Volume 1, No. 4, Nov-Dec 2010

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Trust Management Scheme for Service Discovery in Wireless Ad-hoc Network

Prof. S. Sumathy*
School of Information Technology
& Engineering [SITE]
VIT University
Vellore, India
ssumathy@vit.ac.in

N. Kavitha
MS [SE] SITE
VIT University
Vellore, India
kavi.ms99@gmail.com

M. Kumudini
MS [SE] SITE
VIT University
Vellore, India
salairamya@gmail

Abstract: The new emerging technologies in microelectronic and wireless networks have fostered the proliferation of small devices with limited communication and processing power, which are used in ad-hoc environment. The mobile devices in the ad-hoc network communicate via wireless links without the help of any fixed infrastructure. These devices must be able to discover services dynamically and share them safely considering the requirements of ad-hoc networks such as limited processing, communication power, decentralized trust management and dynamic network topology. To overcome the limitations mentioned, a service discovery protocol namely Secure Pervasive Discovery Protocol (SPDP) is used. The proposed work is an enhancement of SPDP, which support to access the services that are provided by this protocol securely. The files can be transferred from one device to another in the network with the help of the system IP address. Blowfish Algorithm is used to enforce trust by encrypting the files before transferring them thus making it in the unreadable format for unauthorized readers. SPDP provides security until service discovery but the proposed enhancement of SPDP provides security in accessing the services and transferring the file securely. The effort of this work is to discover all the services offered by the network and access them safely with necessary security features.

Keywords*:* Ad-hoc network, service discovery, file transfer, trust, Blowfish.

I. INTRODUCTION

In ad-hoc network, which is composed of limited devices, it is essential to minimize the total number of transmissions, in order to reduce battery consumption of the devices. It is also important to detect the availability of services when a device joins or leaves the network. Security in these networks is also critical because there are many chances of misuse from both fraudulent servers and misbehaving clients. In this paper, a new service discovery protocol with security features, the Secure Pervasive Discovery Protocol (SPDP) is proposed. SPDP is a fully distributed protocol in which services offered by the devices can be discovered by others, without a central server. It provides location of trusted services, protection against confidential information, secure communication, identification between devices, and service access control by forming a reliable ad-hoc network [1]. Thus the services offered by the devices in the network can be discovered and accessed securely.

II. EXISTING METHODOLOGY: SECURE PERVASIVE DISCOVERY PROTOCOL

The Secure Pervasive Discovery Protocol (SPDP) is intended to solve the problem of enumerating the services available in single hop ad hoc networks, composed of devices with limited transmission power, memory, processing power, etc. Ad-hoc networks cannot depend upon any single device permanently present in order to act as central server and further, none of the devices present at any moment may be suitable to act as the server.

SPDP merges the characteristics of both push and pull solution to enhance its performance.

[a] The "**Push**" solution, in which a device that offers a service sends unsolicited advertisements, and the other devices listen to these advertisements selecting those services they are interested in.

[b] The "**Pull**" solution, in which a device requests a service when it needs it, and devices that offers those services answer the request, perhaps with third device taking note of the reply for future use.

A device announces its services only when other devices request the service. Service announcements are broadcast to all the devices in the network, all of which will get to know about the new service simultaneously at that moment, without having to actively query for it. SPDP enables to share services safely, through a trust model between devices, which act like its own Certification Authority (CA).

A. *Abbreviations and Acronyms*

Table: I

Term	Definition
SPDP	Secure Pervasive Discovery Protocol
UA	User Agent
SA	Service Agent
SPDP_UA	Secure Pervasive Discovery Protocol User Agent
SPDP_SA	Secure Pervasive Discovery Protocol Service Agent.

B. *SPDP User Agent*

When an application or the final user of the device needs a service, like a specific service or any service offered by the environment, it requests the service from its SPDP UA.

[a] If a specific type of service has been requested:

[i] The SPDP UA searches for that service in the list of local services and if it's in its cache.

[ii] If it is found, the SPDP UA gives the application the service description.

If it is not found, the SPDP UA broadcasts a SPDP Service Request for that service, and it waits CONFIG WAIT RPLY for replies. If no reply arrives, the SPDP UA answers to the application that the service is not available in the network. If some reply arrives, the SPDP UA updates its cache accordingly. In order to minimize the spreading of service announcements of malicious devices, the SPDP UA does not store services offered by untrustworthy devices. Finally, it gives the application the service descriptions of trusted servers received. In order to allow an application to request its SPDP UA to discover all available services in the network, a new type of service called service ALL is introduced.

a) Whenever an application requests for all available services in the network, the SPDP UA sends a SPDP Service Request searching a service type ALL, and waits CONFIG WAIT RPLY seconds for the reply:

[i] If a reply arrives, the SPDP UA updates the cache accordingly and answers to the application listing the local services plus the services in the cache.

[ii] If no reply arrives, the SPDP UA deletes all services stored in the cache and replies to the application listing only the local services. The SPDP UA in all devices is continually listening on the network for messages (SPDP Service Requests and SPDP Service Replies). Whenever a SPDP Service Reply announcing a service is received, the SPDP UA updates its cache accordingly. Moreover, the device's cache has a limited size. When an SPDP UA hears a new announcement but the cache is full, it deletes the service entry offered by the device with less trust degree or less expiration time.

C. *SPDP Service Agent*

The SPDP SA advertises services offered by the device. It has to process SPDP Service Request messages and generate the corresponding SPDP Service Reply.

[a] If necessary, when a SPDP SA receives a SPDP Service Request with a service type different of service ALL:

[i] It first checks whether the requested service, *S*, is one of its local services, or it is in the cache.

[ii] If it is, it generates a random time *t*, inversely proportional to the availability time of the device, *T*. So, the more time the device is able to offer the service, the higher the probability of the device answering first.

[iii] During this time, the SPDP SA listens the network for any SPDP Service Reply of the same request and it updates the cache accordingly.

[iv] When the timer expires, if the SPDP SA knows about some additional devices offering the service S and that have not been announced yet, it sends its SPDP Service Reply.

[A] When a SPDP request for all services (service type ALL) is received, then the SPDP _SA:

[a] Generates a random time *t*, inversely proportional to the availability time of the device, *T*, and to the number of elements stored in the cache of that interface. So, the more time the device is able to offer the service and the bigger the cache, the higher the probability of answering first. We suppose the device with the highest availability time and the bigger cache is the one with the most accurate view of the world.

[b] During the interval *t*, the SPDP SA listens to the network for any SPDP Service Reply of the same request and it updates the cache accordingly.

[c] When the timer expires, if the SPDP SA knows about some new services that have not been announced yet, it sends its SPDP Service Reply, listing the local services and the services in the cache. In certain network technologies, it is possible to detect if a device is switched off or its in roaming in other network. If this happens, its SPDP SA has to send a SPDP Service Deregister, listing all its local services. When a SPDP UA hears this message, it deletes these services from its cache.

D. *Establishing Trust Relation*

To formalize a decentralized trust model there are three properties of the trust relationships. Let A, B and C is the devices in an ad-hoc network. We define the trust relationship function between two devices A and B, R (A, B). R is a continuous function with values between 0 and 1. We use fuzzy logic rather than the usual Boolean logic. 0 and 1 are extreme cases, but values in between are also possible, for example, 0 for distrust, ½ for ignorance and 1 for trust.

[B] The properties of R are the following:

[a] Reflexive: Every device trust on itself (R (A, A) = 1).

[b] Non-symmetrical: If A trusts on B, not necessarily B trusts on A. If we have R (A,B) = b ^R(B,A) = y not equal to b = y.

[c] Conditionally transitive: If A trusts on B and B trusts on C, then A conditionally trusts on C. In mathematic terms: it exists the pairs, R (A; B) = b ^ R (B; C) = y ➔ R (A; C) <=b. y

Initially new devices have no evidence of past experiences to establish an initial trust value for interaction. There are two sources to form an initial trust degree:

Personal opinion (direct trust) or Recommendations (indirect trust).

Direct Trust: The trust value is formed either by getting some information about the device's nature i.e. device type, owner, etc., or by human intervention.

Indirect Trust: It is applied when there exists trust relationships between some devices. The trust value is created from recommendations, which are given by third trusted parties.

System Architecture

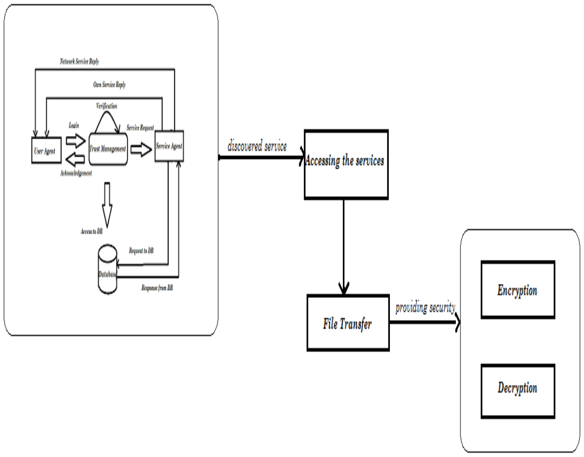


Figure: 1

III. PROPOSED ENHANCEMENT OF SPDP

A. File Transfer

SPDP protocol is mainly for discovering the services that are available in the network, where as the work also support to access the services, which are discovered. The files are transferred from one system to another with the help of the system IP Address. All the devices that are available in the ad-hoc network need not be authorized so there are more possibilities of tracing the files while transferring them. This may lead to the release of confidential information. So to avoid this and send the files safely, Blowfish Algorithm is implemented for transferring the files securely. This algorithm generates a random password with which it encrypts the file before transferring. Hence with the help of Blowfish Algorithm, files can be transferred securely from one device to another in the ad-hoc network.

B. Blowfish Algorithm

The information can be shared between the devices securely using the Blowfish Algorithm.

[a] Design Considerations

[i] Blowfish has a 64-bit block size and a variable key length from 32 up to 448 bits.

[ii] It is a 16-round Fiestel cipher and uses large key-dependent S-boxes.

[iii] Brute-force attack is even more difficult, because of the key length and the time consuming for sub-key generation process.

[b] Design Constraints

There are some constraints for this design, which are as follows:

[i] The input must not be a folder.

[ii] The RAM should not be less than 256 MB.

[iii] Only encrypted file can be decrypted.

[c] Secure File Transfer

The security in file transfer is implemented using the Blowfish Algorithm. This algorithm accepts a file of any type and user's password as input and encrypts the file. The Encrypted files have a new extension, namely ". enc" to distinguish them from other files. Encrypted files are not allowed to be encrypted again for security issues. This system is used to encrypt and decrypts the sensitive information, and makes the data unreadable by the unauthorized readers. The encrypted files can be decrypted with the user's password using the same system. The encrypted file can be deleted once it is decrypted, as per the user's wish because there are possibilities for an intruder to decrypt back the original file if he knows the valid password. The decrypted file can be moved to new destination if required.

A strong Password Generator that can be used while encryption or can be used as separate functionality is developed. The Decrypted files make sense only if the password is correct.

The encrypted file can be sent to the other devices, which are present in the network. All the devices in the network can receive the file, but the device which is having the valid password is considered as an authorized user and only they it decrypt the file to get back the original file. This password can be known to the authorized receiver using telephone or message from the sender.

[d] Methodology

The three functionalities that are used in the Secure File Transfer are as follows:

[i] encrypt File ()

[ii] decrypt File ()

[iii] file Transfer ()

[A] The encrypt File () method takes three parameters namely the file that has to be encrypted, the password used for encryption that is generated randomly, and the destination file name in which it has to be stored. Then the file is encrypted using the blowfish. encrypt () method and saved in the user specified location with an extension of ". enc" to distinguish it from the other ordinary files.

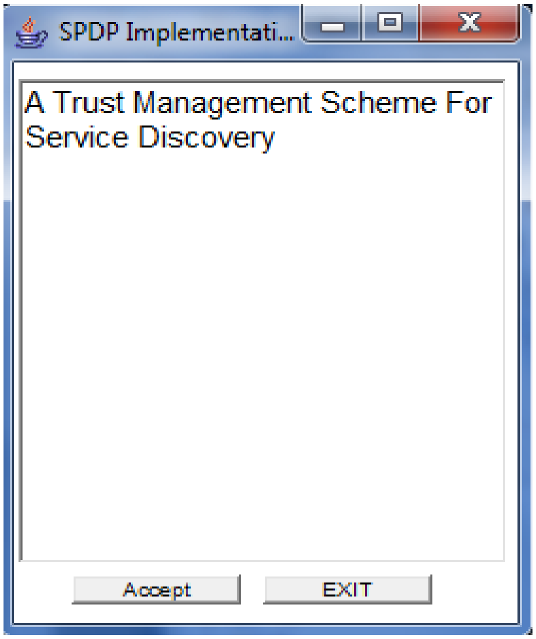
[B] The decrypt File () method takes the encrypted file location with ". enc" extension and the password, which was randomly generated to encrypt that file and the destination file name as the input. With the help of these inputs the method blowfish. decrypt () is used to decrypt back the encrypted file. The decrypted file can be moved to the new specified location.

[C] After encrypting the file, file Transfer () method is initiated for transferring the file using the host IP address. This file transfer is based on the client-server method using sockets. In Exceptional cases, like if the specified file is not found or the file location is not specified correctly then the response will be displayed as File Not Found.

IV. RESULTS AND DISCUSSIONS

The implementation is done with the help of two laptops. Check boxes are available in the sign in page for enabling the user agent, service agent and service de-register. The size of the local cache, remote cache and the availability time of the services are also defined. The local services that are available in the lap can be identified. The services that are added in one lap can also be discovered using the other laptop. These identified services are added in the cache along with their URL. If the cache size is full and a new entry arrives then the cache entry that is made by the device with the least trust degree is deleted.

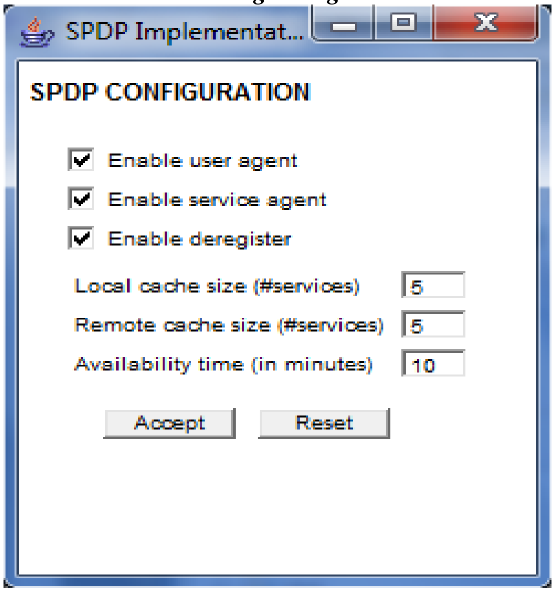
SPDP login page



A Trust Management Scheme For Service Discovery

Figure: 2

Selecting the Agents



SPDP CONFIGURATION

☑ Enable user agent

☑ Enable service agent

☑ Enable deregister

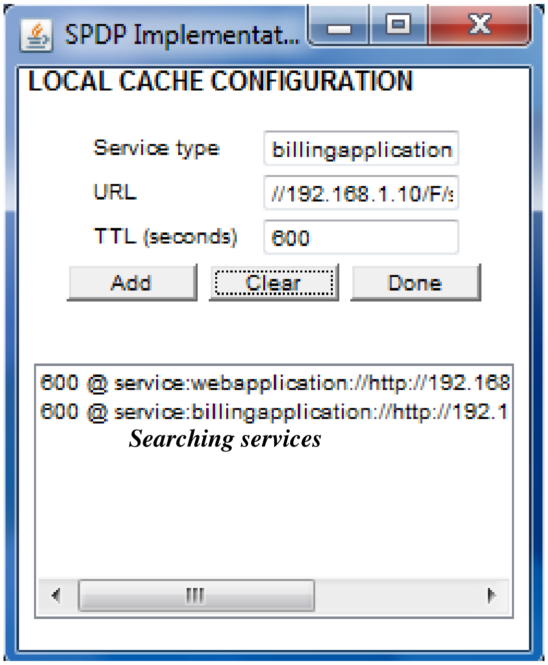
Local cache size (#services) 5

Remote cache size (#services) 5

Availability time (in minutes) 10

Figure: 3

Adding services in cache



LOCAL CACHE CONFIGURATION

Service type billingapplication

URL //192.168.1.10/F/

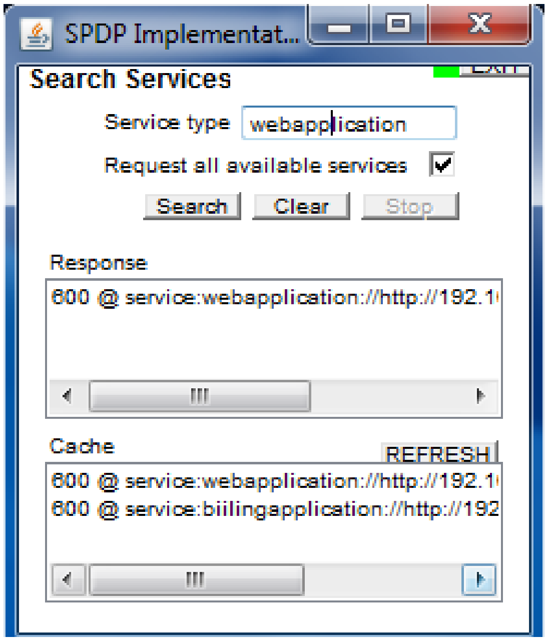
TTL (seconds) 600

600 @ service:webapplication://http://192.168

600 @ service:billingapplication://http://192.1

Figure: 4

Searching services



Search Services

Service type webapplication

Request all available services ☑

Response

600 @ service:webapplication://http://192.1

Cache

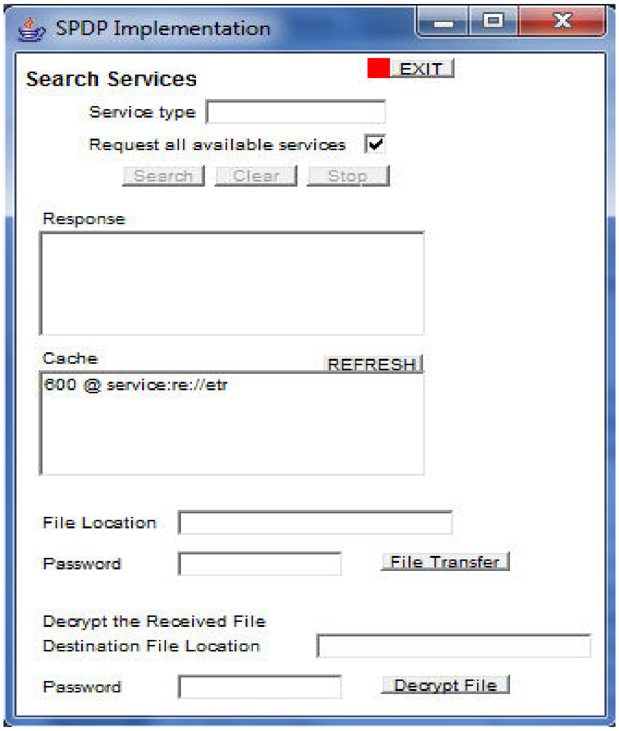
600 @ service:webapplication://http://192.1

600 @ service:biilingapplication://http://192

Figure: 5

The files can also be transferred from one system to another with the help of the system IP Address. To transfer the files securely, files are encrypted using Blow Fish Algorithm. The location of the file must be specified by the user. The password that is used for encryption is generated randomly. The encrypted file will be stored in the location specified. With the help of the application that is running in another lap we can decrypt the file using the same password that is used for encryption. The password can be got from the sender personally.

Secure File Transfer



Search Services

Service type

Request all available services ☑

Response

Cache

600 @ service:re://etr

File Location

Password

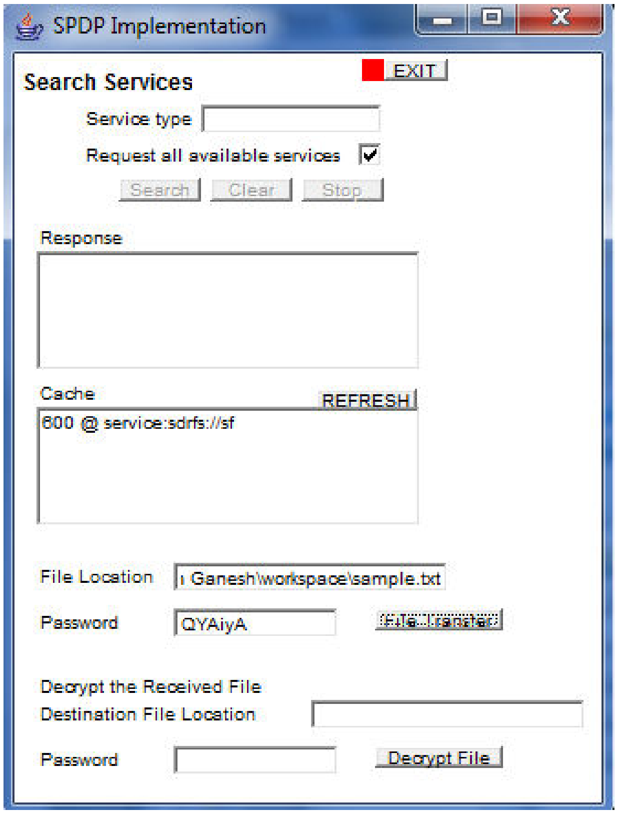
Decrypt the Received File

Destination File Location

Password

Figure: 6

Transferring the Encrypted File



Search Services

Service type

Request all available services ☑

Response

Cache

600 @ service:sdrfs://sf

File Location ı Ganesh\workspace\sample.txt

Password QYAiyA

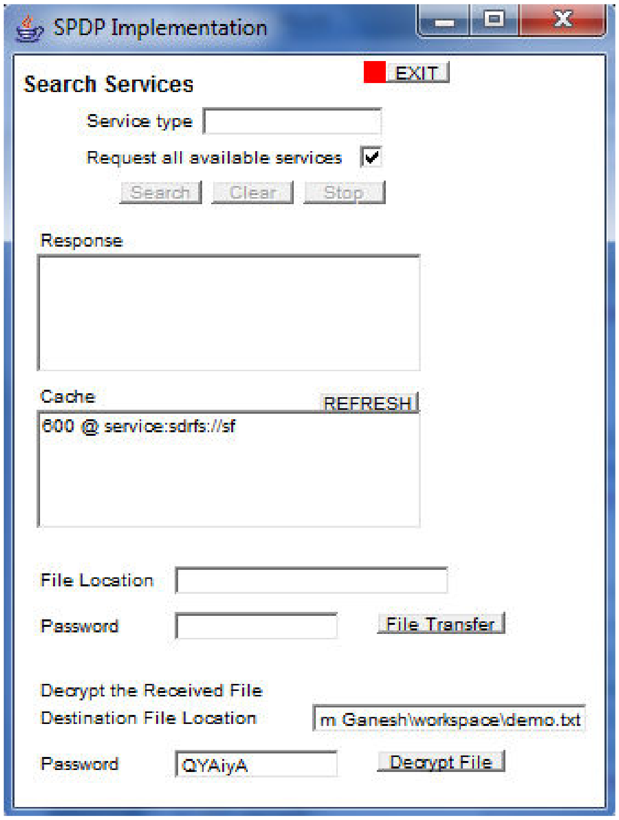
Decrypt the Received File

Destination File Location

Password

Figure: 7

Retrieving the file



Search Services

Service type

Request all available services ☑

Response

Cache

600 @ service:sdrfs://sf

File Location

Password

Decrypt the Received File

Destination File Location m Ganesh\workspace\demo.txt

Password QYAiyA

Figure: 8

V. CONCLUSION AND FUTURE WORK

A Secure Pervasive Discovery Protocol (SPDP) and the software that uses it to discover the services offered in its surroundings is developed in Java. This implementation has been tested successfully in all versions of Windows.

The security support of SPDP has been developed as an independent module, in order to provide security services to other kind of applications. For security, all the functiona lities of Blowfish Algorithm are implemented and tested successfully. This tool can be enhanced in future by adding more features like allowing users to delete encrypted files with authorization because there are possibilities for an intruder to decrypt back the original file if he knows the valid password, and making the user interface more attractive.

Thus, the Services can be discovered safely and dynamically and the Secure File transfer is done using the Blowfish Algorithm in Java. As a future work, this Secure Pervasive Discovery Protocol can be implemented in Java 2 Micro Edition (J2ME) using the Personal Profile of the Connected Device Configuration (CDC). This implementa tion shall be tested in Pocket PC Windows Mobile 2003 devices also. This SPDP implementation is also possible in other devices without support of Java Virtual Machine such as web-cams, to be integrated in a test-bed to obtain results based on real experiments.

VI. REFERENCES

[1] Celeste campo, Florina Almenarez, Damel Diaz, Carlos Garcia-Rubio, Andres Marin Lopez, "Secure Service Discovery based on Trust Management for ad-hoc Networks".

[2] AsadAmirPirzada and Chris McDonald, "Establishing Trust In Pure Ad hoc Networks".
R. Livingstone, 2003.

[3] "A Survey of Service Discovery Protocols in Multihop Mobile Ad Hoc Networks" by Adnan Noor Mian, Roberto Baldoni, and Roberto Beraldi

[4] "Splendor: A Secure, Private, and Location-aware Service Discovery Protocol Supporting Mobile Services" by Feng Zhu, Matt Mutka, Lionel Ni.

[5] "Service Description for Pervasive Service Discovery" by Michael S. Thompson, ScotF. Midkiff, 2005

[6] "Cluster Based Security Scheme for Mobile Ad Hoc Networks" by A. Shajin Nargunam and M.P Sebastian, 2006

[7] "A Survey of Service Discovery Protocols for Mobile Ad Hoc Networks" by Jian Su, Wei Guo, 2008

[8] "Optimized Use of Battery Power in Wireless Ad hoc Networks"by Praveen Gupta, Preeti Saxena, A.K. Ramani, Rajkamal Mittal

[9] S.Kent and R. Atkinson,"Security architecture for the internet protocol (IPSec)", NOV1998.

[10] T. Beth, M. Borcherding, and B. Klein, "Valuation of trust in open networks", in Proceedings of the European Symposium on Research in Computer Security (ESORICS '94, Brighton, UK), Heidelberg, Germany, Nov. 1994, number 875 in Lecture Notes in Computer Science, pp. 3–18, Springer-Verlag.

[11] Matt Blaze, Joan Feigenbaum, and Jack Lacy, "Decentralized trust management",in Proceedings of the IEEE Symposium on Research in Security and Privacy,Oakland, CA, May 1996, IEEE Computer Society, Technical Committee on Security and Privacy, number 96-17, IEEE Computer Society Press.

[12] Alfarez Abdul- Rahman and Stephen Hailes, "A distributed trust model", in Proceedings of the ACM Workshop on New Security Paradigms, Cumbria, United Kingdom, Sept. 1997, pp. 48–60, ACM SIGSAC, ACM Press.

[13] A. Josang and S. J. Knapskog, "A metric for trusted systems", in Proc.21st NIST-NCSC National Information Systems Security Conference, 1998, pp. 16–29.