The security architecture builds on ideas from Proof Carrying Code (PCC), and requires that mobile code is provided with a certificate, ie a condensed mathematical proof that the code is secure. In order to be applicable to global computers, the MOBIUS project will pioneer a PCC architecture that accommodates the distributed nature of global computing and allows enforcement of advanced policies, including both functional properties and advanced security properties such as non-interference or re-source control. To ensure scalability, the MOBIUS project will also extend and combine two prime enabling technologies of PCC, ie type systems and program logics, and use the strength of these two techniques in hybrid certificates, to be verified through combined type checking and proof checking. Finally, the MOBIUS project will develop certificate translation as a means to bring to the code consumers the benefits of program verification, which is almost universally performed at source code level.

To maximize its impact, the MOBIUS project is focusing on Java-enabled global computers, and uses program logics that support the Java Modelling Language (JML). It will allow an implementation of the MOBIUS security architecture to be built on top of existing tools developed within the consortium. This will be evaluated on case studies from a range of application domains, covered by the consortium industrial partners, and by the End User Panel.

The project is coordinated by the French research institute INRIA, and is part of the pro-active initiative Global Computing 2, launched by the Future and Emerging Technologies unit of the IST programme.

Link:
Project web site: http://mobius.inria.fr
Please contact:
Gilles Barthe, Project coordinator:
Tel: +33 4 92 38 79 38
E-mail: Gilles.Barthe@sophia.inria.fr

Virtual Communities (VC) are a means of linking people with common interests, professional occupations, or resource-sharing habits. Several virtual communities already exist – Orkut, Kazaa, or BitTorrent to name the most prominent – and attract millions of Internet users. As a member of a virtual community one can access community resources that should be protected from outsiders. The challenges are to identify someone as a community member, since communities grow and shrink dynamically, and to establish trust between users on the basis of limited knowledge between parties. Existing virtual communities do little to protect their members from malicious internal and external users and as such are not acceptable for commercial use.

VC provide mechanisms that help to implement efficient, secure access to confidential data or protected resources. A member of a VC can access community resources that are not accessible to outsiders. For example, Apple could build a community of users who wish to download high-quality movies from iTunes’ distributed secure servers. There should be no centralized access point, so that the system is more scalable and easier to maintain. Different community members may also have different privileges, including those of reselling music to other members of the iTunes community. The actual quality and the exact limitations of linking people with common interests, professional occupations, or resource-sharing habits. Several virtual communities already exist – Orkut, Kazaa, or BitTorrent to name the most prominent – and attract millions of Internet users. As a member of a virtual community one can access community resources that should be protected from outsiders. The challenges are to identify someone as a community member, since communities grow and shrink dynamically, and to establish trust between users on the basis of limited knowledge between parties. Existing virtual communities do little to protect their members from malicious internal and external users and as such are not acceptable for commercial use.

VC provide mechanisms that help to implement efficient, secure access to confidential data or protected resources. A member of a VC can access community resources that are not accessible to outsiders. For example, Apple could build a community of users who wish to download high-quality movies from iTunes’ distributed secure servers. There should be no centralized access point, so that the system is more scalable and easier to maintain. Different community members may also have different privileges, including those of reselling music to other members of the iTunes community. The actual quality and the exact limitations
on the amount of music that is accessible to community members depend on how much they wish to pay. In this scenario, the outsiders would be all the Internet users who do not pay for this access. As they are not members of the iTunes community, they cannot access any files there and cannot inject fake music into the network. Additionally, such controlled access to community resources helps not only to protect confidential information, but also to plan the use of bandwidth or load of service providers.

Trust management makes security transparent to the end users so that they no longer require substantial knowledge of the security mechanisms. Many existing approaches to trust management require a centralized architecture and thus do not fit well in the distributed nature of the Internet. Therefore, in our research, which started late 2004, we are combining distributed trust management and VCs into one solution that is suitable for both commercial and private use.

The first question we are trying to answer is how to formalize support for VCs in a trust management system. Secondly, because of the highly distributed nature of the Internet, we also need to investigate under which conditions the information necessary for secure user authentication is available. Finally, we aim to construct a suitable system design that provides security that is transparent to the end user.

Each community member must be able to prove membership of a community prior to being granted specific access permission. Such proof can be thought of as a distributed set of signed statements called credentials. Each credential consists of at least an issuer, a subject and a statement representing the actual meaning of the credential. The issuer and the subject of a credential both represent Internet users. A credential expresses the issuer’s trust in the subject, giving the subject the permissions stated in the credential. The number of VCs one can join is unlimited; therefore one can be the subject of many credentials at the same time.

As a typical scenario, imagine that Alice would like to share the videos she took during her last holiday (see Figure 1). She does not want the whole world to see her footage, but rather wishes to restrict the potential audience to her friends. Alice can create a virtual community – ‘friends’ – which contains Alice’s friends. To realize this, Alice issues credentials, one for each of her friends, saying that he or she is a member of the community ‘Alice’s friends’. In practice, Alice does not have to contact her friends personally, but the whole process can take place behind the scenes, using software agents acting on behalf of Alice and her friends. Alice’s friends may not be even aware of the whole process and simply view Alice’s film as if it was their own.

When Bob wants to see Alice’s films, his software agent contacts Alice’s software agent to check whether the credentials provided by Bob’s software agent are valid. Checking validity of credentials is not the only task a software agent must perform. If Alice’s software agent detects that the credentials received from Bob’s software agent are not sufficient to grant permission to see the pictures, Bob’s software agent might be asked to provide additional credentials. As all credentials are stored in a distributed fashion, the job of Bob’s agent is to track down the necessary credentials. The difficulty is in locating all the necessary credentials efficiently.

Imagine that some time later, Bob, encouraged by his good experience with Alice, also wants to share his videos. Bob then creates his own ‘friends’ community, and each member of the ‘Bob’s friends’ community is allowed to see Bob’s movies. Additionally, as Bob trusts Alice, he decides that all members of ‘Alice’s friends’ community can also access his movies. In other words, Bob says that any friend of Alice is also a friend of his. We say that Bob delegates authority over his ‘friends’ community membership to Alice.

Even more sophisticated scenarios might arise. Therefore, our overall objective is to design and implement a sophisticated trust management framework that simplifies the protection of confidential resources in virtual communities. We use a so-called role-based trust management approach to model both trust and virtual communities. In our approach we use logic and logic programming to represent complex trust relationships that can appear in a real life experience. We also propose a new role-based trust management language with a formal declarative meaning based on the well-founded semantics for logic programs.

Our research is conducted as part of the I-SHARE project, which is part of the Freeband consortium funded by the Dutch national BSIK program. I-Share involves five project partners: Delft University of Technology (TU-Delft), Eindhoven University of Technology (TU/e), University of Twente (UT), Vrije Universiteit Amsterdam (VU) and Philips Research, Eindhoven.

Links:
Freeband Communication web site: http://www.freeband.nl/index.cfm?language=en
I-SHARE Project web site: http://www.freeband.nl/project.cfm?id=520
DIES group home page: http://dies.cs.utwente.nl

Please contact:
Marcin Czenko,
University of Twente, The Netherlands
Tel: +31 53 489 3709
E-mail: Marcin.Czenko@utwente.nl

Jeroen Doumen,
University of Twente, The Netherlands
Tel: +31 53 489 2801
E-mail: Jeroen.Doumen@utwente.nl