Institut Eurécom[1]
Mobile Communications Department
2229, route des Crêtes
B.P. 193
06904 Sophia Antipolis
FRANCE

Research Report RR-07-197

Service Discovery: Reviewing Threats and Security Architectures

7 May 2007

Slim Trabelsi, Yves Roudier and Jean-Christophe Pazzaglia[2]

Tel: (+33) 4 93 00 81 00
Fax: (+33) 4 93 00 82 00
Email: {Slim.Trabelsi, Yves.Roudier}@eurecom.fr
Jean-Christophe.Pazzaglia@sap.com

[1] Eurecom's research is partially supported by its industrial partners: BMW, Bouygues Télécom, Cisco Systems, France Télécom , Hitachi Europe, SFR, Sharp, STMicroelectronics, Swisscom, Thales

[2] SAP Labs France 805 Avenue du Dr Donat - BP 1216, 06254 Mougins Cedex – France

Abstract

Service Oriented Architectures (SOA) introduce a loosely coupled interaction model which requires discovering services that enable an efficient interconnection between different application systems or software components. Although service discovery has been thoroughly studied in the past, its security has been vastly ignored. After introducing some security issues of service discovery as illustrated in a healthcare motivating scenario, this paper classifies the various threats to service discovery. For each threat we propose a retort. We describe two solutions (centralized and decentralized) to protect users privacy and ensure access control to sensitive discovery data: a centralized one, relying on a classical policy based approach, and a decentralized one, relying on attribute-based encryption, and especially adapted to the use of SOA for developing ubiquitous computing applications, in which case trust is considerably more distributed.

1 Introduction

Service Oriented Architectures (SOA) introduce a loosely coupled interaction model which serves as the basis to define protocols and procedures that enable an efficient interconnection between different application systems or software components. SOA basic components mainly consist of services, which provide elaborate functions (database access, data processing, business logic…), and of clients that are requesting such services. These two types of players rely on a standardized interface to communicate but do not necessarily share the same implementation platforms, both in terms of programming language or operating system. The SOA paradigm is currently largely promoted by the spreading of Web Service technology. Web Services overstep the limitations of traditional distributed component solutions like Jini [1] or CORBA [2] in that they increase the dynamicity and flexibility of distributed software thanks to the use of XML-based interfaces, WSDL [3] being a perfect example of such interfaces.

Orchestration is becoming an essential feature for developing software for increasingly pervasive systems, in particular with the fast development of ubiquitous computing. The orchestration technique obviously comes at a cost: being able to locate previously unknown services becomes mandatory. Service discovery is an essential part of orchestration that allows a dynamic detection of the services available in the network. Many service discovery protocols have been proposed so far: Universal Plug and Play (UPnP) [4] is a basic discovery protocol used to interconnect small devices in a home network. The Service Location Protocol (SLP) [5] is also used for small and local networks. The Jini lookup service offers Jini clients a flexible and powerful way to find Jini services. It enables service providers to advertise their services and helps clients to locate these services, by using a lookup table (service database).

With the emergence of the Web Service technology, the discovery process should address the heterogeneity of services and platforms from a technical perspective, the complex semantics of service descriptions (e.g. resorting to terminology- or ontology-based descriptions), specific security and trust requirements, altogether with scalability. Web Service discovery solutions like UDDI [6], WS-Discovery [7], or OWL-S based approaches [8] were developed to answer some of these requirements, yet they still do not address most security and trust issues. In the WS-Discovery protocol for instance, security is limited to the use of signatures for verifying the integrity of messages exchanged and for preventing message replay. It is not sufficient to protect sensitive information about services from becoming available to rogue users; private information of a user might also get revealed to a service without any assessment of that service's potential maliciousness. This paper discusses how WS-Discovery may be extended to incorporate appropriate confidentiality and privacy protections restricting the potential matching between a client lookup request and a service profile.

This paper is organized as follow. In the section 2 we provide a healthcare scenario, in which we describe the interaction between mobile users and the

challenges related to these ubiquitous applications. In the section 3 we specify a threat model related to the service discovery. In the section 4 we detail two solutions that can be used to secure the service discovery. Finally we compare our approach with related work.

2 Motivating Scenarios: A Healthcare Case

2.1 Environment and assumptions

Information systems are becoming global and this integration trend is interconnecting systems across standard and well understood administrative boundaries. This results in large and uncontrolled data exchanges which may threaten user privacy and citizen rights in the sense of the European Directive 1995/46/EC [9], on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Healthcare scenarios provide a good example of such protection requirements for patient data. Hospitals are increasingly making use of short range communication capable equipment, mostly based on IEEE 802.11, and will likely take advantage of such an infrastructure to interact with patients in a more transparent manner in the near future.

The following scenario describes how a healthcare system might look like in the near future with the introduction of mobile and ubiquitous computing for monitoring patients, and some threats raised by such technology.

2.2 Scenario story

Since a couple of years, Bob has been diagnosed as diabetic. In order to control his blood sugar level, he carries a monitoring device that regularly measures his blood sugar and some other data such as his heart rate, blood pressure, etc. All these devices are integrated to an e-health terminal. Bob's monitoring device and his motion sensor generate alerts that are correlated and communicated to Bob's physician and to nearby hospitals.

2.2.1 A problem on a business trip

Bob is on a business trip in Italy. After a busy day, he suddenly feels dizzy. Bob takes his e-health terminal and contacts his physician, several hundreds kilometers away. The latter has just come home when he receives Bob's message on his PDA. The physician checks Bob's medical history and compares it with the recent data. The physician calls Bob on his mobile phone and prescribes him a drug. The prescription is done electronically and stored onto Bob's e-health terminal. In addition, he strongly advises Bob to visit a local physician as soon as possible. Of course, Bob does not want anyone to be able to find he is carrying a e-health terminal, even less to access his medical file (apart from his regular physician)!

2.2.2 Bob goes to a pharmacy

Bob checks on-line with a health directory system for the closest pharmacy that has the prescribed drug on stock and sends his prescription. At the pharmacy, Bob presents his e-health terminal as an electronic identity card to the pharmacist, who simply hands the drug to Bob. Bob and the pharmacist mutually sign a receipt using Bob's e-health terminal and the pharmacy computer. Bob should not be able to generate fake prescriptions or to fake a doctor's medication. The same holds true for the pharmacist. Additionally, the information about pharmacy locations should be trusted, so that for instance, Bob does not send his prescription to a pharmacy on the other side of town.

2.2.3 Emergency!

Bob enjoys a wonderful weekend in the countryside when he suddenly feels dizzy again and shortly after faints. His personal health monitoring system raises an alert which is received by the emergency response centre. The emergency response centre locates Bob's mobile phone then sends an emergency response team to Bob's location. After they find Bob, the team authenticates with his e-health monitoring device in order to gain access to the private information stored in the device about Bob's health status and his recent medications. Some situations make it necessary to discovery personal data services then access them, without their owner's authorization. This can only be enabled by contextual access control features. The emergency and rescue team should also discover Bob's terminal, and not that of Eve who is pretending to be Bob just out of maliciousness.

2.2.4 At the hospital

Upon arriving at the hospital, Bob's medical data are retrieved and transferred into the hospital information system. A physician takes care of Bob in collaboration with Bob's physician. Only physicians should be able to discover and access Bob's terminal, locally or remotely, as well as the information it contains.

2.2.5 Epilogue

After two weeks, Bob has recovered enough to be sent back home. The hospital closes his case and sends all data to Bob's physician, as well as the necessary electronic forms to Bob's health insurance. The data gathered on Bob's e-health terminal are anonymized and sent to the Ministry of Health for statistics. Again, an authenticated discovery should be performed in order to send personal medical data to authorized services only, according to the medical workflows in place.

2.3 Healthcare administrative workflow

The health system of many countries, especially in Europe, is going digital. Smart cards are for instance being used for handling the reimbursement of a physician consultation or pharmacy bills. However, a large part of the

prescriptions and medical records of patients is still handled through paper exchanges. Such exchanges are prone to loss, intentional or unintentional disclosure, or even forgery. The procedures used for the reimbursement of physicians' medical acts have also become quite complex. They essentially involve official statements and agreements with the health insurance and require that official documents travel from the physician or patient to the health insurance premises, then back, as well as a lot of paper handling. One of the objectives of healthcare organizations will clearly be to shift most of their workload from administrative to medical control and advice tasks. Integrating all the data into a single healthcare information system is therefore an important task.

2.4 Privacy protection and emergencies

Every country's health system will probably deploy its own access control system to protect patient data for confidentiality and privacy reasons. In particular, accessing to services providing data such as the medical records of a patient, or even retrieving the specialties of the physicians consulted is likely to be feasible only with appropriate privileges. This raises a first issue: if, as envisioned above, the whole health information system has gone digital, how to interoperate between different health authorities with diverse procedures and forms? In addition, some emergency situations might require access to vital patient data notwithstanding any consideration of privacy.

Today and despite the availability of cryptographic solutions, health authorities are still not properly enforcing the protection of personal data (e.g., the lack of encryption of data in the first version of the French health smartcard "Carte Vitale"). Moreover, no system can keep accountable and legally liable traces of the browsing, and even more problematic, of the modification of a patient record.

3 Service Discovery and Security

As shown in the previous section, since ubiquitous services surround users, one of the main challenges is to provide discovery information to this pervasive environment without exposing the user to new threats. This section first describes discovery mechanisms followed by a detailed threat model for service discovery.

3.1 Service Discovery Definition

Communication devices in fixed networks like local LANs traditionally are assigned a static network configuration, or at worst use DHCP to dynamically configure their IP address. The DNS protocol is quite sufficient to find a host in such networks using its IP address or its domain name. With the emergence of new dynamic networks and services where devices are pervasive, the discovery techniques are being adapted in order to find mobile services rather than devices. In particular, this adaptation addresses how to combine services as a logical layer in such systems while taking into account environmental constraints.

Centralized discovery approaches rely on a registry which plays the role of yellow pages, and which clients can refer to. A service advertises its capabilities

to the registry, which will store them for a certain amount of time. A client solicits the registry to find a service by sending a request containing service preferences, which the registry tries to match with the most suitable provider found from the stored advertisements. In that approach, registries have to be considered by the services and the clients as a trusted third party.

Limiting service discovery to registry supported architecture, like many standard SOA based services adopted it in their implementations, reduces the applicability of service discovery in ubiquitous environments. An alternative approach to centralized service discovery mechanisms exists that relies on peer to peer advertisements between services and clients. The decentralized discovery approaches does not rely on some extra third parties, but instead on direct exchanges between clients and services, with no mediation. A server advertises its service capabilities to the users by multicasting the service profiles. Clients have the possibility to cache the service profiles information provided by the services, and reuse it if needed. The client also has the ability to ask for a new service (absent from his cache) by multicasting its request to all available services. This mechanism is used for instance by UPnP.

3.2 Revisiting Service Discovery Threats

This section discusses the threat model of service discovery services and in particular which parts of such systems would be worthy targets to adversaries.

3.2.1 Service Discovery Players

The main players of the discovery phase are the service requester (client) and the service provider (server), even in the case of a registry based service discovery. The specificity of service discovery is that, by definition, these players are initially unaware of their respective existence and of their security policies. In addition, they are often likely members of different administrative domains.

3.2.2 Important data and resources

- Service profile: this data structure contains all the details describing a service (like WSDL in Web Services technology). It could provide the server's URL address, the methods and the parameters necessary to access to the service. We can also find some information about the owner of the server and its location.
- Server's identity: it is contained in a signed certificate and used by the service in order to be authenticated by the other parties of the system.
- Client's Identity: it is also contained in a signed certificate and used by the client in order to be authenticated.
- Client's lookup request content: these data exhibit the client's intentions and interests. Such data in some cases could be considered as private.
- Discovery protocol messages: it can be modified or corrupted in order to disturb the system correct behavior (Denial of Service, Man in the middle attack).

- Registry: the central part of a centralized discovery system, it has to be available (robustness) and it has to protect all the data stored locally.

3.2.3 Threats and Attacks

This section provides a non exhaustive list of threats and the possible attacks that can be built against the data and resources of service discovery players. For each threat, a possible countermeasure is proposed that can be applied in order to prevent the disruption of the service discovery service. A more comprehensive security architecture is presented in the next section. The following description lists threats to the centralized and decentralized service discovery architectures together.

Protocol Messages and Entities

- The registry is not available (service-side): the attacker performs a Denial of Service attack by flooding registration messages. He intends to force the registry to consume its resources such that it can no longer provide its intended service. One of the possible countermeasures is to modify the protocol by adding anti-clogging messages if message parsing is too costly, then blacklist originators of bogus messages.
- Client request disclosure (client-side): client intentions, or activity, or identity may be revealed, directly or indirectly by his service lookup queries. An appropriate countermeasure consists in setting up secure channels (encryption). In our scenario, an insurance company could intercept patient requests in order to illegally evaluate his health condition.
- Interception of request (client-side): the discovery request reveals private information about service discovery clients. A possible attack consists in faking the identity of a registry that is known and trusted and forwarding to that registry. Registry certificate distribution might be an adapted countermeasure to prevent this type of attack. The process of distributing certificates of trusted registries should be protected during the configuration phase of the mobile device. A malicious user could play a masquerade attack in order to obtain illegal drugs from the pharmacist.
- Message modification or drop (client side): if the attacker compromised router from the network, he can intercept and modify or drop the client's lookup message to the registry. The client should protect the message it sends with respect to its integrity and to the authentication of its origin, for instance with a signature or a message authentication code. A redundancy mechanism can be configured to guarantee the delivery of the messages in case of dropping.
- Replay of lookup message DoS (client-side): the attack consists in replaying a lookup message coming from a legitimate client. A sequence number could be added to the message in order to drop the previously processed messages.
- Replay of registration message (registry-side): the attacker replays the registration message of a properly authenticated service in order to update the service profile with wrong information. A signed sequence number

must be added to the registration message in order to take into account the processed messages and drop the relayed ones. A malicious pharmacist could use this attack to redirect all the patients to his shop.

Service Registration (centralized architecture only)

- Registration to a malicious registry (server-side): an attacker might fake being a registry whose identity (and implicitly matching behavior) is known and trusted. Subsequent attacks include preventing clients to match the registered service for instance. Registry authentication is one possible countermeasure. This can be achieved by ensuring a properly protected distribution of the certificates of trusted registries during the configuration of mobile device, which also requires an initial authentication phase during discovery, or by ensuring that registry keys are distributed to mobile devices and that communication with the registry is encrypted with that key.
- A service can be deregistered by an unauthorized party (service-side): occurs when an attacker tries to dereference an active service from the registry which it registered to previously. The use of a nonce (e.g., sequence number) with a signature (MAC) by the registered service for certifying the origin of a de-registration message constitutes a possible countermeasure to such attacks.
- Wrong registration (registry-side): An attacker can send a fake registration message to the registry containing wrong information with fake attributes. To prevent this attack, the registry has to include a verification of the proper certification of attributes of registering services by appropriate authorities together with a proof of identity of the registering party (e.g., signature of registration request).

Matching Process

- Client lookup disclosure (client-side): client intentions or activity might be disclosed if the matching process is open to all services registered. A service may have been established with the objective of gathering statistics about users trying to access a certain profile of services. More dangerously, an attacker might try to get access to confidential information sent by the client at the access phase subsequent to service discovery. The countermeasure to this attack consists in restricting the services whose description matches the client lookup with additional constraints on some of their certified attributes. This specification can take the form of a policy submitted by the client together with his lookup request, and which may refer to the same or to different attributes of the services than those specified in the lookup request.
- Service discovered by unauthorized party (service-side): a typical example of this threat is the possibility for an attacker to determine the identity or content served by a service which wants to be seen or accessible only by a restricted set of other services (service trapping). The countermeasure consists in the delegation a trusted (and authenticated) registry the enforcement of a restrictive policy provided by the service that will allow

the service discovery by authorized clients only. We also recommend the usage of restrictive cryptographic mechanisms. A non-subscriber of the e-Health service that discovers the e-health and uses it illegally.

4 Architectures for a Secure Service Discovery

This section studies two different approaches that can be adopted in order to respond to the security requirements analyzed previously on the threat model. These solutions correspond to the two possible configurations of the system (centralized and decentralized).

4.1 Centralized Architecture

The threat model exposed in the previous section makes it clear that clients should be able to find a service matching their preferences, both in terms of the characteristics of the service and in terms of security and privacy requirements imposed respectively by the service and by the client. On the client side, the user should be sure that only services matching his preferences would be returned: from his point of view, trusting a service should therefore go beyond the simple authentication of the service provider and also encompass a complete certification process of the capabilities of the service. On the server side, the problem is quite similar since the server does not know the users that can potentially gain access to its service. They should therefore be accessible only to client they trust to access them according to a precise behavior guaranteed by some authority.

Assigning the responsibility to enforce such discovery policies to a trusted entity of the system is therefore critical to service discovery. To avoid raising the complexity of service discovery, we do not propose to add a new entity to the system together with a dedicated protocol, but rather to assign this task to the registry. The choice of the registry as being the trusted third party in charge of the policy enforcement is an absolute requirement in centralized approaches, since matching already implicitly is a trusted operation, and matching and policy enforcement are closely tied together.

Discovery policies [15] may be quite simple: the client or the service provides rules that describe who can access their respective profile based on some attributes. In this paper the discovery policy objective is twofold:

- Access Control: discovery constitutes a preliminary form of access control to services by restricting the clients which will be able to subsequently contact a service. The sensitive resource here is the service's profile that must be hidden to the non authorized users.
- Privacy Protection: the client can protect the private information he reveals for each lookup he performs (identity, intentions, favorite services …) from an uncontrolled disclosure.

Usual discovery messages (publish and lookup) should be accompanied by some credential (certificate, key, or token) in order to be authenticated by the registry, by a discovery policy that will be enforced by the registry in order to

protect the entities according to their desires, the whole being secured using a signature based on the credential transmitted for instance.This solution was implemented and integrated into a security platform developed for the European project MOSQUITO [10] as a building block for creating pervasive workflows where new participants are discovered on the fly, in the users' environments.

4.2 Decentralized Architecture

As shown previously, the policy based approach provides an efficient solution for fulfilling the security requirements (privacy, data protection, access control …) of each entity, yet it is bound to be deployed only where trusted registries are available, like for instance in smart buildings. In contrast with the centralized based solution, where users have the possibility to rely on a trusted third party used to protect the sensitive data exchanged during the discovery process (by granting user's policy enforcement), the decentralized solution has to make use of other mechanisms to obtain the same protection elements. It is possible to use a particular encryption scheme, able to express and enforce a policy at the same time, like attribute based encryption [11],[12] or policy based encryption [13]. Using these cryptographic mechanisms, the user has the possibility to encrypt messages according to a policy (ex: Role, ID, Domain …): only users that are holding mandatory credentials will then be able to decrypt messages.

[14] described how attribute based encryption could be used to protect sensitive information contained within the Discovery messages. To reach this objective, the ABE mechanism was applied to the principal messages exchanged during the discovery phase. The attributes used to encrypt the data are contained in a standard format called Service Profile (a set of attributes describing the service). Assuming now that these attributes identify the service, they can then be used to protect the client's service request messages by encrypting the totality or some parts of the message. For example, it is possible to send Encrypt[Message]{Attributes} in order to hide the attributes of services specifically requested by the user (like the requested service type and the identity of the requester). This guarantees that only the services that hold the private keys corresponding to these attributes are able to decrypt and process the Request message. Of course these private keys should be provided by a trusted Public Key Generator (PKG) to certified services only. The PKG should therefore verify the credentials exhibited by the service. This can be done using existing PKI infrastructures and a specific X509v3 profile. The profile should be tuned to capture the attributes describing the service. Symmetrically the service's response (Response Message) must also be protected, especially since the content of the message provides a set of attributes offering a precise description of the service (location, address, URI). All these attributes could be encrypted using a unique identifier of the user that requested the service and also other attributes related to the identity of the user (Role, Status …). In order to avoid carrying extra message exchange, the user can provide its identity in the Request Message. In this case, the encryption method used is Encrypt[ResponseMessage]{Identity,Role} − and

only the owner of this identity and this role (that holds the appropriate private key corresponding to this identifier) is able to decrypt the Response Message. As described previously, this private key can be provided by a PKG relying on existing PKI infrastructure.

5 Related Work

5.1 Threat models for service discovery

An earlier work [16] described some threats attached to service-oriented architectures that use a registry supported discovery. To our knowledge this paper was the first that detailed specific discovery security requirements (Authentication, Confidentiality, Access Control, Trust, Privacy, Non repudiation and accountability) and proposed a modification to the usual message exchange protocol in order to secure the discovery process. This proposal was heavily relying on the use on a trusted infrastructure for the discovery, including a registry that had the task to establish a trust relationship between the different actors of the system. This security model provides mechanisms that should be used during service discovery in order to protect not only the server, but also the service requestor regarding security and privacy.

[17] is another work exposing a detailed threat model of service discovery in ad hoc environments and taking into account the specific constraints of such environments (like power failures, path failures, routing failures, etc.). Its authors detail security requirements close to those presented in [16], yet do not describe a security architecture to overcome these security lacks.

5.2. Architectures for secure service discovery

One of the first papers that initiated the study of security aspects of service discovery is the work of Zhu et al. [18], which outlines a brief threat model, then presents a proxy based solution used to set up trust relationship by exchanging security keys between mobile users.

Other existing studies in the literatures also proposed solutions to secure service discovery like [19] in which the authors propose to add an entity providing a secure Service Discovery Service (SDS), which plays the role of a secure information repository or registry. This SDS helps clients and servers to set up a trust relationship and secure channels between each another: it provides authentication, encryption, signature verification, and message protection using a PKI. This kind of infrastructure is heavy to manage and only based on certificate verification; in this case every user with a valid certificate is able to discover every existing service without any restriction. Contrary to our solution, clients and services do not have any possibility to define their own security preferences regarding discovery. [20] addresses privacy protection aspects of the discovery process. The authors propose the use of Bloom filters to protect the client and server personal information (identity, certificates, attributes…). In this configuration each entity shares a bloom filter describing its identity (for clients) and its profile (service). Membership tests are performed between the directory

and the client using generated Bloom filters in order to match the client's request and verify the access right for the requested service. A huge constraint is related to the Bloom filters vectors that must be shared initially by the participants. We notice also that the scope of the restrictions is very poor compared to our policy solution that provides an efficient semantic expressiveness used to define the security preferences of each entity.

6 Conclusion

In this paper we propose a detailed threat model for the service discovery. We started our analysis by describing a healthcare scenario in which ubiquitous applications are interacting without knowing each others, and that motivates the need for securing service discovery as a source of additional vulnerabilities in ubiquitous information systems. The possible threats and attacks to service discovery in service-oriented architectures are then introduced and classified. Mechanisms destined to overcome these threats are then described: two approaches to making the service discovery process more secure are introduced. The first solution, dedicated to a centralized architecture, relies on the use of a trusted third party that plays the role of a registry (repository) by matching clients' requests with services' profiles and enforces the discovery policy required by each user. The second one, dedicated to a more decentralized architecture like the one found in ubiquitous computing, with many services surrounding the user, relies on attribute based encryption. This cryptographic mechanism is used to protect clients' requests and provides flexible and decentralized access control functionality for limiting the discovery of private attributes to trusted users.

Our undergoing work is focusing on the integration of the secured registry-based service discovery solution within business applications, notably in the eHealth domain. In addition, we are experimenting how the peer to peer discovery solution can be used in conjunction with the pervasive workflow model [21] and its security extensions recently developed within our team. This work aims at providing end-to-end security during the execution of workflow instances in a pervasive and dynamic environment.

7 Bibliography

[1] SUN Microsystems, Jini Specifications, http://java.sun.com/products/jini/

[2] CORBA, http://www.corba.org/

[3] WSDL specifications http://www.w3.org/TR/wsdl

[4] Universal Plug and Play http://www.upnp.org/download/UPnPDA10_20000613.htm

[5] Service Location Protocol, Version 2 RFC 2608

[6] OASIS, "UDDI", http://www.uddi.org

[7] WS-Discovery Specifications http://msdn.microsoft.com/ws/2005/04/ws-discovery/

[8] D. Martin et al, “Bringing Semantics to Web Services: The OWL-S Approach”, Proceedings of the 1st SWSWPC, USA 2004.

[9] European Parliament and Council of Europe of Europe Directive 1995/46/EC, on the protection of individuals with regard to the processing of personal data and on the free movement of such data, October 24, 1995

[10] Mobile Workers Secure Business Applications in Ubiquitous Environments (MOSQUITO) Project IST 004636 https://www.mosquito-online.org/

[11] Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Brent Waters “Secure attribute-based systems”. In CCS '06: Proceedings of the 13th ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA, 2006

[12] A. Sahai and B. Waters, “Fuzzy Identity-Based Encryption”, Advances in Cryptology-Eurocrypt'05.LNCS 3494, pp. 457-473, Springer, 2005.

[13] W. Bagga, R. Molva, “Policy-based cryptography and applications”, FC' 2005, 9th International Conference on Financial Cryptography and Data Security, 28 February-03 March 2005, Roseau, The Commonwealth of Dominica - Also published in LNCS Volume 3570

[14] S. Trabelsi, J.C Pazzaglia, Y. Roudier "Secure Web service discovery: overcoming challenges of ubiquitous computing" ECOWS 2006, 4th IEEE European Conference on Web Services, Zurich - Switzerland, December, 2006

[15] S. Trabelsi, L. Gomez, Y. Roudier “Context-aware security policy for the service discovery” SNDS'07, 3rd IEEE International Symposium on Security in Networks and Distributed Systems, May 21-23, 2007, Niagara Falls, Canada.

[16] S. Trabelsi, J.C. Pazzaglia and Y. Roudier “Enabling Secure Discovery in a Pervasive Environment” 3rd International Conference on Security in Pervasive Computing (SPC 2006) – York – UK – April 2006

[17] A. Leung and C. J. Mitchell, “A service discovery threat model for ad hoc networks”, in Proceedings of the International Conference on Security and Cryptography (SECRYPT 2006), Setubal, Portugal, August 7-10, 2006, INSTICC Press, 2006, pp.167-174.

[18] F. Zhu, M. Mutka, L. Ni, “Facilitating Secure Ad-Hoc Service Discovery in Public Environments”, In Proceedings of the 27th IEEE Computer Software and Applications Conference, Dallas, Texas, USA, 2003

[19] S.E. Czerwinski et al, “An Architecture for a Secure Service Discovery Service”, In Proceedings of MobiCom '99, Seattle, WA, August 1999

[20] F. Zhu, M. Mutka, L. Ni “Prudent exposure: A private and user centric service discovery protocol” Proceedings of the 2nd IEEE International Conference on Pervasive Computing and Communications (PerCom’04) Orlando, USA, 2004

[21] Montagut, Frédéric;Molva, Refik “Enabling pervasive execution of workflows” CollaborateCom 2005, 1st IEEE International Conference on Collaborative Computing:Networking, Applications and Worksharing, December 19-21, 2005, San Jose, USA