Undetachable Threshold Signatures*

Niklas Borselius, Chris J. Mitchell, and Aaron Wilson

Mobile VCE Research Group, Information Security Group,
Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK

Abstract. A major problem of mobile agents is their inability to authenticate transactions in a hostile environment. Users will not wish to equip agents with their private signature keys when the agents may execute on untrusted platforms. Undetachable signatures were introduced to solve this problem by allowing users to equip agents with the means to sign signatures for tightly constrained transactions, using information especially derived from the user private signature key. However, the problem remains that a platform can force an agent to commit to a sub-optimal transaction. In parallel with the work on undetachable signatures, much work has been performed on threshold signature schemes, which allow signing power to be distributed across multiple agents, thereby reducing the trust in a single entity. We combine these notions and introduce the concept of an undetachable threshold signature scheme, which enables constrained signing power to be distributed across multiple agents, thus reducing the necessary trust in single agent platforms. We also provide an RSA-based example of such a scheme based on a combination of Shoup's threshold signature scheme, [1] and Kotzanikolaou et al's undetachable signature scheme, [2].

1 Introduction

A digital signature is the electronic counterpart to a written signature. Thus one way to commit to an electronic transaction is by the use of a digital signature. Recently, the use of mobile agents to commit to transactions for a user has become a topic of interest. Mobile agents, however, face the problem of having to execute in a hostile environment where the host executing the agent has access to all the data that an agent has stored (for instance the private signature key). Consequently, the problem of allowing an agent to sign a transaction on behalf of a user is one of interest.

Undetachable signatures were first proposed by Sander and Tschudin [3] to solve this problem, and are based on the idea of computing with encrypted functions. The host executes a function $s \circ f$, where f is an encrypting function, without having access to the user's private signature function s. The security of the

* The work reported in this paper has formed part of the Software Based Systems area of the Core 2 Research Programme of the Virtual Centre of Excellence in Mobile & Personal Communications, Mobile VCE, `www.mobilevce.com`, whose funding support, including that of the EPSRC, is gratefully acknowledged. More detailed technical reports on this research are available to Industrial Members of Mobile VCE.

B. Honary (Ed.): Cryptography and Coding 2001, LNCS 2260, pp. 239–244, 2001.
© Springer-Verlag Berlin Heidelberg 2001

method lies in the encrypting function f. Whilst Sander and Tschudin were unable to propose a satisfactory scheme, more recently Kotzanikolaou, Burmester and Chrissikopoulos [2] have presented an RSA-based scheme which appears to be secure.

The idea of an undetachable signature is as follows. Suppose a user wishes to purchase a product from an electronic shop. The agent can commit to the transaction only if the agent can use the signature function s of the user. However as the server where the agent executes may be hostile, the signature is protected by a function f to obtain $g = s \circ f$. The user then gives the agent the pair (f, g) of functions as part of its code. The server then executes the pair (f, g) on an input x (where x encodes transaction details) to obtain the undetachable signature pair

$$f(x) = m \text{ and } g(x) = s(m).$$

The pair of functions allows the agent to create signatures for the user whilst executing on the server without revealing s to the server. The parameters of the function f are such that the output of f includes the user's constraints. Thus m links the constraints of the customer to the bid of the server. This is then certified by the signature on this message. The main point is that the server cannot sign arbitrary messages, because the function f is linked to the user's constraints.

However, one problem with this approach is that the agent is still given the power to sign any transaction it likes, subject to the requirement that the transaction must be consistent with the constraints used to construct f. Thus, for example, whilst the constraints may limit the nature and/or value of a transaction, a malicious host may force an agent to commit to a transaction much less favourable than could be achieved.

Thus, to protect further against malicious hosts, a user may wish to use more than one agent and have the agents agree on a bid before committing to it. Hence, a user may send out n agents with the criteria that k of them must agree before committing to a purchase. The obvious solution to such a requirement is to employ a *threshold signature scheme*, meaning that agents can all sign the bid they think 'best' given the user's requirements, and then, on receipt of a sufficient number of these bids, the user's signature can be reconstructed.

However, such a scheme does not possess the means to constrain the power given to a quorum of agents. This motivates the introduction of the concept of an *undetachable threshold signature* which both distributes signature authority across multiple agents and simultaneously constrains the signatures that may be constructed.

The rest of the paper is as follows. In Section 2 we outline the undetachable signature scheme of [2], and in Section 3 we briefly review threshold signatures and give a method of Shoup [1] to construct such a scheme. Finally, in Section 4 we define the concept of an undetachable threshold signature, and show how an example of such a scheme may be obtained by combining the schemes of [2] and [1].