Using mobile agents for secure biometric authentication

Marco Tranquillin
Department of Information Engineering
The University of Padova
Padova, Italy

Carlo Ferrari
Department of Information Engineering
The University of Padova
Padova, Italy
Email: carlo.ferrari@dei.unipd.it

Michele Moro
Department of Information Engineering
The University of Padova
Padova, Italy
Email: michele.moro@dei.unipd.it

***Abstract*—This paper deals with the definition of a strong authentication model, coupling usual password/PIN based methods with a biometric matching, over a Multi Agent distributed infrastructure. When the user authentication procedure involves personal devices, the Multi Agent System model helps in the distribution of data and algorithms thanks to a better partitioning of roles and responsibilities, enhancing robustness to eavesdropping and tampering by properly moving agents around the system itself. The system architecture is based on specialized agents tied to the different devices, which safely communicate using both symmetric encryption for messages and asymmetric encryption to check principals' roles. Moreover, agents can carry on biometric parameters matching algorithms, bringing computation on those nodes with enough computing power. A complete authentication protocol has been developed and two different demos have been devised and tested. They differ for the tasks assigned to the mobile devices in use. Experiments show that agent capabilities, together with their power of migration, help in maintaining a higher level of security when mobile devices are involved.**

I. Introduction

Identity recognition is still an open problem and its solution can help for user authentication, tracking, secure access to restricted areas and, more generally, anytime it is necessary to automatically acquire the identity of a human operator. It is worth to point out that the application of an identity recognition system can go beyond security issues, being the basic for a more sophisticated human machine interface in a pervasive context. A person can be identified through the analysis of her physical peculiar features: automated methods and algorithms for feature acquisition, analysis and recognition stand at the core of Biometrics, that it is focused on the correct representation and measurements of those personal physical invariants, like fingerprints, iris, hand geometry and so on, that are strictly linked to each person and cannot be easily tampered with. In the recognition process acquired raw data are preprocessed to extract relevant geometric arrangements of features that are then compared and matched to those that form the user template stored in a backend database.

A biometric recognition process involves different computational activities distributed among biometric sensors, host computer, personal mobile devices. Modern computational systems can involve heterogeneous machines and devices that greatly differ with respect to their computational power. Some of them can move freely in the working space thanks to wireless connections while remaining strongly tied to their (human) owner. The interaction with the fixed part of the system is managed by proper protocols in order to guarantee security and performances. The design of applications can benefit from novel paradigms and methodologies that explicitly deal with mobility and that support delegation mechanisms in order to move the heavier part of computation on those nodes that can cope with the required service levels.

The mobile agents paradigm is focused on the mobility concept of computation and code [1]. That means the ability to organize programs that can be sent without changes to a number of different computers and that can be executed with the same semantics on each of them. This paradigm is a remarkable option in the context of Multi-Agent Systems (MAS) that in [2] are defined as "a loosely coupled network of problem-solver entities that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity". One of the most interesting and innovative feature of this model is about its integration with modern mobile devices, like smart phone or handhelds, that allows a greater relocation of computing resources linking software with users in a more safe effective and reflective manner. For sure security is a major issue in distributed systems and device mobility requires new more robust solutions [3], [4], [5].

The main goal of this paper is to present how biometric matching methods can be combined with usual password/PIN based methods in those scenarios where mobile devices are involved. The paper is focused on the use of the mobile agent paradigm suitably exploited to integrate a mobile device in the system.

The authentication model that we present is a *strong authentication model* [6]. In fact it involves more than one factor, *something that the user knows* (personal data like name, surname, date of birth etc. and one PIN generally), *something that the user has* (a mobile device with its memory card) and *something tied to the user* (a biometric parameter).

A model using three contextual authentication criteria is more robust and is able to fully take advantage of the mobile agents approach. Unlike previous works that are based on

smart card [7], [8], [9], the innovative issue of this model is the introduction of a mobile device that can be used both like a data repository and an active element inside the authentication protocol, executing matching of biometric templates.

Mobile agents can transfer entire object code in order to manipulate data in a more robust manner (there is no need to copy and paste data from one location to another) and to react immediately either to errors or to exceptions (for example an agent that must install particular software that needs some libraries can autonomously download them from Internet without any user intervention). Thus a MAS supports effective forms of adaptable computations in accordance with the requested level of security and the availability of computational power, and it shows a relevant degree of scalability and maintainability. The framework that we chose for implementing the proposed authentication system is the open-source project *Jade*. It's a worldwide project associated with a great community that has realized a lot of adds-on, like Jade-S and Jade-Leap. The first one is a plug-in that increases the native security level of the platform through the verification of credentials that belong to agents. The second one allows to execute a light version of the Jade framework into a device that has limited hardware resources like mobile phones or handhelds.

The authentication system that we propose is based on biometric parameters (in the first release we use only fingerprint analysis). It addresses the following questions:

- how can we surely transport user private data?
- how can we extract the template from a fingerprint from a trusted source?
- how can we be sure of subjects that are exchanging data?

In the next paragraph the architecture of the system is described while, in the third paragraph, the details of the authentication protocol are given. The fourth paragraph is devoted to a brief final discussion and conclusions.

II. System Architecture

Authentication is carried on by different agents that perform specific tasks and with at least the robustness reachable with elaborated protocols like Kerberos [10]. The involved physical entities of the system are a mobile device, typically a smartphone, a client machine mainly devoted to biometric data acquisition, and a server machine to access the central user database. Within these machines both static and mobile agents run on an agent platform. The overall architecture is built on the following basic elements:

- *server agent*: it must execute the operations at the server side, e.g. generating mobile agents, checking data, querying a database and so on;
- *user phone agent*: running on the mobile device, it provides the GUI for the owner to be authenticated. The necessary communications with the server are performed via messages piggybacked on another (mobile) agent;
- *mobile agent*: it moves between a client and the server carrying all the necessary information for the authentication process; when at client it is delegated for all the communications with the phone agent.

Instances of the previous components can be combined to form the final system, as shown in figure 1.

It is worth to point out that the role of the mobile agent is to load programs "on the fly". Then it is not necessary to preinstall all the software at the client side, reducing the risk of hacking and tampering of those critical components that are devoted to the authentication. Moreover executing software within a remote agency benefits of those encapsulation capabilities of agents providing a stronger degree of security. The abovementioned issues are even more significant when the mobile device can host an agency.

Every client machine is responsible for controlling one biometric sensor from which it gets raw data when requested. The match of the live template, extracted by the client, can be centralized in the server. The use of a mobile device, coupled with one client machine, enables a stronger level of security because also PINs must be provided by the user and, with suitable computational power, the match can be carried directly on the mobile device. In order to reach a useful flexibility and the adaptation of the client to the characteristics of the user phone, all the main functions of the client are carried by a mobile agent coming from the server. For example, a specific mobile agent could be created by the server in accordance with the capability of the mobile device currently shown.

Every communication from one agent to another is protected by symmetric encryption to obscure contents of messages and by asymmetric encryption to apply digital signature to every message in order to let the agent automatically verifying the identity of message sender. Since we use an agent platform, it is possible to identify a specific agent (each agent that is running on a platform has its unique name) and to find its location on the network in order to be sure to talk to a non-tampered agent.

III. The authentication protocol

The biometric based authentication system, as mentioned earlier, leverages the power of the platform agents to execute all the operations that are mandatory to authenticate a user. The protocol is organized into a sequence of four phases:

1) Initialization.
2) Authentication of the mobile device.
3) Authentication of the shared secret.
4) Authentication of the biometric parameter.

These steps refer to the case we called 'Template on phone' that contemplates the presence of an encrypted fingerprint in the mobile device memory and the match of the biometric parameter on the server.

- Phase 1: the initiative is taken by the user who runs an application on her mobile device. After she has provided a PIN, the mobile device application asks the Jade platform server to start an agent called "User Phone Agent" (UPA). This also activates the client to start the authentication: it requests the server for establishing an SSL session.

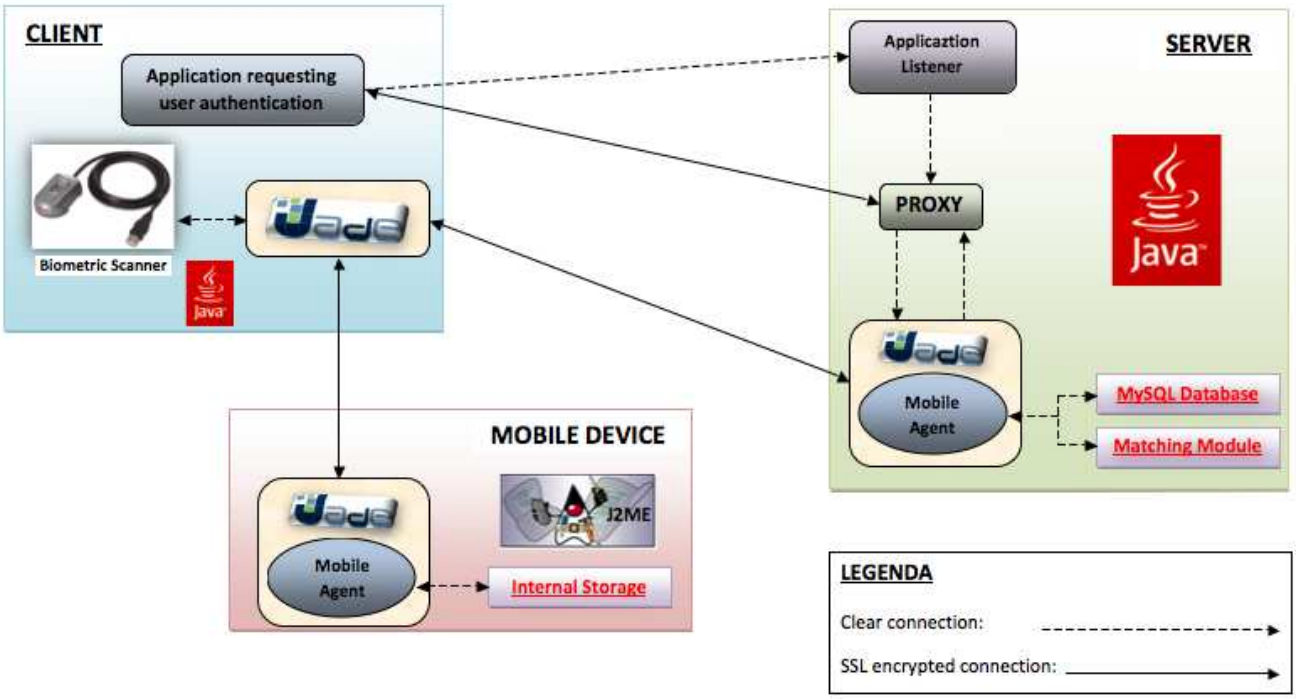

Fig. 1. Internal Model

Finally an other agent resident on the server (simply 'server agent', SA) is activated.

- Phase 2: this phase is dedicated to authenticate the mobile device (*something that the user has*). In order to authenticate the UPA, the SA creates a mobile agent (MA) that brings a challenge (i.e. a prompt to receive a private information as its response [11]), encrypted (with the server AES key) and signed by the SA, to the client by migrating to this latter. The UPA verifies the server signature, extracts the challenge and it sends a response challenge (updated, encrypted and signed), together with the user asymmetric public key, back to the server through the MA. The authentication of the UPA completes when the MA goes back to the server that verifies the encrypted reply using the enveloped public key.
- Phase 3: now it is the turn to get the user personal data (*something that the user knows*). The MA carries a new challenge to the UPA requiring a hashed version of the user personal data and PIN. When it comes back to the server, the SA can search the user data in the local database, through their hashed value. Any time a positive match occurs, the user AES key is extracted from the database record and it is used as the basis of a personal secure communication channel for the next steps. At this point the UPA receives an acknowledgement affirming that the server has correctly recognized the user and consequently the authentication can continue.
- Phase 4: this phase includes the biometric match (*something tied to the user*). The UPA provides the user fingerprint template that is stored in the mobile device memory whereas the client is in charge to provide the live template. The MA goes back-and-forth two times, the first to bring the encrypted stored template, the second one to make the client to get the user live fingerprint. The live template is encrypted with the asymmetric public key of the server so that the client is not requested to maintain any secret key. The two templates are now compared using the established matching algorithm. The last travel of the MA is used to notify the match response to the UPA.

When a 'Match on phone' is possible, i.e. the mobile device can implement the matching algorithm, the live template must be sent to the UPA and the stored template is not moved from its original position in the phone device. Obviously in this case the UPA is responsible for performing such a match and for giving back to the server the final response.

IV. Discussion and Conclusion

In our experimentation we developed the two different scenarios through the following demonstrative applications:

- *Demo 1 "MIDP-TOP"* (Template On Phone): the biometric match is on server, whereas the mobile device is used as the user personal data repository;
- *Demo 2 "MIDP-MOP"* (Match On Phone): the biometric match is performed on the mobile device. In this case the reference template can be stored exclusively on the mobile device.

To complete our prototype we developed a tool named *Enrollment Tool* that allows the system administrator to manage user personal data and keys and, more important, to extract a

template from a fingerprint (that can be loaded from a file or live captured) and to support the initialization of the mobile device repository.

A distributed system with mobile devices requires proper policies to combine different security mechanisms in order to meet a high level of dependability. Biometric-based identity recognition benefits from the association with other (non biometric) personal data, allowing a faster search on a common database, thanks to the use of well established and efficient string search algorithms.

A MAS explicitly supports the partition of responsibilities and roles: in fact it is a model that is distributed 'per se' and that assigns different responsibilities to the various actors. The concurrent use of both biometric and non-biometric parameters asks for such kind of partitioning. When multi-biometrics is concerned, this is even more evident because more client machines equipped with specific devices and tools could be involved. Agents are entities that can be precisely identified within a platform together with their specific responsibilities, acting in favour of their own or on behalf of other entities, particularly in the case of mobile agents.

As previously described, we propose a sequence of three steps recognizing first a mobile device as member of a set of authorized devices, second a user through her personal credentials and finally we perform the biometric match. In this scenario the client should be simply responsible for the reading of the raw biometric parameter. Notwithstanding, the use of a mobile agent permits any client to execute crucial actions on delegation of the server to cope with the limitation of the mobile device. So that also the user mobility is implicitly guaranteed not being restricted to a specific client (e.g. a user could authenticate herself to enter a restricted area through one of several different entrances).

An appropriate combination of secret keys related both to the machines and to the single users provides a good balance between performances and the level of security. It is worth to point out that the Jade security extension (Jade-S) unfortunately presents some hard limitations that make it unusable when mobile devices are involved.

Basic security is provided by the SSL tunnel. Freshness and liveness properties are guaranteed by the challenge-and-response approach, like in the Needham-Schroeder protocol [11]. Every one of the three recognition steps in sequence is correct because it realizes standard security procedures, and it provides its specific level of security, related to the entity under authentication (device, user ID data and biometric data). Whenever a step fails the whole authentication process fails. If a step is fraudulently passed, no effect of further weakness is propagated at subsequent levels that use new and different critical data. The last step, that uses biometric data, is the hardest to be misleaded.

Our experimentation, necessarily carried out on a small group of a dozen people, has proved the correctness and effectiveness of the model. In perspective the MAS architecture shows a sufficient degree of scalability to adapt our model to significantly more complex situations like those requiring several client locations and a great number of potential users. The future availability of more powerful smartphones will also bring the conditions for introducing the agent mobility at the phone level.

ACKNOWLEDGEMENT

This work has been partially supported by the University of Padova Research Project CPDA073251/07, "Algorithms and methods for secure authentication using biometrics data".

REFERENCES

[1] S. J. Russel, P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall International, 2003.

[2] J. F. Dray, M. E. Smid and R. Warnar, *A Token Based Access Control System for Computer Networks*, Proceedings of the 12th National Computer Security Conference, NIST/NCSC, Baltimore, MD, (USA), October 1989.

[3] T. Y. C. Woo and S.S. Lam, *Authentication for Distributed Systems*, IEEE Computer, vol. 25, no. 1, pp. 39–52, January 1992.

[4] *Guideline on User Authentication Techniques for Computer Network Access Control*, National Institute of Standards and Technology, Federal Information Processing Standards Publication 83, National Technical Information Service, Springfi eld, VA, September 1980.

[5] H. Aouadi and PR M. Ben Hamed *Security Enhancements for Mobile Agents Platforms*, IJCSNS, vol. 6, no. 7, pp. 216–221, 2005

[6] R. E. Smith *Authentication: From Passwords to Public Keys*, Addison-Wesley. 2001.

[7] S. Bistarelli, S. Frassi, A. Vaccarelli, *MOC via TOC Using a Mobile Agent Framework*, Proceedings of the 5th Int. Conf. on Audio and Video-Based Biometric Person Authentication, Hilton Rye Town, NY, USA, 2005, pp. 464–473.

[8] S. Bistarelli, F. Santini and A. Vaccarelli *An Asymmetric Fingerprint Matching Algorithm for JavaCard*, Proceedings of the 5th Int. Conf. on Audio and Video-Based Biometric Person Authentication, Hilton Rye Town, NY, USA, 2005, pp. 279–288.

[9] Z. Pozgaj, I. Duretek, *Smart Card in Biometric Authentication*, Proceedings of the 18th Int. Conf. on Information and Intelligent Systems, Varazdin, Croatia, 2007, pp. 319-325.

[10] J. G. Steiner, C. Neuman, J. I. Schiller *Kerberos: An Authentication Service for Open Network Systems* in Usenix Winter Conference Proceedings, Dallas, Texas (USA), 1988, pp. 191-202.

[11] A. S. Tanenbaum, M. van Steen, *Distributed Systems: Principles and Paradigms*, Prentice Hall, 2002.