent that could, if it wishes, decide to complete a transaction.

Overall the preferred structure of a spy agent should be the most commonly used such scheme. Apart from supporting interoperability, the spy agent will also look "normal" and "common" and this minimises the probability of making target hosts suspicious. This satisfies the overall security objective behind the spy agent concept.

## 5.2 Spy agent routing framework

One of the most important aspects of the spy agent system is the routing mechanism that determines the specifics of the spy agents' distribution logic and influences the content of each agent, as described in the previous paragraph. The need for a routing logic arises from the complexity of dealing with multiple agents, issued from a large number of trusted nodes, and being routed over different paths. This becomes necessity since the kind of information exchanged between all the spy agents and an arbitrary target node, including routing information, can determine how well the spy agent requirements, outlined above, are met. The routing scheme should effectively disguise the fact that spy agents originate from a specific client device and are in any way related.

A migration path is the chain of all the platforms that an agent visits during its life (starting from a trusted platform). Thus all migration information included in an assessment spy agent, should not contain (or should minimise the) common elements between peer spy agents that reach the same target platform. This minimises the likelihood of a target platform linking the two spy agents and becoming aware of the spying scenario. Note that the trusted platforms could, for example, be the same mobile terminal, a home computer, or, preferably, random public servers set up for that purpose (this might come at an increased network and end user cost).

Ideally, not making the target hosts suspicious requires that all the information that target hosts retrieve from spy agents is completely uncorrelated. In order to achieve this it is necessary to know the nature of all the inter-relationships between those target hosts that at least one of the spies is going to visit. For example, a typical case is where it can be assumed that all target hosts are uncorrelated, and do not share any information or cross-reference intelligence information. In this case an efficient spy agent routing design requires any two spies to be uncorrelated, only if these two spies eventually visit one common target host. If these two spies do not have a common target host in their migration logic, then it is unnecessary to try to design them to appear to be uncorrelated.

Additionally, it is proposed that the assessment agents should be designed to route themselves through two or more target hosts rather than just one. Otherwise, if a target host receives an agent that insists on migrating to an unknown server (without migrating, for example, to a known competitor), it will have a good reason to refrain from behaving badly (either because it believes that this incoming agent might be an assessment agent or because it cannot see any direct competition). Thus a normally misbehaving server or target platform might behave correctly (just in case), and subsequently the evaluation results will not meet the system objectives. This is because the server may not react similarly when, for example, the incoming agent requests to migrate to a well-known rival service provider. Moreover, the mobile device will not be able to repeat assessment procedures because the host may assess these incoming agents as probably being spy agents, assuming that it keeps records of past events and makes statistical analyses and comparisons.

Some of the discussed routing strategy principles are demonstrated in the example spying routing scenario shown in Figure 4. In this example, a mobile device evaluates three target platforms. The mobile device uses three trusted platforms in order to set up its distributed routing strategy, as well as maintain its anonymity. The device instantiates six mobile agents, separated into two matching groups of three:

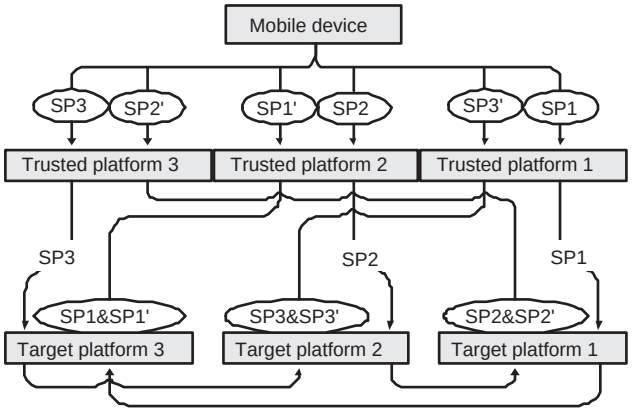


*Figure 4: Testing three target platforms with the aid of three trusted platforms and six spy agents*

- The first three spies start their journey from a single trusted platform and then migrate to two target platforms in turn. The spy agents do not contain any logic dictating where they should migrate after visiting a second target platform.
- The three spies in the second group, which are called guidance spy agents and are in one-to-one correspondence with the spies in the first group, contain migration information for the corresponding agents of the first group. They visit the platform where the corresponding spy is waiting to be instructed where to go next. With this technique the spy agent's anonymity can be further enhanced.

In this scenario it is suggested that two agents executing in a remote host may intercommunicate. It is assumed that this is allowed only between two mobile agents that have been instructed to do so. Hence, agent intercommunication will only be allowed between a spy agent and its respective guidance spy agent. It is suggested that mobile agents may employ cryptographic techniques that will thwart a potential security breach of this requirement. For example standard cryptographic protocols can be used for mutual authentication and integrity (see for example, Boyd and Mathuria [16]).

In more detail, the first spy agent, $SP1$, (see Figure 4) leaves from the first trusted platform, $TrP1$, visits the first target platform, $TaP1$, migrates to $TaP3$, and waits to meet the first guidance spy agent, $SP1'$, coming from $TrP2$. Similarly $SP2$ starts from $TrP2$, migrates to $TaP2$, then $TaP1$, and waits for further instructions from $SP2'$ coming from $TrP3$. In a symmetrical fashion, $SP3$ waits in $TaP2$ for guidance. Finally, $SP1$ is instructed to return to $TrP2$, $SP2$ to return to $TrP3$, and $SP3$ to return to $TrP1$.

Examining $TaP1$, it can be noted that it hosts $SP1$, $SP2$ and $SP2'$. Hence, $TaP1$ is able to retrieve the following information regarding the migration paths for $SP1$, $SP2$ and $SP2'$:

- $SP1 : (TrP1, TaP1, TaP3)$
- $SP2 : (TrP2, TaP2, TaP1, TrP3)$
- $SP2' : (TrP3, TaP1, TrP3)$

It is clear that $TaP1$ cannot correlate the routing behaviour of $SP1$ and $SP2$, since the only common element in their routes is itself, $TaP1$. It can observe that the routes of $SP2$ and $SP2'$ share $TrP3$, but this does not look suspicious since $SP2$ is expecting guidance from $SP2'$. In a

symmetrical fashion, possible routing correlations for $TaP2$ and $TaP3$ lead to similar conclusions. Hence it is argued that this routing design satisfies the basic spying requirements.

An obvious potential weakness of this routing design is the fact that all target platforms could learn about the spying scenario if they exchanged information. However, this weakness can be avoided by employing more trusted platforms and designing the routes for each agent in such a way that the risk of platform collusion revealing spy agent behaviour is minimised.

## 6. Trust evaluation

The final assessment of the trustworthiness of a target platform could yield estimates for a multitude of security issues, such as the probability of the host reading or altering private data that should never be accessed, blocking or diverting migration, etc.

These assessments could be achieved in a variety of ways, for example by comparing the data retrieved by the various agents with that obtained by agents using different routes. Also the returned agents could be examined to see if they have been altered in any way other than in terms of their retrieved data — this might include blocking or changing a migration route. For example, the agents might contain a temporary email address which can be monitored to determine if spam emails will be sent in the future. If unsolicited emails are received at this address, then one of the hosts visited by the agent containing this address may be suspected of having violated its policy, e.g. by reading private data in the agent. Similar techniques can be implemented in order to provide a trust level for a number of hosts.

By using multiple agents, the gathered information can be cross-referenced, and more accurate predictions can be made, since each spy agent visits more than one target host. Hence, if an attack is detected it will be possible to investigate it and identify a suspect target platform with greater certainty. On the other hand, if an unmodified agent is normally returned without delay, then it can be assumed that with high probability all visited target platforms have behaved properly.

For example, consider the multiple-agent scenario shown in Figure 4. Assuming that the target platforms demonstrate their genuine behaviour and do not modify their behaviour to avoid detection (e.g. if they suspect the purpose of one or more of the spy agents), then this protocol architecture enables us to make deductions along the following lines. Suppose that only $SP1$ and $SP2$ have been tampered with, but not any of the other four agents. Then since $SP1$ visited $TaP1$ and $TaP3$ and $SP2$ visited $TaP2$ and $TaP1$, it seems that $TaP1$ is more likely to have misbehaved. Generally it is envisaged that more detailed and fine-grained conclusions can be drawn by examining all the spies at the end of their routes, as long as it is possible to predict to what degree malicious target platforms can modify their behaviour to avoid detection.

## 7. Implementation issues

A spy will attempt to encourage hosts to misbehave, in order to obtain the most accurate assessment of their genuine behaviour. Furthermore, the very existence of assessment agents will generally mitigate the existence of malicious service providers, since they will be unable to distinguish between spy agents and normal e-commerce agents. Hence, it will not be possible for them to know if security policies can be breached without being detected.

Generally, the design requirement to prevent target hosts from determining whether or not they are dealing with a spy agent and hence behave according to their genuine character, contradicts with the requirement to make more precise assessments about each one of the target platforms. Deterministic assessments can be made by sending a number of spy agents to just one target host, but such a strategy is likely to raise suspicion in the target host, resulting in atypical (correct) behaviour by malicious target hosts. On the other hand, spy agents that migrate through a large number of competitors can provide confidence that their real-life behaviour is exhibited. However, in this case assessments of any individual host would be rather uncertain.

It is apparent that the optimum protocol architecture should try to balance the pros and cons in each case and mount the best winning strategy. However the precise optimal strategy will be highly variable and will depend on how many

trusted devices a mobile terminal has, what the computational expenses are in each case, what the objective of the analysis is (e.g. to maintain high quality security profiles; to perform an ephemeral test), when the results are needed (e.g. immediately or within a fixed time period), how desperate the terminal is for very accurate results, and how much it is willing to pay for these results.

## 8. Conclusions and future work

This paper introduces the concept of spy agents, justifies the benefits that can be gained from utilising them, and describes a generic structural and routing protocol that leverages the idea of disguising the identity of a spy agent.

The focal point of the proposed framework is the manner with which spy agents can be coordinated in order to fetch network information, from which probabilistic conclusions about the security of a number of network entities can be drawn. The success of a spying scenario depends on the amount of cross-referenced retrieved information, provided that the fundamental spying security requirements are met. Thus it is proposed to use a plurality of spy agents and a plurality of trusted platforms that will coordinate the spy agents' movements, while hiding the spies' identities and the way they are associated.

Spy agent networks could offer a very responsive and reliable security service to end terminals, the price of which will depend on the reliability of the service. Assessments of high quality could be further exploited by other applications in order to adapt their security to the existing circumstances as well as control the overall risk in a fine-grained manner. This methodology is ideal for remote surveillance and risk assessment.

In future work it is necessary to instrument metrics, which will quantify how good a specific routing protocol is, and design mechanisms that will deploy efficient spying routes for specific scenarios. Ultimately, it should be possible not only to evaluate a target host, but also to determine what will be the exact benefit of utilising a specific number of resources within a specific spying scenario, from a security point of view.

## References


[1] C. G. Harrison, D. M. Chess, and A. Kershenbaum, "Mobile agents: Are they a good idea?" IBM Research Division, T.J. Watson Research Center, Yorktown Heights, NY, Technical Report, March 1995.

[2] W. Jansen and T. Karygiannis, "Mobile agent security," National Institute of Standards and Technology, NIST Special Publication 800-19, August 1999.

[3] N. Borselius, "Security for agent systems and mobile agents," in *Security for Mobility*, C. J. Mitchell, Ed. IEE, London, 2004, ch. 12, pp. 287–303.

[4] B. S. Yee, "Using secure coprocessors," Ph.D. dissertation, Carnegie Mellon University, 1994.

[5] C. Wang, J. Hill, J. Knight, and J. Davidson, "Software tamper resistance: Obstructing static analysis of programs," Department of Computer Science, University of Virginia, CS Techical report CS-2000-12.

[6] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang, "On the (im)possibility of obfuscating programs," in *Advances in Cryptology, CRYPTO*, U. G. Wilhelm, L. Buttyàn, and S. Staamann, Eds. Springer-Verlag, 2001, pp. 1–18.

[7] J. Riordan and B. Schneier, "Environmental key generation towards clueless agents," in *Mobile Agents and Security*, G. Vigna, Ed. Springer-Verlag, 1998, pp. 15–24, lecture Notes in Computer Science No. 1419.

[8] T. Sander and C. Tschudin, "Towards mobile cryptography," in *IEEE Symposium on Security and Privacy*, Oakland, CA, May 1998, pp. 215–224.

[9] S. K. Ng and K. W. Cheung, "Protecting mobile agents against malicious hosts by intention spreading," in *Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)*, H. Arabnia, Ed., vol. II, 1999, pp. 725–729.

[10] N. Borselius, C. J. Mitchell, and A. T. Wilson, "On mobile agent based transactions in moderately hostile environments," in *Advances in Network and Distributed Systems Security, Proceedings of IFIP TC11 WG11.4 First Annual Working Conference on Network Security*, B. D. Decker, F. Piessens, J. Smits, and E. V. Herreweghen, Eds. Kluwer Academic Publishers, Boston, November 2001, pp. 173–186.

[11] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "A simple network management protocol (snmp)," IETF, Manual RFC 2401, May 1990.

[12] M. Günter and T. Braun, "Internet service monitoring with mobile agents," *IEEE Network Magazine*, vol. 16, no. 3, pp. 22–29, May / June 2002.

[13] P. C. Chan and V. K. Wei, "Preemptive distributed intrusion detection using mobile agents," in *11th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE Computer Society, 2002, pp. 103–108.

[14] E. H. Spafford and D. Zamboni, "Intrusion detection using autonomous agents," *Computer Networks*, vol. 34, no. 4, pp. 547–570, 2000.

[15] D. Dasgupta and H. Brian, "Mobile security agents for network traffic analysis," in *DARPA Information Survivability Conference and Exposition II (DISCEX-II)*. Anaheim, California: IEEE Computer Society Press, June 2001, pp. 12–14.

[16] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*. Springer-Verlag, 2003.