Untraceability of Mobile Agents

Rafał Leszczyna[1] and Janusz Górski[2]

[1] European Commission, Joint Research Centre,
Via Enrico Fermi, 21020 Ispra (VA), Italy
[2] Gdansk University of Technology,
Narutowicza 11/12, Gdańsk, Poland

10 December 2004

Abstract

In the article we present two untraceability protocols for mobile agents. Comparing to other solutions, the advantage of the protocols is that they support agent's autonomy in choosing the migration path.

1 Introduction

Anonymity is the property that ensures that a user may use a resource or service (the items of interest) without disclosing his/her identity [1]. *Untraceability* is a subclass of communication anonymity [2] assuring that the identity can not be inferred by tracing the message.

Anonymity plays a crucial role for various activities conducted in the Internet. For example, in health counselling, a patient suffering from an embarrassing disease or from an addiction may wish to stay anonymous when asking for an advice [3]. Gulcu et al [4] describe four categories of the Internet applications where the anonymity is required[1]. These are: discussion of sensitive and personal issues, information searches, freedom of speech in intolerant environments and polling/surveying. For some of these uses, the anonymous access was an enabler and contributed to increasing the popularity of the Internet.

We proposed the untraceability protocol for mobile agents – the *Protocol I* and investigated its security [5]. The study showed that the protocol, while having the advantage of being effective, is prone to the costly, nonetheless possible to perform – *cordoning-off attack* – i.e. the attack in which the attacker compromises all platforms surrounding the source one [5]. Consequently we introduced the *Protocol II* which eliminates this vulnerability at the cost of putting more computation workload on the source platform and restricting agent autonomy in the beginning of its route. Hereby we gave choice to the users: they can use more effective but slightly less secure Protocol I, or choose the Protocol

[1] This is important to note that the authors don't claim this set to be exhaustive.

II, to receive stronger protection. In this paper, due to space limitations, we present only the details of the first protocol and just sketch the idea of the latter. Comparing to other solutions [6, 7], the advantage of the protocols is that they support agent's autonomy in choosing the migration path.

2 Protocol assumptions

It is assumed that a platform guarantees that third parties (other agents, users) are not informed about the presence of other agents if the latter do not want to do so. It means that it is impossible to introduce any agents aiming at observing and following other agents. It is also assumed that each platform is well isolated, so it is impossible to learn its state from outside. Each platform owns an individual symmetric key and a private asymmetric key. A platform must also have access to all needed public keys of other platforms. However, the actual implementation of the key management is out of scope of this article. Each platform stores the identifier of the previous platform visited by an agent until the agent leaves out. This identifier is available to the agent.

3 The Protocol I

The *Protocol I* is dedicated to obfuscate the identity of an agent source platform while keeping the agent capable to autonomously select the next platform to visit. The protocol is intended to be as little resource consuming as possible. Encryption is employed only when necessary and only to an essential content.

Generally, the idea of the protocol is that while migrating, the agent encrypts the identifier of the last visited platform (using the public key of the present platform) and puts it to the LIFO queue stored in the agent. After achieving the goal, when the agent wishes to come back to its base platform, it uses the queue to find its way back. Down the route back the identifiers are subsequently decrypted using each platform's private key.

The pseudo-code of the protocol is presented in Listing 1. The variable m is used only to illustrate subsequent steps of algorithm and is not stored explicitly in the agent state. Storing the counter of visited platforms in the agent state would allow the attacker to identify the base platform, since he/she could always read how many platforms are left to the base platform. If the attacker located himself/herself right next to the base platform, it could easily recognize that the preceding platform is the source platform of the agent. This doesn't take place in the proposed protocol. Even very close to the base platform attacker can not recognize the situation. For the same reasons the LIFO queue of platform identifiers is initially filled with a number of random values. The notation $(\mathsf{B}_1, \mathsf{B}_2, \ldots, \mathsf{B}_\mathrm{n})$ represents the binary concatenation of the values $\mathsf{B}_1, \mathsf{B}_2, \ldots, \mathsf{B}_\mathrm{n}$. The nonce N_k is used to assure uniqueness of obtained values.

Listing 1: The basic protocol pseudo-code.

1. The LIFO queue of encrypted platform identifiers (stored in the agent's state) is initially filled with a number of random values
2. m=2
3. The agent moves to the platform AP_m
4. The agent processes its task on the platform AP_m
5. If mission accomplished then go to 12
6. The agent decides which platform to visit next (AP_{m+1})
7. The platform AP_m computes the hash value of the predecessor identifier ID_{m-1}, the own identifier ID_m and the successor identifier ID_{m+1} obtaining $H(ID_{m-1}, ID_m, ID_{m+1})$
8. The platform AP_m encrypts the identifier ID_{m-1}, the hash value (ID_{m-1}, ID_m, ID_{m+1}) and the nonce N_m with its secret key K_m obtaining $K_m(ID_{m-1}, H(ID_{m-1}, ID_m, ID_{m+1}), N_m)$
9. The agent adds $K_m(ID_{m-1}, H(ID_{m-1}, ID_m, ID_{m+1}), N_m)$ to the end of the queue of encrypted platform identifiers
10. m=m+1
11. Go to 3

// Returning to the source (AP_1) from the last platform on the route (AP_n):

12. m=m-1
13. The agent moves to the platform AP_m
14. If $AP_m == AP_1$ then finish()
15. The platform AP_m takes out (and does not put it back later) the first encrypted platform identifier available from the LIFO queue of encrypted platform identifiers
16. If the queue was not compromised the taken part should be the one encrypted with the secret key K_m of the platform, if it is not then go to 20
17. The platform decrypts the encrypted platform identifier and obtains ID_{m-1}, $H(ID_{m-1}, ID_m, ID_{m+1})$ and the nonce
18. The platform verifies the hash value $H(ID_{m-1}, ID_m, ID_{m+1})$
19. If the verification does not fail then go to 12
20. Perform the emergency scenario // The verification of hash value has failed - the string of encrypted platform identifiers was compromised

---

4 The Protocol II

The *Protocol II* was designed as an extension of the basic protocol to make it resistant to the cordoning-off attack. In the protocol, the source platform arbitrarily chooses a particular number of platforms the agent has to visit initially and creates the list of their encrypted identifiers. This serves as the initial route letting the agent to obfuscate its source address. After leaving this initial route, the agent is free to make decisions about which platforms to visit next. It autonomously roams the network to achieve its goal and after succeeding returns to the last platform of the initial route. Then, to come back to its source platform, it must follow the initial route in the reverse order. In the extended

protocol, only the last platform of the route knows the destination address but it is not able to recognize the source address. The first platform knows the source address but without the knowledge of being the first in the route (it could be just another platform on the route).

5 Summary

We presented two untraceability protocols, which unlike other solutions support autonomy of agents migration. The Protocol I assures more balanced distribution of processing over all agent platforms but an attacker can compromise untraceability if he/she manages to perform the costly cordoning-off attack. The Protocol II eliminates this vulnerability at the cost of putting more computation workload on the source platform and restricting agent autonomy in the beginning of its route. In the next step of the research we plan to implement and validate experimentally the protocols upon the JADE platform. We also aim at performing a more extensive study of the performance of the protocols.

References

[1] National Institute of Standards and Technology (NIST). *Common Criteria for Information Technology Security Evaluation - Part 2: Security Funtional Requirements*. U.S. Government Printing Office, 1998.

[2] Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity - a proposal for terminology. draft v0.21. 2004.

[3] EU IST-2002-507591 PRIME. Requirements version 0 part 3: Application requirements.

[4] Ceki Gulcu and Gene Tsudik. Mixing email with Babel. In *Proceedings of the 1996 Symposium on Network and Distributed System Security (SNDSS '96)*, page 2. IEEE Computer Society, 1996.

[5] Rafał Leszczyna and Janusz Górski. Untraceability of mobile agents. Technical report, European Commission, Joint Research Centre, Institute for the Protection and security of the Citizen, December 2004.

[6] Dirk Westhoff, II Markus Schneider, Claus Unger, and Firoz Kaderali. Protecting a mobile agent's route against collusions. In *Proceedings of the 6th Annual International Workshop on Selected Areas in Cryptography*, pages 215–225. Springer-Verlag, 2000.

[7] M. Enzmann, T. Kunz, and M. Schneider. Using mobile agents for privacy amplification in the trade with tangible goods. In *6th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI)*, volume IV, Orlando, Florida, USA, July 2002.