MWNS 2009

MOBILE AND WIRELESS NETWORK SECURITY

In conjunction with
IFIP Networking 2009

Hakima Chaouchi
Georg Carle
Maryline Maknavicius

Editors

Hakima Chaouchi,
Georg Carle,
Maryline Maknavicius
(editors)

Mobile and Wireless Network Security

MWNS 2009

Shaker Verlag
Aachen 2009

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Printed in Germany.

ISBN 978-3-8322-8177-9
ISSN 0945-0807

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen
Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9
Internet: www.shaker.de • e-mail: info@shaker.de

CONTENTS

PREFACE

It is hard to believe in security when it is so easy to access the communication media such as wireless radio media. However, the research community in industry and academia has for many years extended wired based security mechanisms or developed new security mechanisms and security protocols to sustain this marriage between wireless/mobile networks and security. Note that the mobile communication market is highly growing for different services and not only mobile phone service. That why securing wireless and mobile communications, is crucial for the perennial of the deployment of services over these networks.

Wireless and mobile communication networks have gained a tremendous success in today's communication market both in general or professional usage. In fact, getting communication services anytime, anywhere and on the move has been an essential need expressed by connected people. This becomes true thanks to the evolution of communication technologies from wired to wireless and mobile technologies, but also the miniaturization of terminals. Offering services to users on the move has significantly improved productivity for professionals and flexibility for general users. However, we cannot ignore the existence of important inherent vulnerabilities of these unwired communication systems, which gives the network security discipline a key role in convincing users to trust the usage of these unwired communication systems supported by security mechanisms.

Since the beginning of networking era, security was part of the network architectures and protocols design even if it is considered as slowing down the communication systems. Actually, network security is just a natural evolution of security of standalone or distributed operating systems dealing with machine/network's access control, authorization, and confidentiality and so on. Even though the context has changed from wired to wireless networks, we are facing the same issues and challenges regarding security. More precisely, it is about preserving the integrity, confidentiality and availability of resources and the network. Other security issues more related to the users such as privacy and anonymity are also important from the user's point of view today, especially with the new need of tracking criminals, but here in this book we are concerned only by the network security, and we also included two chapters dealing with important security issues and solutions to secure downloaded applications in the context of mobile operator, and copyright protection by watermarking techniques.

Several security mechanisms have been developed such as authentication, encryption, access control, and others in order to offer secure communications over the network. According to the network environment, some security mechanisms are more mature than others due to the early stages of certain networking technologies

such as wireless networks, ad hoc or sensors networks. However, even with maturity, and even if they are already widely implemented in marketed products some security mechanisms still need some improvement. It is also important to consider the limited resources of mobile terminals and radio resources to adapt wired network's security mechanisms to wireless context. These limited resources have direct impact on security design for this type of networks.

MWNS 2009 is the second event trying to gather security and network experts together to present their research work in wireless and mobile networks security in different network architectures such as mobile, ad hoc and mesh, sensors and wireless networks.

Special thanks to our main sponsors IFIP Networking, Institut Telecom, TUM university, and CNRS Samovar.

Asso.Prof. Hakima Chaouchi, PhD.

Co-Editor of "Emerging wireless technologies" book by Hermes 2008
Co-Editor of "Wireless and Mobile Network security" book by ISTE 2009
Editor of "Internet of Things: technologies and architectures" book by ISTE 2010

MWNS 2009 Committees

Workshop Chairs

Hakima Chaouchi, Institut Telecom, Telecom Sud Paris, France

Maryline Maknavicius, Institut Telecom, Telecom Sud Paris, France

Technical Program Chair

Georg Carle, University of Tubingen, Germany

Publicity Chair

Tara Ali Yahiya, Telecom Sud Paris, France

Technical Program Committee

Tara Ali Yahiya, Telecom Sud Paris, France

Abderrahim Benslimane, University of Avignon, France

Faouzi Bader, CTTC, Spain

André-Luc Beylot, ENSEEIHT, France

Georg Carle, University of Tübingen, Germany

Hakima Chaouchi, Telecom Sud Paris, France

Jean Michel Combes, Orange Labs, France

Vasilis Friderikos, King's College of London, United Kingdom

Ivan Ganchev, University of Limerick, Ireland

Olivier Heen, INRIA, France

Jose Araujo, Alcatel Lucent, France

David Cho, Nanyang Technological University, Singapore

Maryline Maknavicius, Telecom Sud Paris, France

Hasnaa Moustafa, Orange Labs, France

Jose Marcos Nogeira, UFMG, Brazil

Mairtin O'Droma, University of Limerick, Ireland

Kobus Roux, Meraka Institute, South Africa

Franck Veysset, Orange Labs, France

Peter Schoo, DoCoMo Euro-Labs, Germany

Farid Naït-Abdesselam, University of Sciences and Technologies of Lille

Mohamad Badra, CRNS LIMOS Laboratory, France

Providing Identity Assured User Generated Services Using IMS

Seppo Heikkinen

Tampere University of Technology
P.O.BOX 553
FIN-33101 Tampere, Finland
seppo.heikkinen@tut.fi

Abstract. The advent of ubiquitous computing will increase the dynamism in the relationships of the entities. This is especially true, if the visions about the proliferation of the amount of small operators become reality. Also, even though the operators would like to tightly control the service provisioning landscape, it is more likely that the innovative concepts come from external parties. The user themselves might be the service creators as they already have become content generators. However, the service providers still should be compensated for their efforts as this can lead to better quality services. Naturally, some try to fraudulently get services without paying. Thus, non-repudiable solutions are needed that ensure that the services are provided and paid for. In this paper we investigate an architecture that uses IP Multimedia Subsystem (IMS) as service platform to provide user generated services in an assured and secured way.

Keywords: Accounting, hash chains, HIP, IMS, service provisioning

1 Introduction

The current mobile operator landscape is dominated by static relationships between incumbent operators. Operators have created tight agreements about their interaction with the interest in retaining the control of their subscribers. But when we consider IP based services available in the Internet, there is less subscriber control and innovative service concepts can be provided. It has been claimed, though, that some operators try to intervene by, for instance, blocking or degrading the quality of Voice over IP (VoIP) traffic, i.e., in order to prevent the users circumventing the competing call services provided by the operator.

However, operators have also taken steps to provide service platforms of their own in order to compete against the innovative service providers of the Internet. One such architectural initiative has lead to the development of IP Multimedia Subsystems (IMS), which could allow the operators to provide rich services in a flexible and quality assured manner. Even though the architecture is envisaged to provide many different kind of services, even ones provided by external parties, the deployment still relies on the existence of "well known" partners, who provide reliable user and accounting information without any strong protective measures. This does not

promote flexible and dynamic interaction models, especially considering that the research on ubiquitous technologies and ambient networking suggests an increase in the dynamic relationships between various entities. These visions provide enablers that could change the operator landscape in such a way that also small players, even individuals, could act as operators and provide, for instance, access services in easy manner. Naturally, when the dynamism increases, the security issues become even more important, because you no longer can rely on the “good behaviour” of a known and well established operator.

The operators also need to consider flexible service provisioning models. Thus, they should support innovative ideas coming outside their walled gardens. Social media has shown that ordinary users can be content creators, so they might as well excel in service creation. This is already starting to show in the creation of service mashups using simple tools, such as Yahoo Pipes. Additional incentive is given, if the users can be compensated for their efforts.

In this paper we discuss the potential of employing IMS in such an ambient networking environment, where the relationships are based on dynamic interaction and more assurance is needed for the actions taken. We investigate how it would be possible to provide user generated services using the available service infrastructure. Such services could be, for instance, related to streaming media or one could envisage enhancements to the current peer-to-peer interaction (like the piecewise approach of BitTorrent) to ensure fair sharing ratios. Also, VPN like tunnelling services could be provided. Thus, the idea is to sketch enhancement for IMS to allow the users provide services in a flexible way, but also take into account the liability aspects, so that the compensation from the provision of resources can be guaranteed giving more incentive to create attractive services. The operator is included in the initialisation of the service, hence taking into account the interests of operators to retain certain level of control of the actions of their customers, for which they have accepted liability (to a certain limit).

From a technical viewpoint, the solution is based on the idea of running Host Identity Protocol (HIP) on top of Session Initiation Protocol (SIP) and using the IMS home operators to ensure the identity of the communicating end points. The user providing the service receives non-repudiative evidence of the service usage in the form of hash chain tokens, which are strongly bound to the identity of the service consumer. Thus, emphasis is on the secure naming of the entities, so that every party of the transaction, i.e. payers and payees, can be reliably identified.

This paper is organised as follows. In the next section we briefly discuss the related work and technologies employed in our solution concept. The third section sketches the proposed solution on high level. In the fourth section we discuss the characteristics of the system and the fifth section concludes the paper.

2 Related work

Host Identity Protocol is an experimental proposal for introducing an additional identity layer to the Internet stack [1]. This "3.5 layer" enables decoupling the dual nature of IP addresses, so that the end point identity and locator roles can be separated

into two distinct concepts. Thus, better mobility and multihoming solutions can be provided. As the identifiers in this model have cryptographic properties they are able to implement secure naming of entities. HIP functionality is based on a handshake procedure, which uses four messages to authenticate the end point identities and create keying material with the help of Diffie-Hellman exchange for protecting the association between the communicating entities. This base exchange also includes puzzle mechanism to mitigate denial of service concerns.

IP Multimedia Subsystem (IMS) is an architecture designed by 3GPP to facilitate the provision of multimedia services over IP packet networks [2]. Thus, it aims at being a service platform for the next generation networks (and current ones as well), which allows delivering rich services to users with assured quality of service (QoS). It should be noted that even though these services are generally expected to be telecom centric, IMS has the potential of becoming a more general platform for service provisioning integrating various application domains, i.e., in the fashion of service oriented architecture. Another, a more cynical viewpoint might argue that it is just a desperate attempt of telecom operators to retain their walled garden thinking against innovative (and sometimes anarchistic) Internet originated service development. Nevertheless, the decision to allow interaction between operator IMS infrastructure and external service providers is political, as well.

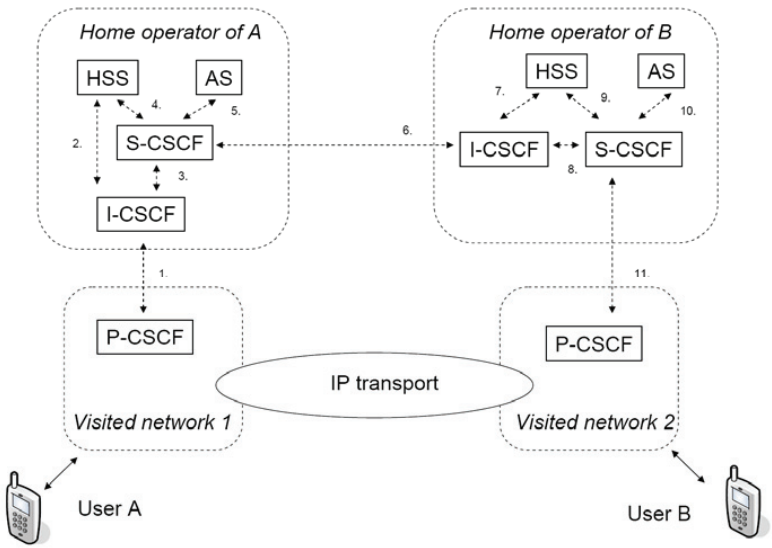


Fig. 1. A simplified view on IMS architecture in terms of session establishment

At the heart of IMS is Session Initiation Protocol, which works as a signalling protocol for setting up and managing the sessions between the parties. Even though IMS is supposed to support variety of different kind of service models, it currently

very much relies on the static interaction with the known partners. In other words, the interacting operators are expected to be trustworthy without any strong technical measures to ensure the data integrity, and application services are tightly connected to the home operator domain. A simplified IMS architecture is presented in Fig. 1 in terms of session establishment, emphasising the hop by hop nature of SIP signalling (different kind of topologies are also possible). It includes different proxy elements (Call State Control Function, CSCF), subscription management (Home Subscriber Server, HSS), and application servers (AS). A detailed presentation of IMS can be found in [2].

Charging in IMS does not contain any strong security mechanisms, as it just uses SIP headers and Diameter parameters to convey information about the relevant events. Thus, no non-repudiation, for instance, is offered and the user is at the mercy of the operator. There is, however, considerable amount of work in the field of micropayments about using hash chains to provide granular payment solution in various contexts. One such is presented in [3], which used KeyNote credentials along with hash chains to implement One-Time Password (OTP) coins, without any strong bindings to the actual traffic, though. Hash chains themselves were already presented by Lamport in 1981 as one-time password mechanism [4]. Our approach is based on the ideas presented with OTP coins, but uses SPKI certificates as an assertion mechanism to provide non-repudiation. [5] already discussed the details of using such system integrated with HIP. In [6] hash chains were used to provide authentication and non-repudiation solution to a system integrating the use of WLAN and 3G network with the help of EAP.

Many other key management solutions are also available, such as IKEv2 [7], but perhaps the closest one for the purposes of this paper would be Multimedia Internet Keying (MIKEY) [8], which is especially suited for SIP scenarios. However, the intention of our solution is also provide identity association establishment and possibility to negotiate additional associations. While MIKEY could be extended to include such functionality, the choice was to go for more identity oriented approach.

2 Scenario overview

The presented scenario builds on the premise of entity identities. In other words, it is expected that every entity, even networks, is in possession of an identifier, for which it is able to provide proof of possession. One such example, used in HIP, is Host Identity Tag (HIT), which is a hashed representation of the public key and has the benefit of providing a concise representation of the identity. Additionally, it is assumed that the operator relationships are dynamic. This follows the line of ambient networking thinking, which suggests that technical development makes it feasible also for the small players to assume the operator role. So, current assumptions of pre-existing static roaming agreements are no longer valid and there is more uncertainty about the trustworthiness of the received data.

The idea of the scenario is to provide user generated services to other users in such a fashion that non-repudiable accounting records can be generated. The previously unknown user identities are vouched for by their respective home operators. With the

employment of the principles of HIP, the users are able to secure their connections and ensure that the service is provided to the correct entity. Note that the users are in possession of both identity layer idenfiers, e.g., HITs and typical SIP layer identifiers.

2.1 Service registration

User B, who wishes to provide services of her own, has basically two options in the architecture we are envisaging: the service is provided directly by the user or it is provided through an external provider, i.e., it acts as a service proxy. In the latter case one can see two further options depending on the association between the user and the proxy. It could be an application server provided by the home operator of B, but it also could be totally independent entity with no direct administrative connection with the home operator, e.g., YouTube kind of entity in the case of streaming service.

In any case, the user is responsible of contacting her home operator in order to update the initial Filter Criteria (iFC), which will dictate the processing of the SIP messages directed to or originating from the user. This way S-CSCF, a central routing entity in IMS, knows how to route the message directed to the service identifier of the service of the user. Service identifier is a typical SIP URI, which is either assigned by the home operator or suggested by the user in accordance with any previous agreements she has with the operator. The home operator should provide a separate registration service for its users that has the possibility of updating iFC accordingly, be it a redirection to a service provided by the user or a proxy server.

In case of an external proxy provider, the user also is responsible of providing the usage offer terms and a delegation certificate, which authorises the provider to act on behalf of the user. The home operator might have a policy that dictates that it has to receive such assertion at the time of the registration, but it is not entirely necessary as the authorisation is more crucial at the time of the negotiation of the service usage. Proper authorisation ensures, though, that no illegitimate redirection requests take place. SPKI certificate naming the HITs of the parties of delegation is used.

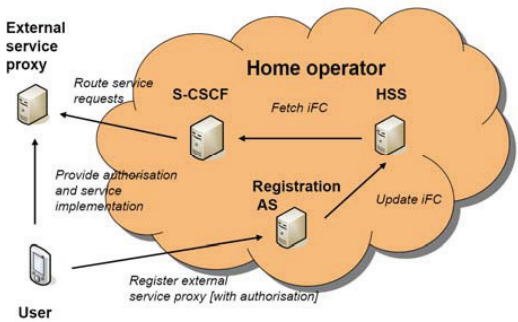


Fig. 2. Interaction of service elements within the context of user generated services

Tasks for the different elements are summarised in Fig. 2. It gives a general idea how different entities interact within the envisaged scenario, which includes the use of

proxy service provider. Within the operator network the elements can use Diameter, whereas subsequent message routing is done with SIP as usual. The registration and interaction with the service proxy can use other protocols, in simplest case it could be HTTP. The user interface for them can be dependant on the implementation, as well.

2.2 Service usage negotiation

In this work it is assumed that the interested user, i.e., user A, is aware of the service identifier. In other words, it is expected that the identity of the service is learnt some out of band means. This could be an entry in an external directory or an advertisement received during interaction with another service, like a related web service.

User A initiates the connection to the service using SIP INVITE semantics. However, this is enhanced with similar functionality as described in [9]. Basically, this means adding HIP related data to the message. Thus, A sends a SIP message, which also has similar content as in HIP R1 message, i.e., identity of the user and cryptographic parameter suggestions for the session. Puzzle mechanism is not included as it is assumed that the home operators are able to protect their own customers from the flood of signalling messages. In a sense, one can view IMS infrastructure as an indirection architecture providing rendezvous between the end entities. Note that if the service provider, i.e., B, would rather assume the responder role and require puzzle solving, it could use provisional response (PRACK message) to switch the roles and send its own R1 instead. We do not, however, consider this case further here.

The home operators are responsible for attaching their own assertions, which ensure the validity of the used identifiers, i.e., SIP URIs and HITs. This includes providing a signature in SIP headers, but could also include an additional certificate in SIP body to restrict the rights of the subscriber. Assertions are needed to ensure the liability of the parties, i.e., end users know that the operators are willing to vouch for the previously unknown identities and the operators know that the other operator is willing to accept the liability of behalf of its own subscribers. Another option would be that the user is in possession of a certificate, which has been previously issued by the home operator and assures the subscriber status of the given user identity. While it provides some performance benefits as the home operator does not need to sign every message, it cannot ensure the authenticity of the other SIP headers and is subject to possible revocation considerations and checks.

In addition to the mechanisms described in [9], user B attaches an offer statement to the "I2" message, which tells what sort of compensation she expects. In essence, this tells the hash chain token release frequency she expects to receive as a proof of the service usage, e.g., per kilobytes or per minute. In case of service proxy, the proxy is responsible of doing this, but it also has to include the delegation certificate issued by the user B as the service offer is bound to the identity of B. "I2" also contains identity and session specific parameters as in typical HIP message. The offer can be an SPKI certificate, much like in [5].

User A binds itself to the service offer by calculating a hash over the request and signing it. Additionally, A creates a hash chain of suitable length by recursive hash operations and includes the anchor value of this chain, i.e. the lastly generated one, to

the signed statement. This is sent back to B (or proxy) in the "R2" message, which concludes the negotiation phase. After this the parties are in possession of each others identities and parameters, which they can use to secure a data traffic session between themselves. The flow of messages is depicted in Fig. 3.

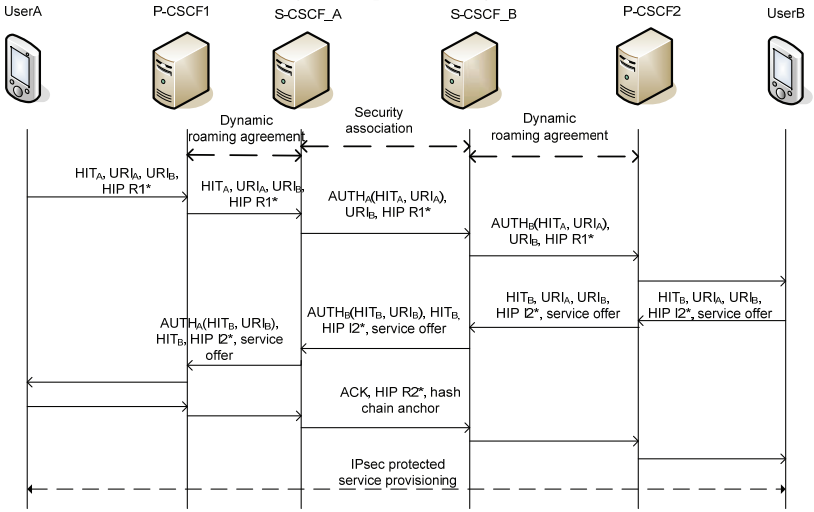


Fig. 3. Signalling flows for establishing an identity association and service usage terms

2.3 Connection establishment

The parties A and B or, in case of proxied service, service proxy and A establish IP connectivity using an association they negotiated. This could be a connection protected by IPsec as in [10] or it could be a stream protected with SRTP [11]. An important thing is that the session keys are derived from the keying material that was created during the negotiation phase using the HIP mechanisms. This way the traffic is bound to the used identities and unauthorised parties do not have access to the provided service.

2.4 Association update

The association needs to be "refreshed" according to the frequency agreed upon during the negotiation phase. In essence, this means sending hash chain tokens, which ascertain to the service provider that the client is still willing to pay for the service. In practise chain values are conveyed using SIP messages using a suitable header extension. While it would be possible to do this directly between the communicating end points, in our approach this is done through the operator infrastructure. This is not more effective, but it gives the operators a chance to record the accounting information as well. Otherwise they might have less incentive to approve the use of

this kind of architecture. This way the external proxy provider has less incentive to cheat, i.e., report faulty usage figures, because the home operator is aware of the amount of tokens used. While the service provider, be it a user or a proxy, keeps receiving hash tokens, it continues to provide the service. If not, then the provision of service is terminated. Similarly, if the service is not received, the consuming user does not provide any additional tokens. SIP flows for providing hash values are depicted in Fig. 4.

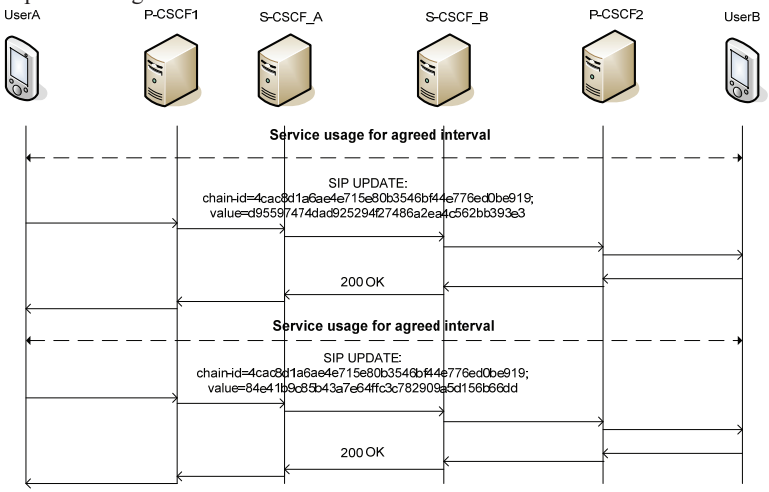


Fig. 4. Submission of new hash chain values

2.5. Compensation

After termination of the service, the service provider needs to be compensated for the provision of its resources. The received hash values (or rather, just the last value and the total amount is sufficient) along with agreement certificates are presented to the home operator and it will compensate its user accordingly. It is assumed that the home operators have agreed on clearing procedures between themselves. Naturally there can be some flat rate agreements, but the accounting can be based on the hash chains, which provide non-repudiable evidence of the service usage. The home operator of A should use this information to bill its subscriber. In case proxy service provider is used, the clearing is done between it and the home operator of B. The delegation provided to the proxy can provide additional information regarding the revenue sharing between the proxy and B or this can have been provided separately during the registration of the service with relevant authorisation.

Accounting is based on the frequency of the hash chain values and they do not exhibit any value as such, but there could be a general agreement about the monetary value of a single unit. This information should be available during the offer and corresponding response phase as it will tell the implication in real world terms. Again, there might be some flat rate agreements, which allow the user to use certain kind of

services for a certain amount (cf. maximum call minutes in certain flat rate subscription plans). However, it is also possible to provide Advice of Charge type of functionality as specified in 3GPP [12]. In practise this would mean providing XML documents to the user specifying the value of a single hash chain value. This will result in lengthy SIP messages, though. One option is also to make the hash chain tokens to correspond to the debit units specified for the IMS online charging [13].

3 Discussion

3.1 Trust and liability

The assumption is that the user trusts her home operator and they have exchanged identities, so the home operator is aware of the user identifiers both on application and identity layers. Such registration could be included, for instance, to the Authentication and Key Agreeement (AKA) procedure, when they both authenticate each other. Note that the user could be using a short term identifier in order to protect her privacy when interacting with external entities. Thus, every statement and assertion is bound to secure identifiers, for which is it possible to provide proof of possession.

As described in [9], the identity needs to be bound to the relevant SIP headers and message content. This takes place with the SIP Identity mechanism described in [14], although with the modification that an additional pair of identity headers is created for expressing the end entity identity as well as the corresponding signature. Additionally, the SIP level identity needs to be explicitly expressed with an additional header value, so that the both parties get the notion of each others identity, which they plan on using. An example of the relevant headers is given in Table 1.

Table 1. SIP header values used for conveying identity information

Header	Example value	Explanation
P-End-Pub-Identity	<sip:userA_public@home.net>	Public SIP identity of the message sender
P-End-Identity	"Ccp+C2..<clipped>..zmJp CB7rBXGe+DnutU="	Signature created by sender (base64)
P-End-Identity-Info	<urn:hit:8dc49622d9be6fca7f1ecb8f3e6738e2>; alg=rsa-sha1	User identity used for signature creation
Identity	"ZYNBbHC..<clipped>..PKb fU/pryhVn9Yc6U="	Signature created by the operator (base64)
Identity-Info	<urn:net:homeA: aad1d9518a9bde5b8f3b5c6b59b6970e>; alg=rsa-sha1	Operator identity used for signature creation

In our scenario it is also assumed that the home operators have established an association. This entails securing their connectivity, but it also has established an agreement about the liability constraints. In other words, they are willing to guarantee the costs generated by their own subscribers in interaction with the entities of the other home operator. This creates the basis of trust the home operators have to each

other's assertions. Thus, if home operator A asserts that user A is its subscriber, it also states that the costs of user A will be guaranteed to the negotiated limit. As the user B trusts her home operator B, she knows that the operator B will be liable for any actions of the entities it has asserted.

The agreement between operators A and B can be pre-established like nowadays or it can be created on the fly. The latter one requires dynamic roaming agreement procedures, which are discussed, for instance, in [15]. However, trust based on liability is in effect here as well. Unless an operator has some other knowledge about the identity and behaviour of the other operator, additional evidence is required. This could be based on an assertion of a financial institution or an operator organisation, like GSM Association.

3.2 Non-repudiation and accounting

The non-repudiation property of the proposed architecture comes from the combination of the used identities and the hash chains. When the service provider has bound its identity through signature to the service offer, it cannot claim larger compensation at later point. The user of the service binds her identity to the offer and the hash chain anchor, which is a first value of the chain she uses to provide evidence of the service usage. Due the assumption of a secure hash function, the user is the only one, who is able to create such a chain, i.e., hash function is irreversible (preimage resistance). Because the respective home operators have expressed their trust on the used identities, the end entities have a notion of trustworthiness of their communicating partner.

Use of incremental payment in the form of hash chains also allows the parties to react to any discrepancies in the service provisioning and usage. If the service provider does not keep receiving hash chain values at agreed intervals, it can stop providing service. On the other hand, if the service is not provided as agreed, the user can stop providing additional values. Thus, if there is evidence of service usage in the form of hash token, the service has been undeniably used.

Incremental solution also allows the home operator to keep track of the costs generated by the user. If the user exceeds her credit limit, the operator can terminate the service usage by preventing additional tokens and notifying the parties accordingly. Additionally, the home operator gets assured accounting information, which it can relate to the charging information coming from the visited operator. This can be beneficial in a case, where, for instance, the IP connectivity of the user is provided directly by the GGSN of the visited operator instead of tunneling the traffic first to home operator (as currently often can be the case). This way visited GGSN can more reliably charge for changed quality of service of the visiting user, e.g., because certain service requires better QoS. IMS originally is intended for this kind of service dependant QoS treatment, but operators have been reluctant to let go of the tight control of their customers. It should be noted, though, that colluding users can bypass the operator and provide no accounting figures from their service usage, but then they have some external notion of each others trustworthiness and it would be hard to prevent anyway. They do not enjoy the potential QoS benefits, though.

3.3. Challenges

While we envisage more dynamic networking and operator landscape, the suggested solution still has some assumptions about the parties and their network topologies. Denial of service can be one major concern, if external parties are able to inject or modify traffic without restrictions. Even though we suggest use of HIP, we have stripped off the puzzle mechanism in favour of not introducing additional roundtrips to the typical SIP INVITE flows. The service provider needs to do signature validation and setup a state for service negotiation. Thus, operator assistance is needed in order to prevent message floods targeted at crippling the end entity. However, as it is also expected that the entities have already registered with their home operators and relevant signalling associations have been created, there is already some security present to secure the connections, e.g., secure attachment solution discussed in [16].

The proposal also sets additional performance requirements for the home operators as they have to create signatures in order to authorise identities. In high volume traffic signing every message would be quite unacceptable. There is the option of using pre-issued certificates as mentioned above, but not all the messages need this kind of service, only the initial service negotiation invocation. Those transactions that need to use assured accounting can be directed to an application server that can take care of the message processing. In other words, S-CSCF does not need to concern itself with signing operations. Using additional application server also allows parallel processing in the case, where the operators have to first create a dynamic roaming agreement, before any further interaction can take place. This negotiation actually is likely to dominate the performance impact over signature generation.

A potential complication is also the length of the messages. As the negotiation procedure contains both SIP and HIP specific data, it is not possible to fit that to a single UDP datagram in every case. According to SIP specifications, the implementations then have to switch using TCP. This has a slight performance penalty [17], but can also be additional problem if Network Address Translation (NAT) is in place. Thus, NAT traversal techniques would be needed. However, if all network elements were HIP enabled, SIP layer could use HITs instead of IP addresses as contact points as done, for instance, in [18]. In large messages, SIP content redirection mechanism for fetching assertions might be one option, as well.

4 Conclusion

In this paper we have sketched an architecture for providing user generated services with the help of IMS. It employs the principles of HIP and expects every entity to be in possession of a secure identifier. This allows secure naming of participants and also presents an accounting solution, which can be used to provide non-repudiable evidence of the service usage. As it uses hash chains, it is able to provide granularity to the service provisioning so that in case of misuse service interaction can be terminated and further waste of resources prevented.

Because of all the assumptions it may not be realistic to expect this kind of solution to enjoy deployment anytime soon. Hence, it is targeted towards future networks, which exhibit more ubiquitous and dynamic characteristics. However, this does not mean that parts of this work, like allowing user generated services within IMS framework with appropriate authorisation, would not be a feasible approach for current development as well. The worth of IMS is, after all, dictated by the richness of the provided service portfolio.

5 References

1. Moskowitz, R., Nikander, P., Jokela, P. (Ed.), Henderson, T.: Host Identity Protocol. IETF RFC 5201. Apr 2008
2. 3GPP: IP Multimedia Subsystem (IMS). 3rd Generation Partnership Project Technical Specification. TS23.228 V8.1.0. June 2007
3. Blaze M. et al. TAPI: Transactions for Access Public Infrastructure. In: Personal Wireless Communications. Sep 2003
4. Lamport L.: Password authentication with insecure communication. In: Communications of the ACM, vol. 24, no. 11. 1981
5. Heikkinen S.: Non-repudiable service usage with host identities. In: Second International Conference on Internet Monitoring and Protection. Jul 2007
6. Yang, C., Yang, Y., Liu, W.: A Robust Authentication Protocol with Non-Repudiation Service for Integrating WLAN and 3G Network. In: Wireless Personal Communications, vol 39, no. 2. Oct 2006
7. Kaufman, C.: Internet Key Exhange (IKEv2) Protocol. IETF RFC 4306. Dec 2005
8. Arkko J., Carrara E., Lindholm F., Naslund M., Norrman K.: MIKEY: Multimedia Internet KEYing. IETF RFC 3830. Aug 2004
9. Heikkinen, S.: Establishing a Secure Peer Identity Association Using IMS Architecture. In: Third International Conference on Internet Monitoring and Protectionz. Jul 2008
10. Jokela, P., Moskowitz R., Nikander P.: Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP). IETF RFC 5202. Apr 2008
11. Tschofenig H., Shanmugam M., Muenz F.: Using SRTP transport format with HIP, IETF Internet-Draft draft-tschofenig-hiprg-hip-srtp-02, expired. Oct 2006
12. 3GPP: Advice Of Charge (AOC) using IP Multimedia (IM) Core Network (CN) subsystem. 3rd Generation Partnership Project Technical Specification TS24.647 V8.0.0. Sep 2008
13. 3GPP: Telecommunication Management; Charging Management; Diameter charging application. 3rd Generation Partnership Project Technical Specification TS32.229 V8.4.0. Sep 2008
14. Peterson, J., Jennigs, C.: Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP). IETF RFC 4474. Aug 2006
15. 3GPP. Network Composition Feasibility Study. 3rd Generation Partnership Project Technical Report. TR22.980 V8.1.0. June 2007.
16. Heikkinen S.: Security and Accounting Enhancements for Roaming in IMS. In: 6th International Conference on Wired / Wireless Internet Communications. May 2008
17. Nahum E.M., Tracey J., Wright C. P.: Evaluating SIP Proxy Server Performance. In: 17th International workshop on Network and Operating Systems Support for Digital Audio & Video. Jun 2007
18. So J.Y.H., Wang J., Jones D.: SHIP Mobility Management Hybrid SIP-HIP Scheme. In: Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. May 2005

Towards A General System for Secure Device Pairing by Demonstration of Physical Proximity

Yasir Arfat Malkani, Dan Chalmers, Ian Wakeman and Lachhman Das Dhomeja

Department of Informatics, University of Sussex
BN1 9QJ, Brighton, UK
{y.a.malkani, d.chalmers, ianw, l.d.dhomeja}@sussex.ac.uk

Abstract. Co-location of devices is a useful basis for access control policies for ad-hoc connections, as physical security, visibility and social norms provide reassurances to the device owners and participants. There are various possible techniques for demonstrating co-location through physical interactions, which others have started to explore. In some cases these provide the basis for encryption, in others simply confirmation of presence. In all cases these techniques are dependent upon hardware capabilities, offer varying physical scope and levels of attack resistance, and require different levels of user attention and visible public action. Different trade-offs amongst these considerations are desired in different situations. In this paper we present a framework for negotiating such pairings. This facilitates device identification, matching of pairing techniques to requirements, chains of communication to bridge between devices of different capability and improved security by combining techniques where possible.

Keywords: Authentication, security, co-location, discovery, pairing.

1 Introduction

Devices offer services. Device owners are willing for the devices of other people in the same location to use their device's services. How can we prove that these devices are co-located? How can we choose the most appropriate method(s) to prove co-location? There have been many recent proposals to provide secure device pairing [1-8] all varying in their security against different attacks, the needed hardware capabilities and the necessary level of user attention. In a world of heterogeneous devices and requirements, we need mechanisms to allow automated selection of the best protocols without requiring the user to have an in-depth knowledge of the minutiae of the underlying technologies. In this paper, we describe such a mechanism.

As motivation, let us introduce Angela, who is working in a well reputed organization. She organizes a meeting with representatives of some customers to give them a confidential briefing about a new product that her company is launching in near future. The meeting is organized in a hotel equipped with modern smart devices, but which is unfamiliar to Angela. On the meeting day, Angela is getting late, so she leaves her office in hurry and forgets to print some important documents required during the meeting. When she reaches the hotel, she wants to pair her laptop with a

nearby printer to print the documents, without having to gain special permissions on the hotel network or pass files to a receptionist. That she has been allowed into the room with the printer is sufficient credentials. Next she goes to the meeting room, where she wants to pair her laptop with the projector *securely*, since the presentation carries some sensitive data. In addition to preventing eavesdroppers on a connection expected to last for several hours, Angela's laptop selects a mechanism that allows her to demonstrate to the room that the data is coming from her laptop. After her meeting and before leaving, she needs to discuss a confidential issue with her boss. At this time, she wants to pair her Bluetooth enabled headset with her mobile phone. Finally when she finishes everything and needs to leave the hotel, she wants to provide the hotel with a signature stored on her work smart-ID card to use in authenticating their invoice.

The scenario presented above embodies common problems in pervasive computing of ad-hoc interactions with unfamiliar devices and institutions, but can also make use of physical presence. It gives rise to two major concerns regarding the pairing process. First is how Angela makes sure that no one else can modify or read the sensitive data sent to the various devices. This requires setting up of keys for encryption, but also correct device selection in an unfamiliar environment. Second, while pairing the devices she needs to discover which pairing processes can be applied in each situation. To the best of our knowledge, there is no any existing secure pairing system that best fits in all four situations of the scenario. For example accelerometer based techniques are not practical for large devices, in a large room with a roof mounted projector radio signal and close-range techniques are likely to fail. Where a choice of pairing techniques is available not all users will be able to judge which one is the best to use. Further, a pairing system must not increase the complexity and the cost of the devices by requiring expensive dedicated hardware in all devices, but should accommodate the existing capabilities of the pairing partners and should be flexible enough to accommodate future technologies. We believe that a general pairing infrastructure for smart spaces can improve the security and usability of the pairing process. Our proposal is an attempt to integrate pairing schemes in a single model that facilitates association of any pair of devices in several situations by using their common co-location capabilities, and also to relieve user from choosing between dozens of pairing schemes.

The proposed architecture consists of two functional components: co-location servers and devices. Devices register their capabilities with an easily found database stored on the co-location server. When two devices need to associate, the client can query the co-location server to discover and acquire the required information to initiate a secure pairing with the target device. Different interactions to demonstrate proximity are possible and the selection requires consideration of the level of proximity required, the ease with which the interaction can be mimicked by an impostor, the availability of matching sensors to work with, the longevity of the association, and the desirability of the interaction being public. Based on the information from the co-location server, both the client and resource mutually execute a common co-location protocol. This protocol will involve the generation of a key from interaction with the environment – a successful pairing will arise when matching keys are generated. The selected interactions will generate an appropriate key for the nature of the intended association.

2 Background

The problem of secure device pairing continues to be a very active area of research in pervasive computing environments. The issue got significant attention from many researchers, after Stajano and Anderson in their seminal paper [9] highlighted the challenges inherent in secure device association. Their work [9, 10] has been considered as the first effort towards secure transient association between devices in ubiquitous computing environments. They proposed a master-slave model which maps the relationships between devices. The pairing process is done by agreeing a secret key over the physical connection (such as using a cable). Though the secret key is transferred in plain-text and cryptographic methods are not used, it is susceptible to dictionary attacks. In reality, it is also difficult to have common physical interfaces in all the devices, and carrying cables might not be feasible all the times. Balfanz et al. [2] extended Stanjano and Anderson's work and proposed a two-phase authentication method for pairing of co-located devices using infrared as a location limited side channel. In their proposed solution, pre-authentication information is exchanged over the infrared channel and then the user switches to the common wireless channel. Slightly different variations, of Balfanz et al [2] approach, are proposed in [4, 6, 11, 12], which also use location limited side channel to transfer the pre-authentication data. The common problem with these approaches is twofold: first, they need some kind of interface (e.g. IrDA, laser, ultrasound, etc) for pre-authentication phase and are vulnerable to passive eavesdropping attack in the location limited side channels, e.g. two remotes and one projector. Some location limited side channels, such as infrared and laser, are highly vulnerable to denial of service (DoS) attack. Some other pairing schemes including Bluetooth require the human operator to put the communicating partners into discovery mode. After discovery and selection of a device, the channel is secured by entering the same PIN or password into both devices. Although it is a general approach, it gives rise to a number of usability and security issues [13, 14]. For example, a short password or PIN number makes it vulnerable to dictionary or exhaustive search attacks. Further, in Bluetooth pairing an adversary can eavesdrop to break the security from a long distance using powerful antennas.

Recently proposed schemes [1, 5, 7] use audio and/or visual channels for a secure pairing process. Seeing-is-Believing (SiB) [5] uses two dimensional bar codes for exchanging security relevant information between the devices; while the Loud and Clear [1] system exploits annunciated nonsensical sentences corresponding to a shared key. Both of the schemes suffer from a few problems, such as SiB requires that one of the peers must be equipped with camera; while in Loud and Clear a speaker is required. Camera equipped devices are usually prohibited in high security areas; while the latter is not suitable for hearing-impaired users. Further, bar code scanning requires sufficient proximity and light in SiB; while Loud and Clear places a burden on the user for comparison of audible sequences. An adversary can easily subvert bar code stickers on devices in SiB; while ambient noise makes authentication either weak or difficult in Loud and Clear scheme. Saxena et al. [7] extended the work of McCune et al. [5] and proposed a scheme, which requires one device to be equipped with a light detector or a camera and the other with a single LED. When the LED on the device blinks, the other device takes a video clip. Then, video clip is parsed to

extract an authentication string. This scheme has many of the limitations as SiB, such as requiring close proximity and a camera. More recently, the idea of shaking the devices together to pair them has become more common. In this approach two devices are hold and shaken together simultaneously, common readings from the embedded accelerometers in the devices are used to pair them together. Smart-its-friends [15] was the first effort towards this approach. The follow-on method to Smart-its-friends is shake well before use [8]. Mayrhofer and Gellersen extended the Holmquist et al. approach and proposed two protocols to securely pair the devices. Both of the proposed protocols exploit the cryptographic primitives with accelerometer data analysis for secure device-to-device authentication. Shaking the devices together is always not possible, since there is large variety of devices, such as printers, projectors and laptops that can not be shaken.

In contrast to all of above approaches, Varshavsky et al. [3] proposed Amigo [3] system, which exploits the knowledge of common radio environment of communicating partners to securely pair the two co-located devices. Since Amigo exploits Diffie-Hellman key exchange method with the addition of a co-location verification stage, it is computationally not feasible for many devices in pervasive computing environments. Further, there may be many pervasive computing environments where wireless communication is not in use, where the radio data is not available to process or where the wireless network is easy to eavesdrop on while remaining hidden.

In summary, no one has yet devised the perfect pairing protocol. Pairing protocols vary in the strength of their security, the level of required user intervention, their susceptibility to environmental conditions and in the required physical capabilities of the devices. In the remainder of this paper, we show how different protocols can be integrated within a general architecture for proving co-location, which is sensitive to the trade-off amongst the identified strengths.

3 System Architecture

Figure 1 illustrates the high level architecture of the proposed system, and figure 3 shows a more detailed sequence diagram of communications used in the proposed system. Devices move between four states: *initialization and registration, device discovery, authentication and paired*. In the *registration* process, the device generates capability information to send to the co-location server. Thus, each device becomes a visible part of the system and can benefit from any other legitimate device in the system by creating an association with it. After *registration*, a client moves into the *discovery* state. The client searches for pairable devices in the vicinity during this state by querying the co-location server. The latter performs a match-making process based on the client's query. It produces communication and co-location capability information based on common capabilities of both client and matching device(s) (resources). The co-location server provides this information to both devices to smoothly derive the operations of subsequent *authentication* state.

Once the device enters the *authentication* state, the received information from the co-location server is used to execute a common authentication scheme. Finally, if the client is successfully authenticated, it enters the *paired* state. During the *paired* state,

the client periodically enters an evaluation process, where the expiry condition of the given credentials is tested. Based on the outcome of the evaluation process, the client could either remain in a *paired* state or the given credentials are revoked. In the remaining part of this section, we will discuss the design details of our proposed system.

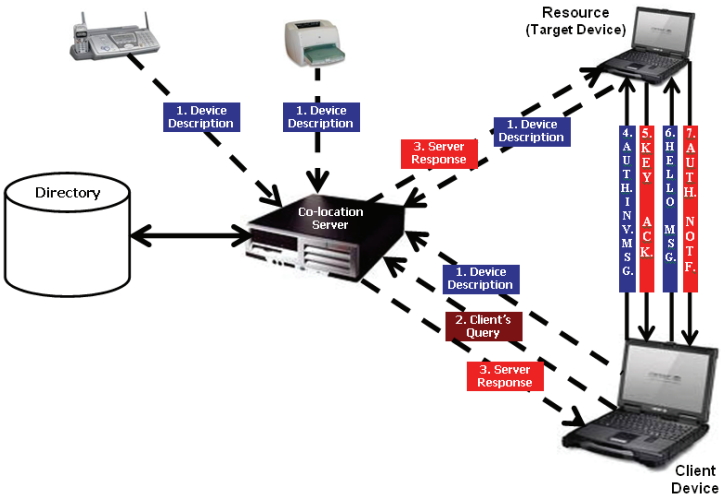


Fig. 1. High level architecture of the proposed system

3.1 Bootstrapping and Registration

Bootstrapping in our model refers to the system initialization and advertisement of co-location servers. Devices discover the co-location servers for registration by listening to a multicast address. A co-location server periodically multicasts its address, so that devices can find it and so register. During registration, the device component is responsible for providing its capabilities in XML form to the co-location server to store in the directory.

The co-location server might run with other local services (e.g. DNS, print) to limit the deployment costs. We are considering all the devices registered with the same co-location server as potentially co-located. Each co-location server is responsible for handling a particular domain, but it is possible that these will overlap or that an impostor might run a server which fails to provide matches as a denial of service attack (we return to security later). These problems can be overcome by performing a search in parallel on all available servers, prioritizing those that provide successful matches in future. A combination of fine-grained deployment of servers, located access (through network schemes) and location services are expected to locate the various devices in the system. Typical semantics of these interactions will involve searching for devices "within x meters", "the nearest", "the device labeled y", or "a device in location labeled z" (where the label is provided by the user). None of these

mechanisms is fool-proof and require open access to location systems, user input, or scanning location tags in addition to the system described here. The process of co-location will allow users to reject a choice and get the next alternative – and, of course, to verify that the device is the one they believe it to be.

A problem arises when a registered device (either in paired or unpaired state) moves out of the domain of its current co-location server without performing de-registration with the existing co-location server or before the expiry condition of its registration. Un-pairing will be handled by the paired devices maintenance arrangements, as they may move together – so the pairing correctly does not require the co-location server to continue. De-registration is required to avoid clients attempting to pair with devices which are no longer present. Explicit de-registration is hard to ensure. Expiry will also be provided, but requires a traffic overhead / timeliness trade-off. Where multiple co-location servers have a trust relationship new registrations may cause speculative de-registrations in adjacent domains to smooth the hand-over process. Finally, the server may need to offer an alternative match where a device is no longer available.

3.2 Device Discovery

Discovery mechanisms play an essential role in ad hoc communications. Several discovery protocols have been proposed to facilitate dynamic discovery of services/devices. Some well known discovery protocols include Service Location Protocol (SLP), Secure Discovery Service (SDS), Bluetooth Service Discovery Protocol (SDP), Microsoft's Universal Plug and Play (UPnP) and Jini, Sun's Java-based approach. Each has its own design considerations. For example, SLP and UPnP are designed for TCP/IP networks; while SDS and Jini are restricted to Java applications, and SDP supports only Bluetooth device/service discovery. Detailed comparisons of discovery protocols can be found in [16-18]. Here one can argue that our approach resembles Jini. As a matter of fact, security has not been major goal/objective of Jini and it is based on Java; so, it supports the same weak/light security mechanism as Java offers. Further, non-encrypted Remote Method Invocation (Java RMI) is used for all the communication in Jini that makes it susceptible to eavesdropping, and also Jini does not support resource (service) side authentication. Moreover, in Jini when a client-device wants to create association with the resource-device, the object/programming code is downloaded from the Jini Lookup Table, which is used to pair the devices. This mechanism also introduces a security risk in pairing model as one can launch/put malicious code in the Lookup Table. Service discovery protocols are not the focus of this work, so to simplify analysis of the problem, we decided to focus on our requirements independent of existing technology. After we have proved our solution, we shall incorporate functionality back into existing protocols such as SDP or UPnP if appropriate.

For our initial tests, we used XML to describe the registration and discovery messages mechanism in the proposed architecture. It is portable and flexible enough that we can easily incorporate additional features in the discovery process. Figure 2 shows the XML based device description template and its corresponding DTD document.

During discovery, the device component is responsible for sending an XML-based query to the corresponding co-location server in order to find the required device. When the co-location server receives the client's query, it goes through a match-making process to find the possible matching device(s) in the domain. As a result, if the matching process succeeds, the co-location server generates an XML document with the required information in order to send it to the client and resource for subsequent authentication process. If a compatible device[1] doesn't exist, then the co-location server recommends any possible device(s) by relaxing the strict condition of common co-location capabilities and leaves the client to decide whether to create an association using third party support. If, even after relaxing this condition, there isn't any matching device, the co-location server simply sends a "device not found" message to the client.

(a)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE colocation [
 <!ELEMENT Device (SemanticName,SemanticLocation,CommProtocol,
ColocCapability>

<!ELEMENT SemanticName (Name)>
   <!ELEMENT Name            (#PCDATA)>

<!ELEMENT SemanticLocation (Name)>
   <!ELEMENT Name            (#PCDATA)>

<!ELEMENT CommProtocol(Name,Address)>
   <!ELEMENT Name            (#PCDATA)>
   <!ELEMENT Address         (#PCDATA)>

<!ELEMENT ColocCapability(Name,Type)>
   <!ELEMENT Name            (#PCDATA)>
   <!ELEMENT Type            (#PCDATA)>
]>
```

(b)

```
<Device>
    <SemanticName>
          <Name></Name>
    </SemanticName>

    <SemanticLocation>
          <Name></Name>
    </SemanticLocation>

    <CommProtocol>
          <Name> </Name>
          <Address></Address>
    </CommProtocol>

    <ColocCapability>
          <Name></Name>
          <Type></Type>
    </ColocCapability>
</Device>
```

Fig. 2. (a) DTD for device description (b) XML-based device description template

3.3 Authentication

Authentication is an important part of the pairing process, as it becomes the basis of a secure association between the client and target device. If the authentication process/scheme is weak, then the user can not trust (from security point of view) the pairing system as a whole. In this process, devices exploit the common information received from co-location server to mutually agree on a scheme to generate a key and execute the authentication operation. We are considering a symmetric key to create secure encrypted channel between the devices. Currently, devices generate a key from the data acquired from sensors as suggested by the co-location server during

[1] A pair able device that supports some common co-location capabilities as client for proving its physical existence in the same proximity.

discovery process. Sensors and a key generation algorithm for the devices are selected based on the received information from the co-location server.

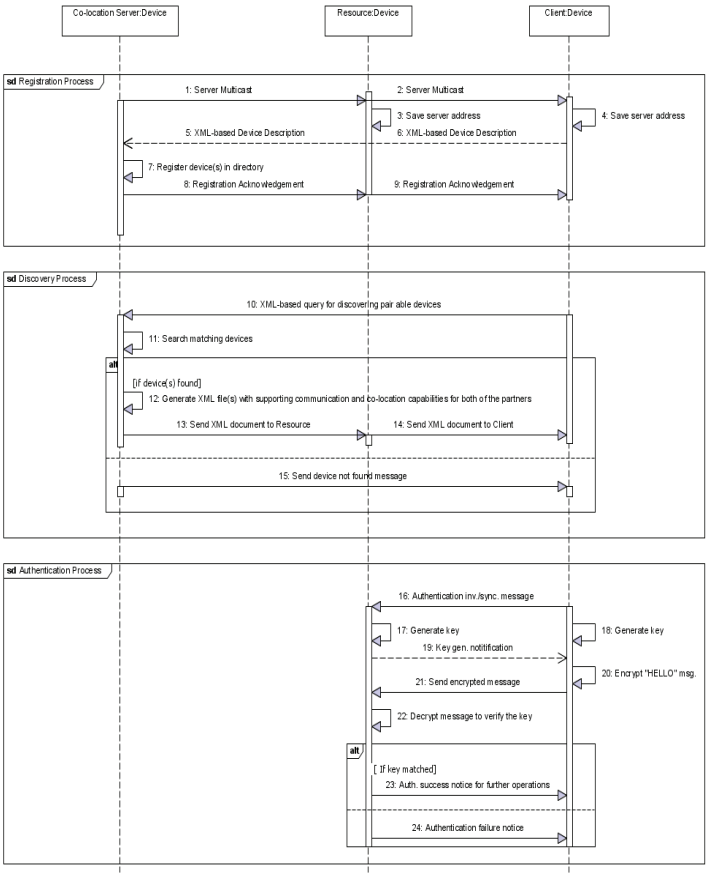


Fig. 3. Message sequence chart describing the communication pattern for the proposed system

The client-side device component establishes a connection with the intended resource using the communication channel, as described in the received XML from co-location server. Once connection acknowledgement from the resource is received, it sends an authentication invocation message to the resource in order for synchronization and key generation operations to commence. Sensors with same or equivalent capabilities (as recommended by co-location server) on both devices acquire the data from local environment. An encryption key is derived from the collected data samples. When the client receives the key generation completion acknowledgement from the resource, it encrypts a “HELLO” message and transmits it to the resource. The resource decrypts the received message. If the decrypted message

is recognized by the resource, the client is authenticated and both the devices enter into paired state. Devising encryption algorithms and generating keys from sensor data is not the focus of our research, so we shall not discuss this further here.

3.4 Security Analysis

Like the schemes for device pairing we build on, we make an assumptions that physical presence and visible actions meet the real access control requirements of the kind of ad-hoc situations described. The devices involved can make use of common sensing capabilities to generate acceptable, strong keys without exposure to third parties or administrators' intervention.

Prior work for device pairing varies greatly in the assumptions about device capabilities, user competence and involvement, as well as security considerations. Understanding the details of various attacks/vulnerabilities in wireless communication is very important in order to determine an appropriate defence strategy for the pairing process. The most significant risk in short range wireless communication (e.g. 802.11, Bluetooth, etc) is that the underlying communication channels are open to everyone including bona-fide users as well as intruders, and thus these cannot be physically secured the same way as a wired network. For example, 802.11 standard uses an encryption system called Wired Equivalent Privacy (WEP). WEP has known vulnerabilities [19], such as it is susceptible to attacks on data and as well as user authentication. These weaknesses allow an intruder to both inappropriately intercept data and also gain access to a network by impersonating a legitimate user. In the case of Bluetooth, devices operate on the 2.4 GHz ISM band. Each Bluetooth device has a unique address, which gives some trust/confidence to user in the identity of the device during association process. For Bluetooth devices to securely associate, an initialization process uses a PIN based approach. Although, the Bluetooth security architecture is relatively secure, it has been vulnerable to key spoofing, address spoofing and PIN cracking [13, 20]. Other threats for wireless communication include well known Man-in-the-Middle (MiTM) and Denial-of-Service (DoS) attacks.

The main goal of an adversary attacking an association model is to fool the legitimate device to associate with adversary's device. Since we are proposing a system for secure device association in close proximity, the threat model considers co-location as the main property to establish a secure channel between two devices. We define the model as follows: two devices that are registered with the same co-location server need to form a secure association between them. By "secure association", we mean that no eavesdropper may decrypt or falsify messages between the communicating partners. We also address the issue of authenticity, which requires that both devices should be able to demonstrate (confirm) the co-location property of each other by the human participants identifying the physical devices involved.

We assume the presence of adversary trying to attack from the same physical space, the next room, the next floor of the building, or possibly from a remote location. Further, it has surveyed the location where the two legitimate devices are attempting to pair and also knows the co-location capability information of the communicating partners. The adversary can use this knowledge to convince one or both of the legitimate devices that it is co-located with them. Since, the problem is

demonstrating that two legitimate devices are physically in the same place, verifying that a communicating partner is not an imposter is very important. We consider an impostor attack where the adversary succeeds in pairing with one of the legitimate device by proving falsely that it is physically co-located with it.

Another threat is when a fake co-location server is introduced. This highlight the risk of two possible attacks: denial-of-server (DoS) attack and potential for impersonation attack. We are not considering DoS attack that is result of frequency jamming, since this would affect any communications system. In our proposed solution co-location server only recommends/suggests the common possible method(s) of authentication, but cannot impose any particular scheme. Also, it is not providing any code or information regarding keys to the co-location server, so controlling this device does not provide any privileged information. One possible attack is that a malicious co-location server would only suggest pairing with compromised devices or using weak protocols. Compromised devices are a risk in any system; exclusion of obvious physical devices would cause the server to be questioned; once some basic association has been formed devices may improve the strength of their pairing through maintenance of the connection, which does not require the co-location server. Another possible consideration to mitigate this risk is that each device before registration authenticates the co-location server to check that it is the actual server with which they want to register.

4 Development Status

We have implemented a proof of concept version of the proposed system, which has given us positive results. During these tests, we used PhidgetInterfaceKits along with several sensors and three laptops. Since, the work is still in progress, so more detailed implementation of the system and results has been left for future work.

We want to further clarify that in our proposed scheme, the co-location server only provides bootstrapping information to two unknown devices in an ambient environment, so that pairing process can be commenced. It is the responsibility of device component to execute the authentication scheme to prove the physical co-location property of devices. Moreover, we are not considering the traditional centralized server-based approach. Our proposed system can be implemented with or without directory service. When deployed without a co-location server, peer devices (i.e. client and resource) can locate each other directly using local broadcast or multicast techniques.

Currently, we are investigating a number of authentication strategies to aid the design of our system. Further, we need to consider a number of issues along the way, such as looking into efficient credential revocation mechanisms and device-chaining (i.e. when two devices are in the same proximity but are unable to perform direct authentication because of long distance, then there is the need of another device sharing the proximity with both of the devices to mediate the authentication between them). We are also interested in descriptions of authentication quality (strength of keys, ease of mimicking pairing action, visibility of pairing actions) and their use in selecting mutually acceptable authentication scheme. To aid in the process of

determining if the proposed system is successful, we shall use several scenarios that highlight a number of aspects of secure device pairing. We shall also conduct a usability and more detailed security analysis. Results obtained from these analyses will be compared with other existing systems offering pairing mechanism.

5 Conclusion

Pervasive computing has given the vision of *'anytime anywhere'* computing systems, which differ from more traditional computing systems due to the ad-hoc, spontaneous nature of interactions among devices. These systems are prone to security risks, such as eavesdropping but require different techniques to traditional access control to manage. Physical proximity is however a good basis for establishing associations. Many devices will carry sensors for other purposes, which could be used in order to demonstrate this proximity. Recently, secure device pairing has gained significant attention from researchers and a significant set of techniques and protocols have been proposed. Some of these techniques consider devices equipped with infrared or laser transceivers, other require embedded accelerometers, cameras, speakers, microphones and displays. The issue of a universal pairing mechanism is still unresolved. To this end, we attempt to fill the gap left by prior work and propose a general device pairing scheme for pervasive environments. The benefit of this approach from the user's point of view is to eliminate confusion as to what process to follow while pairing devices, and from application and technological point of view is its capability to securely pair the devices under a number of different contexts (in terms of device capabilities).

Acknowledgments. Thanks to Jon Robinson of Software Systems Group for his feedback on this work.

References

1. Goodrich, M.T., et al., Loud and Clear: Human-Verifiable Authentication Based on Audio. in 26th IEEE Intl. Conf. on Distributed Computing Systems, ICDCS 2006.
2. Balfanz, D., et al., Talking to Strangers: Authentication in Ad-hoc Wireless Networks. in Symposium on Network and Distributed Systems Security (NDSS '02). 2002. San Diego, California.
3. Varshavsky, A., et al., Amigo: Proximity-Based Authentication of Mobile Devices. in UbiComp 2007: Ubiquitous Computing. 2007. p. 253-270.
4. Spahic, A., et al., Pre-Authentication using Infrared. in Privacy, Security, and Trust Within the Context of Pervasive Computing, 2005. p. 105-112
5. McCune, J.M., et al., Seeing-is-Believing: Using Camera Phones for Human-Verifiable Authentication. in IEEE Symposium on Security and Privacy, 2005. p. 110 - 124.
6. Mayrhofer, R. and M. Welch. A Human-Verifiable Authentication Protocol Using Visible Laser Light. in 2nd Intl. Conf. on Availability, Reliability and Security (ARES'07) 2007.
7. Saxena, N., et al., Secure Device Pairing based on a Visual Channel. IEEE Symposium on Security and Privacy 2006. Oaklan, CA. p. 306-313.

8. Mayrhofer, R. and H. Gellersen, Shake Well Before Use: Authentication Based on Accelerometer Data. in 5th International Conference on Pervasive Computing (Pervasive-07). 2007.
9. Stajano, F. and R. Anderson, The Resurrecting Duckling: security issues for ubiquitous computing. Computer, 2002. 35(4): p. 22-26.
10. Stajano, F., The Resurrecting Duckling - What Next?, in Revised Papers from the 8th International Workshop on Security Protocols. 2001, Springer-Verlag.
11. Mayrhofer, R., et al., An Authentication Protocol Using Ultrasonic Ranging. Technical Report. 2006, Lancaster University.
12. Mayrhofer, R. and H. Gellersen. On the Security of Ultrasound as Out-of-band Channel. in IEEE International Symposium on Parallel and Distributed Processing (IPDPS-07), 2007.
13. Shaked, Y. and A. Wool. Cracking the Bluetooth PIN. in 3rd ACM Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys '05), 2005. Seattle, Washington.
14. Jakobsson, M. and S. Wetzel, Security Weaknesses in Bluetooth. Lecture Notes in Computer Science, 2001. 2020: p. 176+.
15. Holmquist, L.E., et al., Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts, in 3rd international conference on Ubiquitous Computing. 2001, Springer-Verlag: Atlanta, Georgia, USA.
16. Zhu, F., et al., Classification of Service Discovery in Pervasive Computing Environments. MSU-CSE-02-24, Michigan State University, East Lansing, 2002.
17. Bettstetter, C. and C. Renner. A Comparison of Service Discovery Protocols and Implementation of the Service Location Protocol. in Proceedings of EUNICE 2000, Sixth EUNICE Open European Summer School. 2000. Twente, Netherlands.
18. Ververidis, C.N. and G.C. Polyzos, Service discovery for mobile Ad Hoc networks: a survey of issues and techniques. in IEEE Communications Surveys & Tutorials, 2008. 10(3): p. 30-45.
19. Borisov, N., et al., Intercepting mobile communications: the insecurity of 802.11. in 7th ACM Annual International Conference on Mobile Computing and Networking (MobiCom '01), 2001. Rome, Italy: ACM.
20. Hager, C.T. and S.F. Midkiff, An analysis of Bluetooth security vulnerabilities. in IEEE Wireless Communications and Networking (WCNC 03), 2003. 3: p. 1825-1831.

Secure communications between multi-capacity devices with authentication support by network operators

Jean-Philippe Wary[1] and Maryline Laurent-Maknavicius[2],

[1] SFR, 1 Place Carpeaux,
92915 Paris La Défense, France
[2] CNRS Samovar UMR 5157, TELECOM&Management SudParis,
9 rue Charles Fourier, 91011 Evry, France
{jean-philippe.wary@sfr.com, Maryline.Maknavicius@it-sudparis.eu}

Abstract. This paper proposes to benefit from each original network authentication procedure provided by operators to allow mutual authentication between two multi-capacity devices and guarantee the same security level to both of them. Operators can agree providing jointly this authentication service so multi-operator crossed authentication infrastructures can take place for instance over internet. As such, users needing strongly secure interconnectivity (e.g. SIP usage or over ad hoc infrastructures) can access to this service through Internet with no huge extra costs contrary to PKI or Kerberos solutions. Additionally to this attractive marketing offer, authentication could become a new growth for operators.

Keywords: Key Agreement, Mutual Authentication, Device Pairing, Secure Pairing, Multi-Capacity Devices.

1 Introduction

In several situations, users through their terminals need to mutually authenticate and secure data traffic across a wireless communication channel. IP telephony is one of these applications where users would like to establish a voice call at low cost over Internet but with security and privacy guarantees. They want to be sure that their correspondent is as claimed, and their communications will not be eavesdropped. File sharing over Internet, or over an ad hoc network is another example where the identity of the entities must be guaranteed and the data exchanged need confidentiality and integrity protection. All these applications require opportunistic communications to be initiated with high-level security.

The arrival of multi-capacity devices on the market brings diversity in terms of technological means (3G, Bluetooth, ad hoc, Internet…), and the nature of the interconnection which might be direct between two devices or performed across a network. The intermediary network, if any, can be infrastructureless (e.g. ad hoc network) or under the supervision of an operator (e.g. 3G). Due to the wide variety of their features, the network access technologies have very different security levels, ranging from a weak level (Bluetooth) up to a strong level (3G). The usage of multi-

capacity devices thus brings more flexibility to users as crossed technology combinations may help solving the security session establishment between devices.

A number of security solutions were published in the last few years in order to help any pair of devices getting into contact for the first time to interconnect securely. A first pairing approach relates to devices close to each other (i.e. in the same radio coverage) that need auxiliary channel(s) for transmitting an authenticated secret for next securing their direct exchanges. Other approaches are under the assumption of an existing trusted third party like Kerberos, Wireless PKI (WPKI) or an AAA service. Most of them are mono technology solutions, i.e. having the same network interface enabled to perform both security establishment and traffic exchange.

This paper proposes a new approach that benefits from the original network authentication procedure performed by the operators (e.g. a cellular network operator, an Internet Service Provider…). Any subscribers having Internet connectivity are able to mutually authenticate, and secure their communications, whatever the underlying interconnection technology in use. This approach has several advantages. Deployment of it is easy and no huge extra cost is needed as the security material is already available in the terminals. Multi-operator crossed authentication is made possible. Both users benefit from the high-level security offered by the operators. Finally, for operators, the authentication service itself can be a new source of income.

The organization of the paper is as follows. First, section 2 presents related works and highlights the need for designing a new C2C (Customer-to-Customer) oriented approach. Section 3 describes the network architecture and clarifies the prerequisites of our approach. Section 4 gives the conceptual description of the approach and section 5 concludes.

2 Related works

In the past five years, a number of research and standardization works were conducted on how to initiate a secure session between any two users (i.e. their devices) getting into contact for the first time. They do not know each other and they do not share any common context (e.g. pre-shared key). Beyond the authentication problem, the session key establishment problem raises. According to the underlying network technology, several approaches were investigated.

2.1 Secure pairing approaches

The pairing approaches relate to devices that are in the same short-range radio coverage, and can directly interconnect. Bluetooth and Wi-Fi are two examples of our everyday life needs in direct and close interconnectivity. In the literature [1], the security approaches rely on some auxiliary physical channel(s) that can be authenticated by the users, and serve to communicate some secrets. The originality of the approaches lies in the nature of the channel that is classically visual, audio,

touch… and based on the devices' available features such as LEDs, beeping, vibration, or any synchronized combination of them. Usually the users are asked to proceed to the validation of the channel by comparing character strings, the good synchronization of light/sound/vibration signals on both devices… In some cases, users support the synchronization itself by putting closely together the devices in a certain position or shaking the devices together [2]. The relevance of the secure pairing approaches can be measured by their user-friendliness, the rate of false negatives, and their rapidity of execution.

2.2 Trusted third party approaches

Other approaches are under the assumption of an existing and online trusted third party like Kerberos, Wireless PKI (WPKI) or AAA service, but these approaches are B2C oriented only. They permit a customer to authenticate to a service or network provider, but they do not solve the C2C connectivity security problem. Even if Kerberos [3] could be pretty easily adapted to C2C communications, it is very heavy in terms of number of exchanges and CPU processing. Kerberos requires that users are previously known to one of the Kerberos servers and it does not fit to the inter-domain authentication.

The AAA service [4] is an internal service used by operators to authenticate their subscribers with a high level security before affording them access to their networks. With the Diameter protocol [5], inter-domain authentication between operators is possible, but as it is standardized and used today, this AAA authentication service can't be accessed by any other external entity. Some research efforts are in progress in that direction to help users establishing secure communications over ad hoc networks thanks to some delegated AAA ad hoc nodes [6] or a distributed AAA service [7].

Finally, WPKI [8] is adapted to mobile users that need to authenticate to a service provider (mainly e-governmental and banking services) or to sign a document. The online WPKI service of the cellular network operator is acting as a proxy between the users and the service providers, handling the authentication of the users based on private/public key. The WPKI provides a unidirectional crossed-technology authentication, with the user asking for a service access from a terminal (e.g. PC) and performing unidirectional authentication from his cell phone. The resulting authentication level is the one provided by the operator and SIM card usage.

2.3 Approach [9]

The approach [9] is also worth presenting as it considers a crossed technology authentication based on mobile phone authentication. From a PC, a user can authenticate to any Internet application server with the help of an online identity provider belonging to the cellular network operator. The identity provider helps the user to download java applets on the PC, so the PC can locally access to some SIM USB dongles, or locally communicate to his cell phone through Bluetooth. This authentication is unidirectional, B2C oriented, and does only support mobile phone authentication.

2.4 Strong need for designing a new authentication approach

None of today's authentication approaches support all the following features:

- mutual authentication with the same level of authentication for both parties;
- crossed-technology authentication, the multi-capacity device can operate authentication on one of its enabled interface and handle data traffic on another interface;
- inter-domain and multi-technology authentication, so any subscribers of operator A using access network technology T#1 can authenticate to any subscribers of operator B with technology T#2;
- C2C, C2B and B2C authentication, any entities having the capability to authenticate to any operators can be authenticated by any other entities.

To simplify and strengthen security in C2C, C2B or B2C communications, and prepare a secure and open environment for next coming applications, there is a strong need to develop a new authentication framework and protocols. The next section describes the objectives of our proposed authentication approach and the observed constraints.

3 Architecture, prerequisites and constraints

The objective of our approach is to provide a bidirectional and flexible authentication service offering a possibly large choice of authentication methods, with no high extra cost for the operators and users, with a symmetrical approach for the authentication handling.

The assumptions of our solutions are as follows (see figure 1 for notations):

- The Entity-A is a subscriber of Operator-A, and Entity-B to Operator-B. The Operator-A is used to authenticate Entity-A on network access technology T#1, and the Operator-B is used to authenticate Entity-B on access technology T#2. The Entity-A is uniquely identified by the Operator-A with the following NAI (Network Access Identifier): Entity-A@Operator-A. The Entity-B is uniquely identified by Entity-B@Operator-B.
- The Entities A and B are equipped with multi-capacity devices, and at least one of the interfaces of the device is common (technology T#3) for the entities to exchange their data traffic. The device of Entity-A has the following available technologies T#1, T#3 and T#5, and the device of Entity-B is provided with interfaces of technologies T#2, T#3 and T#4.
- The Operators A and B have previously signed an agreement to offer a crossed authentication service to their subscribers and/or to provide mutually requested authentication vectors to their Authentication Gateway (for instance AG(B) is able to request an authentication vector for a specific customer of Operator-A)[2].

[2] This type of agreement is already in use today between 2G and 3G Mobile Network Operators in order to provide international roaming to their mutual customers. In this case, the legacy

- The Entity-A is originally authenticated over the technology T#1 by the Operator-A, and there is another type of authentication over the technology #5 realized by the Authentication Gateway AG(A). For this service, AG(A) uses the authentication vector (AV) computed by Operator-A[3] using the technology T#1.
- In the same way, the Entity-B is originally authenticated over the technology T#2 by the Operator-B, and there is another type of authentication over the technology #4 realized by the Authentication Gateway AG(B). AG(B) uses likely some authentication vectors (AV) available in the Operator-B's infrastructure (T#2).

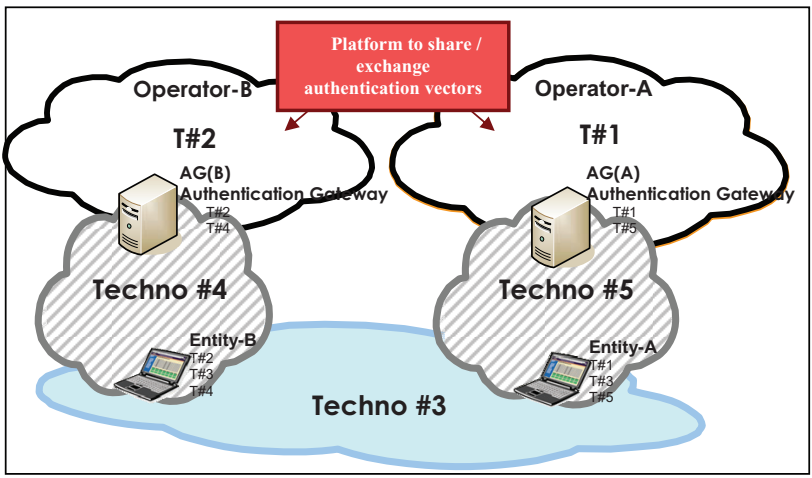


Figure 1: Architecture of our authentication approach

The EAP authentication methods can provide a shared secret that might serve to bootstrap a security protocol between entities A and B.

4 Description of the concept

This section is organized in four parts:
- A simple way to extend EAP-AKA usages over Internet,
- A generalization of the mutual authentication concept,

network and technology used to exchange these authentication vectors is the SS7 network and protocol.

[3] For instance, the Operator-A may be a Mobile Network Operator using the 3G technology (T#1), in this case the authentication vector to be supplied to AG(A) to authenticate the Entity-A over internet (technology T#5) is naturally based on EAP-AKA protocol definition and the Entity-A is authenticated by AG(A) with the EAP-AKA protocol.

- Identification of open issues, in particular regarding the potential gradient of trust regarding authentication methods available to each entity.

4.1 A simple way to extend EAP-AKA usages over Internet

We illustrate in a simple way the concept through the usage of EAP-AKA (Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement) [10] which is the authentication method to be implemented between AG(A) and the Entity-A.

In this example, the Entity-A is authenticated by AG(A), which has requested the necessary authentication vector to the native MNO operator (on which the Entity-A is regularly registered, it means that Entity-A has some SIM card issued by Operator-A). In case of success of the EAP-AKA authentication phase, the two parties AG(A) and Entity-A then share the same set of secret keys: a 128 bits session key for integrity check (IK_A) and a 128 bits session key for encryption (CK_A). The two parties are then able to build a secure channel with (IK_A) and (CK_A), and only those two parties are able to know the values of the two keys (IK_A) and (CK_A). It means that every packet can be ciphered with these keys and the other party, is the only one (already authenticated) able to use and know the secret key to decipher the packets.

This technology is currently deployed by MNO operators to offer Wi-Fi access to their customers without any extra access control scheme.

In our case, it is now possible to have a simplified view of the general case, if we consider that:

- The two Entities A and B are owned by the same Operator-A (for instance a 3G MNO),
- The Entity-A is already sharing a secure channel with AG(A) over Internet (it means that AG(A) already authenticated Entity-A through the EAP-AKA protocol),
- The Entity-B is not connected on Internet, but is connected to an ad hoc network (technology T#3) on which Entity-A is already connected.

As discussed in the state of the art, if Entity-A shares its Internet access with the ad hoc network, then Entity-B may establish a secure channel with AG(A) through the EAP-AKA protocol (all the communications will be routed by Entity-A).

Our proposal is that Entity-A plays the role of AG(A) regarding Entity-B to allow Entity-A to authenticate Entity-B by requesting AG(A) (with which Entity-A already shares a secure channel based on a first EAP-AKA challenge), the necessary authentication vectors (for EAP-AKA protocol) computed in the infrastructure of the Operator-A. At the end of this second EAP-AKA challenge, the Entity-A will have authenticated the Entity-B with the level of trust provided originally by the 3G Authentication protocol and it will share with Entity-B the keys ($IK_{A=>B}$) and ($CK_{A=>B}$).

It means that any 3G customer is able to authenticate "strongly" any other 3G customers over ad hoc technology as long as it is able to communicate with its Authentication Gateway AG(A) (through SMS or Internet access for instance). The authentication is qualified as strong since the challenged customer is using its SIM card to answer to the EAP-AKA Challenge, and today 3G Mobile Network Authentication are not repudiable or broken.

It has to be noted that over the ad hoc network, Entity-B is able to authenticate Entity-A by replaying the same protocol with AG(A) (AG(A)= AG(B)):

- The authentication of Entity-B to AG(A) routed by Entity-A over Internet,
- Entity-B requests an Entity-A's authentication vector (EAP-AKA protocol) to AG(A),
- The authentication of Entity-A by Entity-B over the ad hoc technology through the received EAP-AKA protocol based on the AG(A) received authentication vectors.

At this moment, Entity-A and Entity-B share four 128 bit keys:

- From the first step when Entity-A authenticates Entity-B: ($IK_{A=>B}$) and ($CK_{A=>B}$),
- From the second step when Entity-B authenticates Entity-A: ($IK_{B=>A}$) and ($CK_{B=>A}$).

4.2 A first level of generalization of the mutual authentication concept

We illustrate the generalization of the concept with the following hypothesis:

- H1: Entity-A is already authenticated (EAP-AKA protocol) by the Authentication Gateway AG(A) through the technology T#4 and a secure channel over (IK_A) and (CK_A) is already established between Entity-A and AG(A).
- H2: Entity-B is already authenticated (by a proprietary "weak" algorithm based on a password hashed with a random challenge) through technology T#5 by Authentication Gateway AG(B) and a secure channel over a session key (KS_B, computed by the derivation of the password with a random value) is already established between them.
- H3: Entities A and B are able to communicate over a dedicated technology #3, which normally does not provide security features.
- H4: The two Operators A and B are able to exchange authentication vectors through a dedicated mean. Operator-A supplies to AG(B) some EAP-AKA authentication vectors, and Operator-B supplies to AG(A) some proprietary authentication vectors (which may be composed of: Random-Value, RES_B: Result of a first hashing function applied to the customer password and the Random value, a session key: KS_B the result of a second hashing function applied to the customer password and the Random value).
- H5: The Entity-B wants to establish a secure session with the Entity-A over the technology T#3.

The following steps apply:

- Entity-B invites the Entity-A to establish a session and supplies its identity Entity-B@Operator-B to Entity-A,
- Entity-A requests directly AG(A) for Entity-B@Operator-B authentication vectors,
- AG(A) requests Operator-B for specific authentication vectors AV_B for the customer Entity-B@Operator-B,
- AG(A) sends back to Entity-A the necessary information and the way to proceed to the Entity-B's authentication,
- Entity-A authenticates Entity-B and in case of success, it provides to Entity-B its Identity: Entity-A@Operator-A (in other cases, Entity-A may close the session). At this step, Entity-A and Entity-B share the values: Random-Value, RES_B, KS_B . KS_B is a secret value which is not exchanged over the technology T#3,
- Entity-B requests AG(B) for Entity-A@Operator-A's authentication vectors AV_A,
- AG(B) requests Operator-A for specific authentication vectors for Entity-A@Operator-A's customer.
- AG(B) sends back to Entity-B the EAP-AKA authentication vector ($AV_{B=>A}$),
- Entity-B authenticates Entity-A. If successful, Entity-B and Entity-A share the secret values: ($IK_{B=>A}$) and ($CK_{B=>A}$). Otherwise, Entity-B may close the session.
- At the end, Entity-A and Entity-B have proceeded to a mutual authentication and are able to build a secure channel between them based on this mutual authentication. The secure channel may be based on a session key $SSK_{A/B}$ computed by each party with the following shared secret values: KS_B, $IK_{B=>A}$ and $CK_{B=>A}$.
- The use of a shared secret key $SSK_{A/B}$ is equivalent to an implicit mutual authentication, because only the other already authenticated party may be able to use and know the secret key $SSK_{A/B.}$

To generalize the concept, we have no hypothesis on the available authentication methods for each technology, we only consider that each of these methods allows the operators to compute and supply authentication vector (AV) that may contain the necessary information to proceed to a one-way authentication and in case of success, it establishes a session key SSK.

4.3 A second level of generalization of the mutual authentication concept

The assumptions of the section 3 apply, and have to be completed with the following hypothesis:

- H1: Entity-A is already authenticated by Authentication Gateway AG(A) and a secure channel over (SSK_A) is already established between Entity-A and AG(A).
- H2: Entity-B is already authenticated by Authentication Gateway AG(B) and a secure channel over (SSK_B) is already established between Entity-B and AG(B).
- H3: Operator-A is able to provide AV_A to GA(B) on request, and AV_A includes a pre-computed session key: ($SSK_{B=>A}$).
- H4: Operator-B is able to provide AV_B to GA(A) on request, and AV_B includes a pre-computed session key: ($SSK_{A=>B}$).
- H5: Entities A and B are able to communicate over a dedicated technology #3.

The following way to build a mutual authentication between the parties A and B:

- Entity-B invites the Entity-A to establish a session and supplies its identity Entity-B@Operator-B to Entity-A,
- Entity-A requests directly AG(A) for Entity-B@OperatorB's authentication vectors,
- AG(A) requests Operator-B for specific authentication vectors ($AV_{A=>B}$) for Entity-B@Operator-B customer,
- AG(A) sends back to Entity-A the necessary information ($AV_{A=>B}$) and the way to proceed to the authentication of Entity-B,
- Entity-A authenticates Entity-B and if successful, it provides to Entity-B its identity: Entity-A@Operator-A. Otherwise, Entity-A may close the session. At this step, Entity-A and Entity-B share a secret value: ($SSK_{A=>B}$) which was not exchanged over technology T#3,
- Entity-B requests AG(B) for Entity-A@Operator-A authentication vectors,
- AG(B) requests Operator-A for specific authentication vectors ($AV_{B=>A}$) for Entity-A@Operator-A customer,
- AG(B) sends back to Entity-B the authentication vector ($AV_{B=>A}$),
- Entity-B authenticates Entity-A. if successful, Entity-B may close the session. Entity-B and Entity-A share two secret values: ($SSK_{A=>B}$) and ($SSK_{B=>A}$) which were not exchanged over technology T#3,
- At this stage, Entity-A and Entity-B proceeded to a mutual authentication and are able to build a secure channel between them based on this mutual authentication. The secure channel may be based on a session key ($SSK_{A/B}$) computed by each party with the following shared secret values ($SSK_{A=>B}$) and ($SSK_{B=>A}$),
- The use of a shared secret key $SSK_{A/B}$ is implicitly equivalent to a mutual authentication, as the other party (already authenticated) is the only one able to use and know the secret key ($SSK_{A/B}$).

Mutual authentication between two parties is thus achieved over the technology (T#3) by using the existing infrastructures and the native security services provided by their native Operators.

4.5 Open issues

Several open issues are identified:

- How to manage the fact that the operators A and B may provide different levels of security and trust in their native authentication scheme for their customers? Is it possible to clearly evaluate/measure the level of the provided mutual authentication if operators offer dissymmetric levels of security for their own customers' authentication?
- Is it possible to define a way to manage at each entity's level an Information Security Policy to protect internal assets in case of dissymmetric level of authentication during the mutual authentication phase?
- This scheme is a new way of growth for operators by charging delivery of authentication vectors. A new billing scheme similar to the transportation of voice might emerge with payment by the entity requesting the authentication vector or the entity doing the checking. On which bases can the operators build a new revenue scheme?
- Is it possible to easily extend this mutual authentication between two entities to some group authentication ? This will be helpful to secure the access to some multicast applications and their multicast data.

5 Conclusions

This study allows any customers to authenticate mutually over any technologies as long as they still be able to communicate with a trusted entity, i.e. their native home operator. The interesting point is that the proposed concept doesn't need any costly investment as it completely reuses existing technologies and platforms (EAP-AKA, HSS and GBA for Mobile network operator, EAP-TLS and each existing EAP method).

As we demonstrated, if one of the parties is only protected by a login/password technology [11], the secure channel established with another party using a SIM card improves the security of the channel and the mutual authentication between the parties. Improvement is high in comparison to the weak security level offered by the use of password technology.

We are convinced that there is a huge interest today regarding the 3 billion SIM cards used over the world to secure mobile network communications, in particular if they can be reused by customers to communicate over unsecure networks. The use of these authentications, as described in this paper, might strongly help to support secure mutual authentication in a number of communications scenarios over the world.

A patent application has been filed in August 2008 (under the number FR 0855595).

6 References

1. Saxena, N., Voris, J.: Pairing Devices with Good Quality Output Interfaces. **In:** Distributed Computing Systems Workshops, 2008. ICDCS '08. 28th International Conference on, pp. 382--387 (2008)
2. Castelluccia, C., Mutaf, P.: Shake them up!: a movement-based pairing protocol for CPU-constrained devices. In: Shin, K.G., Kotz, D., Noble, B.D. (eds.), Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services. MobiSys 2005. pp. 51--64. Seattle, Washington, USA (2005)
3. Neuman, C., Yu, T., Hartman S., Raeburn, K: The Kerberos Network Authentication Service (V5), IETF Request for Comment 4120 (2005)
4. C. de Laat, Gross G., Gommans L., Vollbrecht J., Spence D.: Generic AAA Architecture, IETF Request for Comments RFC2903 (2000)
5. Calhoun, P., Loughney, J., Guttman, E., Zorn, G., Arkko, J.: Diameter Base Protocol. IETF Request for Comments RFC3588 (2003)
6 Chaouchi, H., Laurent-Maknavicius, M.: Toward a New ad hoc node design for secure service deployment over ad hoc network. In: Workshop on Mobile and Wireless Networks Security. MWNS 2008. pp. 1--11, Singapore (2008)
7 Larafa, S., Laurent-Maknavicius, M., Chaouchi, H.: Light and Distributed AAA Scheme for Mobile Ad hoc Network. In: First Workshop on Security of Autonomous and Spontaneous Networks. SETOP 2008. pp. 93--104. Loctudy, France (2008)
8. WAP Forum: Wireless Application Protocol, http://www.wapforum.org
9. Van Do, T., Jonvik T., Feng B., Van Thuan D., Jorstad I.: Simple strong authentication for Internet applications using mobile phones, In: IEEE GLOBECOM 2008. New Orleans, USA (2008)
10. Arkko, J., Haverinen, H.: Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA). IETF Request for Comments RFC4187 (2006)
11. Blunk, L., Vollbrecht, J., PPP Extensible Authentication Protocol (EAP), . IETF Request for Comments RFC2284 (1998)

On AAA framework in opportunistic ad hoc network: OLSR usecase

Willy Jimenez, Hakima Chaouchi

CNRS SAMOVAR Lab/UMR 5157
TELECOM & Management SudParis, 9 rue Charles Fourier, 91011 EVRY, FRANCE
{Willy.Jimenez, Hakima.chaouchi}@it-sudparis.eu}

Abstract- In order to simply identify participating nodes in an ad hoc network, authentication is one good option to bring some trust in such dynamic and infrastructure-less network. Ad hoc network technology offers cheap and flexible network coverage extension. However it needs AAA support in order to allow a service provider to offer services over these ad hoc networks. The AAA should be distributed so that ad hoc nodes can benefit from such service even if there is no constant connectivity to the access network. [1]. This paper deals precisely with the AAA issue in an ad hoc network, it presents our architecture to handle AAA services in an ad hoc network with opportunistic connectivity to the infrastructure. First we introduce the general idea of distributing the AAA service on what we call "*special ad hoc nodes"* which will be preconfigured by a network operator, then a description of a possible design of this special ad hoc device is provided. We also propose AOLSR, an extension to OLSR protocol to provide routing services only to authenticated nodes.

Keywords: ad hoc networks, AAA, OLSR, authentication, trust, virtualized OS.

1 Introduction

In the context of Always On era, ad hoc technologies integration with the infrastructure is without any doubt a clever way to extend at low cost the network access coverage. However, a real and business oriented service deployment over ad hoc network requires firstly identifying the communicating nodes, securing the communications, and resource accounting. We believe that the integration of ad hoc and infrastructure-based technologies coupled with efficient security and accounting techniques is the answer for the urgent demand of network operators for appropriate architectures to host secure and large scale ubiquitous services.

This work is supported by Telecom Sud Paris, and ANR SARAH project 2006-2010

There are several security threats in ad hoc networks. First, those related to wireless data transmission such as eavesdropping, denial of service in message replaying, message distortion and active impersonation. Second, those related to ad hoc construction of the network. This means that attacks can come also from inside the mobile ad hoc network (MANET). Therefore we cannot trust one centralized node to support for instance the AAA service, because if this node would be compromised the whole network would be useless. Another problem is scalability. Ad hoc networks can have hundreds or even thousands of mobile nodes. This introduces important challenges to security mechanisms [2].

As most of the security issues in ad hoc networks are caused by trust less nodes. Authentication process is a strong solution to eliminate those misbehaving nodes or at least identify them. Nevertheless, ensuring authentication service in a self organized network is not easy to realize. We propose in this work to build a distributed AAA service in ad hoc network where the AAA service which is classically centralized in the infrastructure network is somehow distributed or designed hierarchically. These services will be securely distributed in the servicing ad hoc nodes. We also propose an authentication based OLSR routing named AOLSR to ensure the forwarding service only to authenticated nodes.

The remainder of the paper is organized as follows. First the distributed AAA architecture is presented in section 2, followed by the description of AOLSR protocol. Finally we present our approach analysis and we conclude this paper.

2 The proposed distributed AAA architecture

A number of research work was conducted on the classically centralized AAA functions, but very few studied the possible interactions between AAA and ad hoc networks since AAA service is classically centralized and ad hoc network is distributed and uncontrolled by its nature. Thus, the introduction of AAA into ad hoc environment is not an easy task due to the self organising aspect of the ad hoc network. However, by ensuring; as it is proposed in our paper, a secure delegation of the AAA service over some special ad hoc node, it is possible to consider extending the AAA service in the ad hoc network. This architecture is depicted on the Figure 1 below. It targets deploying several mechanisms such as authentication, authorization, accounting, key management, and other network services such as Neighbour and Service discovery mechanisms that are also necessary to provide information for the ad hoc node in order to allow him to get the appropriate service.

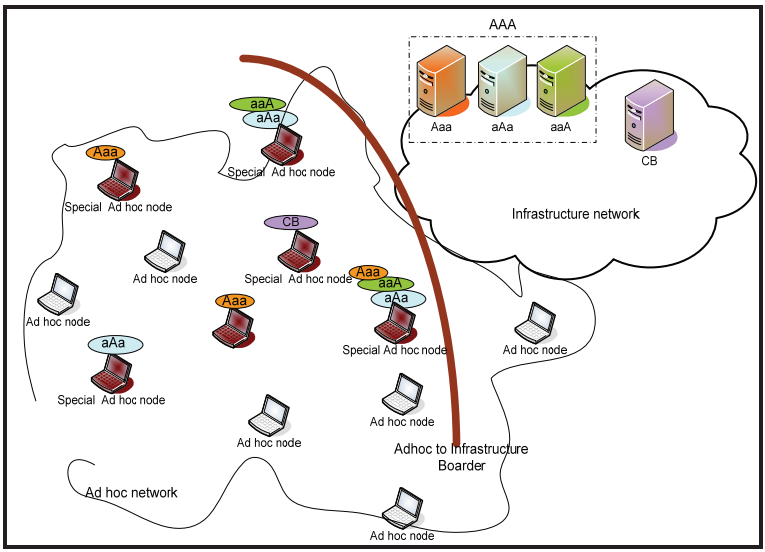


Figure 1: Delegated AAA service in a MANET [1]

To answer the question: How to distribute the AAA service over the ad hoc network? We propose two complementary design choices:

- the special ad hoc node design
- and the authentication based routing protocol design; the AOLSR.

2.1 Special Ad hoc node design

What we mean by special ad hoc node is a node capable to offer in a secured way the delegated network service from the operator in our case; the AAA service. The idea is to have either a dedicated ad hoc machine pre-configured by the operator to serve as a special ad hoc node for AAA service in the ad hoc network area; it might be a sort of robot-router moving in certain area. It can also be a machine belonging to trusted entity, or it can be a special running environment in a user's terminal. This special environment has to be isolated from the user's running environment. For this purpose, we decided to use virtualization techniques to create a special virtual machine in the ad hoc node that will run the delegated network services such as AAA service. We assume that in the future, virtualization will be less battery and resource consuming.

In this section we consider from a theoretical point of view, the use of the virtualization technologies in order to have a solution to deploy secure services in MANETs by using special ad hoc nodes. These nodes will run at least two environments, one for authentication controlled by the operator and one for the user. Virtualization with isolation concept will securely separate the two environments

embedded in the called trusted special ad hoc node. We studied different virtualization systems such as XEN, VMware and the security issues[3,4,5,6].

As stated earlier, the main idea is to have a distributed authentication service and bring it as close as possible to the users in a MANET. To do so, an operator should configure several nodes that will be part of the MANET; these nodes will have two different environments using virtualization software. The number of these special nodes will depend on the network size and the desired coverage service.

In the native environment of the node, the operator installs the AAA server; aka RADIUS server with a database of the customers that could use the services offered by the operator itself or a third party. The service offered to the customers is loaded in a virtual machine. Under these conditions, there should be a trust relation between the operator and the service provider if it is a third party, since sensitive data will be stored in the host operating system and the service provider will have physical access to the node.

Additionally it is important to consider the security issues regarding virtual machines implementation. However, the advantage of having virtual environment for the service offered is the possibility of migrating the service to another virtual machine in the case the service is down or is attacked by hackers reducing the down times and increasing service availability.

We designed our special ad hoc node by using open source software; implemented as follow:

- Prepare several laptops with Linux as host operating system.
- Install virtualization technology.
- Install "freeradius" server, and customers' database. For security database content should be encrypted.
- Configure RADIUS server and policies.
- Create virtual machines for the service provider according to the required operating system.
- Prepare the booting sequence of the laptops to leave them ready to work with all the services activated.

Next step should be signing a confidentiality agreement with the service provider and deliver the special ad hoc nodes to users who would like to provide the connectivity via the ad hoc service to other customers. The incoming customer will buy the service and use their accounts with the telecom operator to access it.

In order to have an idea about the authentication time, we built a scenario with three nodes, one of them with a virtualization application, as shown in the following picture:

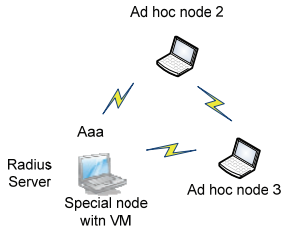


Figure 2: MANET with special ad hoc node for Measurements

All the nodes are using Linux as operating system; the virtual machine is Windows XP and was created using Xen and VirtualBox. The equipment used for the test is listed in the following Table 1:

Special node with VM	Ad hoc node 2	Ad doc node 3
HP xw4600 Core2 E6850 @ 3GHz 4GB RAM WPN111 Wireless USB Adapter 250GB HDD Fedora Core 8 + ndiswrapper + freeradius	Dell Latitude D410 Pentium M @ 2GHz 2GB RAM Intel PRO Wireless 2915 802.11 a/b/g Wi-Fi 40GB HDD Fedora Core 8 + radtest	Dell Inspiron 6400 Core2 T7200 @ 2GHz 2GB RAM Intel PRO Wireless 3945 802.11 a/b/g Wi-Fi 160GB HDD Fedora Core 8 + radtest

Table 1: Equipment list

The special node in figure 2 has freeradius server and the virtual environment; the nodes are using the radtest command to perform the authentication and the time is measured with Wireshark tool (traffic capture tool).

Two scenarios are considered per virtualization application, in the first one, the VM is ON but with no activity, in the second the VM is busy with a file transfer with the opposite node, it means, if we measure the time for node 2, the window VM is transferring a file with node 3 and vice versa. The transfer was done using Filezilla application.

VM type in Special Node	Node 2	Node 3
VirtualBox VM	2,881	1,150
VirtualBox VM busy	10,312	10,742
Xen VM	3,245	1,880
Xen VM busy	10,656	27,810

Table 2: Authentication Times expressed in mili seconds

From Table 2, we can observe how the authentication time increases when the VM is busy, which is reasonable since the kernel of the special node has more work to do, forward packets to/from Windows VM to one of the nodes; additionally the occupation of the air interface also increases. Then, it is important to evaluate all these factors in order to dimension the topology of the network and guarantee a target performance. Also it is important to study the power consumption when using virtual systems on an ad hoc node.

2.2 Authentication based ad hoc routing design- AOLSR

Based on the fact that special ad hoc nodes are available in the ad hoc network, we just need to ensure secure authentication to maintain access control in the ad hoc network. For this purpose, we propose to extend secure OLSR (sOLSR) [8] to include authentication.

sOLSR is a solution which adds a security mechanism to the OLSR protocol behavior [9]. sOLSR uses a cryptographic shared key to sign all the packets in order to ensure the integrity of OLSR control traffic data; only the nodes that share the key can participate in the routing domain; meaning that messages without verifiable signatures are discarded.

Additionally, to prevent replay attacks, secure OLSR uses timestamps, so the nodes exchange their timestamps before allowing the flow of any traffic between them. In this exchange process, three new messages types are introduced in OLSR:

- Challenge message
- Challenge-response message
- Response-response message

However, the exchange occurs between neighbors that have not registered timestamps of each other and where the traffic cannot be validated by the signature check. It means these messages are signed internally, and they carry their own digest/signature and they are never stacked with other OLSR-messages but rather sent in OLSR-packets of their own.

sOLSR description

The Timestamp Exchange Process

When A receives a signed message from a neighbor B, for which A has no registered time value, A initiates the timestamp exchange process. It means A sends a challenge message. The message is broadcasted since A might not have a route to B. The challenge message contains a 32-bit nonce. A then signs this message with a digest of the entire message and the shared key.

B has to respond to this message with a challenge-response message. B first generates the digest of its IP address, the received nonce and the shared key. B then generates a 32-bit nonce and transmits the nonce, the timestamp of B, the digest and a digest of the entire message and the shared key.

When A receives the challenge-response message from B, it first tries to validate the data. If the digest can be validated, then the timestamp of B is used to create the difference of time between A and B. A then generates a response-response message and broadcasts it to B. The message contains the timestamp of A, a digest of A's address, the nonce received from B and the shared key and a digest of the entire message and the key.

Finally, when B receives the response-response message from A, it tries to verify the digests. If they can be verified, B uses the received timestamp to register its time difference to A and the process is now completed.

Secure OLSR Implementation

The secure version of OLSR is implemented as a plugin of the olsrd daemon. It captures all the incoming traffic, verifies the packets and removes the signature message and updates the size field of the OLSR packet header. The plugin also intercepts all outgoing OLSR traffic to add the signature messages and updates the packets size as shown in the Figure 3 below [7]:

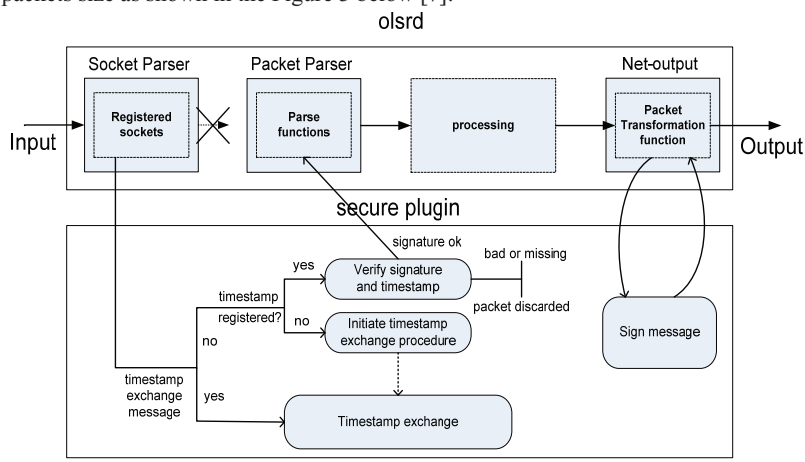


Figure 3: sOLSR design [7]

2.3 Proposed AOLSR description

Continuing with the purpose of having distributed authentication in MANETs now we propose the protocol part. In this case we want to authenticate all the nodes that will participate in the MANET routing domain, if authentication fails the node is not added and the ad hoc routing is blocked; it's the access control.

It means, there will be one special node inside the network which also runs a RADIUS server; the special ad hoc node, which will restrict the access to the routing domain through nodes authentication. The solution is implemented as a plugin of the OLSR daemon, and it is done modifying the existing secure plugin of olsrd as explained later.

Authenticated Optimized Link State Routing protocol or AOLSR is a new implementation of OLSR protocol where nodes are required to be authenticated by an AAA server as a previous requirement to participate in the routing domain. It is noted that if AOLSR is activated, OLSR and sOLSR packets are dropped by AOLSR nodes.

This new plugin has been developed by us based on secure OLSR plugin and then it supposes there is an existing key shared among all nodes. The idea of this new version is to add an authentication step as a mechanism to avoid that an attacker that compromised the key could join the routing domain. It means, in the case a node gets the shared key; it won't join the network unless it is authenticated by the radius server.

AOLSR uses the same process of secure OLSR, and it just adds some extra conditions in the behavior, as can be seen in the next Figure 4.

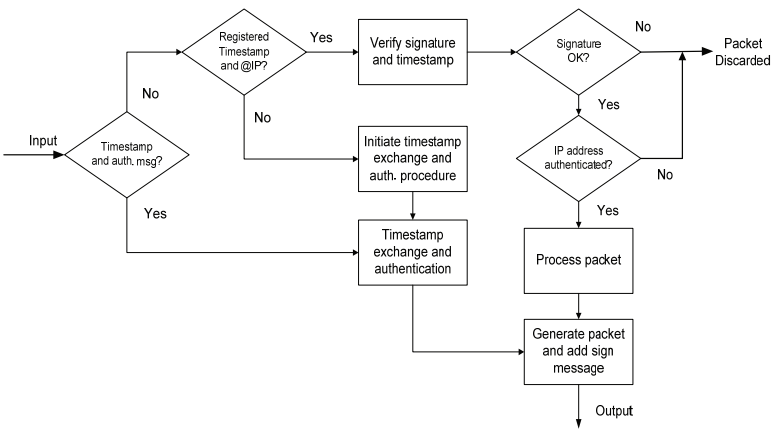


Figure 4: AOLSR - Flow Diagram

In our implementation we consider:

- Three nodes: A, B and C are considered in a two hop topology.
- A is a special node that has a radius server for authentication purposes.
- The IP address of the node is used as user name for authentication purposes.

- The radius server is implemented using freeradius server.
- The radius client request is done using the command “radtest”.

Now, we will describe how the AOLSR protocol functions considering two steps. At the beginning node A is available and node B is the first node coming to join the network. Once A and B are in the same MANET then C will join them and will be required also authentication.

First Node Authentication Scenario

The situation for this case is shown in the Figure 4. A is the main node of the network and is running AOLSR and the radius server.

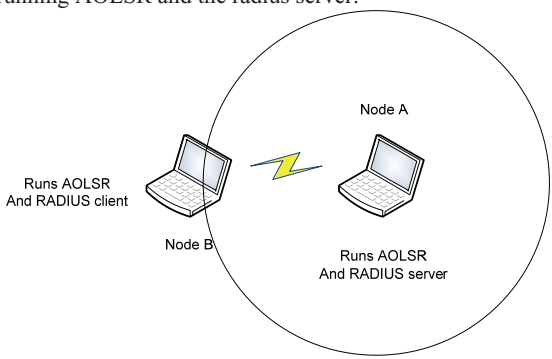


Figure 5: AOLSR - First node authentication scenario

Then B comes running AOLSR and wants to join A. When A receives any signed message from the new neighbor B, for which A has no registered time value, A initiates the timestamp exchange process. A sends the challenge message. The message is broadcasted since A might not have a route to B.

B has to respond to this message with a challenge-response message, but before that B authenticates with the radius server in A.

When A receives the challenge-response message from B, it first tries to validate the data. If the digests can be validated, then the timestamp of B is used to create the difference of time between A and B. Additionally, A checks the radius log file to verify the successful authentication of B. If it is good, A adds B to its authorized IP address list. Later A generates a response-response message and broadcasts it to B. The message contains the timestamp of A.

Finally, when B receives the response-response message from A, B uses the received timestamp to register its time difference to A and also B adds A in the trusted IP address list.

When A received another message from B, A already has B in the timestamp database, and then A will check also the authenticated IP address list, where B will be if the authentication process was successful. If not, then any packet coming from B is going to be rejected.

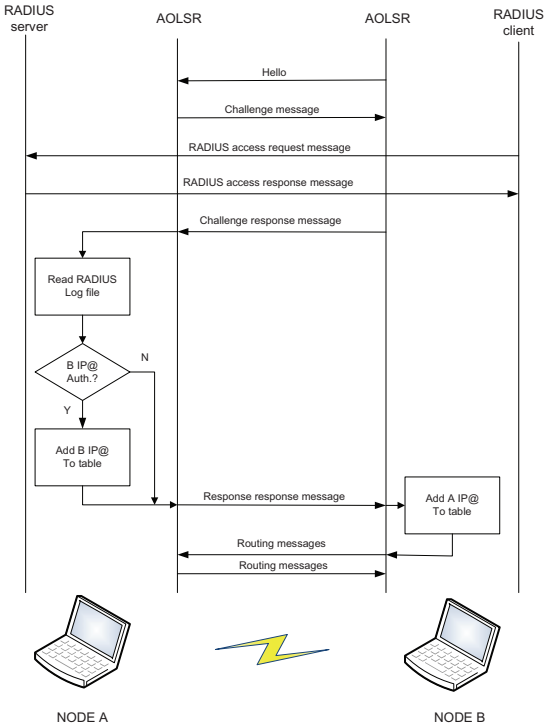


Figure 6: First node authentication - Logical Operation

Second Node Authentication Scenario

In this scenario a node C is the new node that wants to join the network formed by nodes A and B. The location of node C is next to B but cannot communicate directly to node A as shown in the next figure.

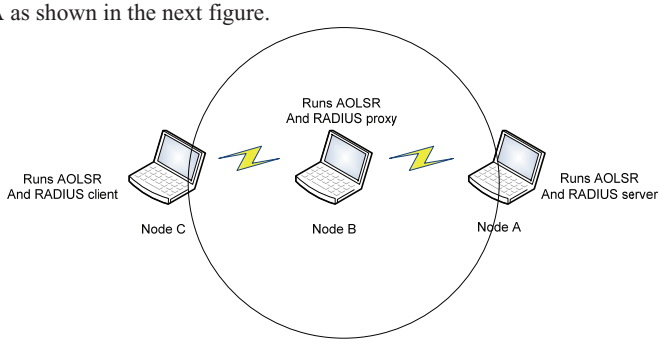


Figure 7: AOLSR - Second node authentication scenario

Since C cannot communicate with A directly, we assume that B is running a radius proxy also which will be used for C authentication.

When it arrives, node C sends a signed message to B, then a process similar to the one described above is initiated.

Node B has no registered time value for node C, thus B initiates the timestamp exchange process sending the challenge message.

When C receives the challenge message, it performs the authentication with the radius proxy running in node B, which forwards the request to the RADIUS server running in node A. After that C sends B a challenge-response message.

When B receives the challenge-response message from C, the timestamp of C is used to create the difference of time between them. Additionally, B checks the radius log file to verify the successful authentication of C. If it is good, B adds C to its authorized IP address list. After, node C generates a response-response message for B which contains the timestamp of C.

Finally, when C receives the response-response message from B, C uses the received timestamp to register its time difference with B and also registers B in the trusted IP address list.

When B received another message from C, B already has A in the timestamp database, and then B will check also the authenticated IP address list, where C will be if the authentication process was successful. If not, then any packet coming from C is going to be rejected.

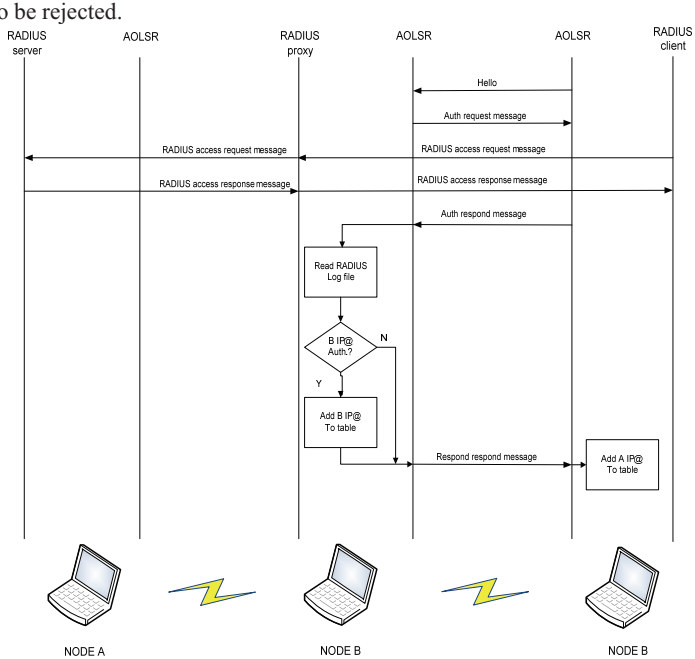


Figure 8: Second node authentication - Logical Operation

AOLSR Practical Implementation

For our practical implementation we add some new functions to the secure plugin code, especially to module olsrd_secure.c. We kept the main functions of the plugin because as explained before it intercepts all the incoming traffic and checks if they are timestamp exchange messages, if yes then the plugin processes the packet according to the packet type, if not it just removes the signature and sends it to the general packet parser function; and this behavior is helpful for our authentication process.

Then, new functions were added to create, add and consult a list of known IP address, and it is done after a successful login of the incoming new node. The authentication is verified reading the RADIUS log file and verifying the authentication of the new node's IP address. Additionally new lines were added to force the look up in the registered IP address list before routing.

The authentication process was embedded in the timestamp exchange process, it means when a new node arrives, it will initiate the timestamp exchange process and will be authenticated at the same time. Now, it was necessary to add a flag to differentiate the node with the RADIUS server and node with the proxy RADIUS in order to avoid that an incoming node tries to authenticate the server. For the RADIUS server and proxy we use freeradius server, and for the client the tool radtest.

3 Conclusions

We proposed two possible complementary solutions to implement the distributed authentication service in a MANET: using virtualization to build special ad hoc node that securely offers delegated network services such as distributed AAA service in MANET, and securing the routing protocol with our proposed AOLSR. We proposed to use virtualization to build special nodes in the network that provides the Authentication service on an ad hoc node in a secured manner by isolating the user environment from the AAA service embedded in the user's ad hoc node. While in the second solution; AOLSR, the routing protocol is modified to perform authentication as previous step of joining the MANET routing domain. This is to allow only authenticated nodes to benefit from the routing protocol.

From a performance point of view of the special ad hoc node, we got some authentication time measurements that show better response in the case of VirtualBox, however to conclude that VirtualBox is really better it is necessary to perform a more extensive test, we just wanted to have a reference for the authentication times. Before using virtualization it is important also to know the possible vulnerabilities of the package to use in order to implement basic security mechanism that can ensure the isolation and protection of the host environment and guest machines.

In the case of securing the routing protocol; AOLSR, we worked with an active routing protocol called OLSR. We used an open source daemon implementation (www.olsr.org). This daemon already has a secure plugin which uses a shared key to sign the packets and uses timestamps to avoid replay attacks and establish neighbor trust relationships. Regarding the key, it assumes that it is already available for the nodes that join the network. In our solution we modified the secure plugin of OLSR

daemon to add an authentication phase and we called it Authenticated OLSR (AOLSR), thus we assume also the availability of a shared key while the authentication process could protect the network from attacker that were able to get the shared key. It means, there is an initial node that acts as Authentication server and runs AOLSR which will enable the formation of a MANET, then any node that wants to join the network has to run AOLSR and has to be authenticated by the server, if succeeds it will be part of the network. For practical implementation we added new functions in the secure plugin to have authentication, in fact we keep the timestamp procedure and we added access control conditions for authentication and a new table to keep the known and authenticated IP address of the neighbors. We tested the implementation among three nodes and it worked, however a deep evaluation is required to test the performance of this solution. Of course, a larger ad hoc network is better to experiment our proposal.

4 References

[1] Chaouchi H. and Laurent-Maknavicius M. Annex 1 of French patent: Intégration de la technologie ad hoc dans la chaine de valeur des télécommunications, *INPI n° 0756559, 2007*.

[2] H Yang, H Y. Luo, F Ye, S W. Lu, and L Zhang, "Security in mobile ad hoc networks: Challenges and solutions" (2004). IEEE Wireless Communications. 11 (1), pp. 38-47. Postprint available free at: http://repositories.cdlib.org/postprints/618

[3] Antonopoulos A. Securing Virtualized Infrastructure: From Static Security to Virtual Shields. Senior Vice-President & Founding Partner, Nemertes Research. http://hackreport.net/wp-content/uploads/2007/03/nemertes-issue-paper-securing-virtualized-infrastructure.pdf

[4] Ormandy T. An Empirical Study into the Security Exposure to Host of Hostile Virtualized Environments. http://taviso.decsystem.org/virtsec.pdf

[5] Ferrie P. Attacks on Virtual Machine Emulators. Symantech Advanced Threat Research
http://www.symantec.com/avcenter/reference/Virtual_Machine_Threats.pdf

[6] King S., Chen P., Wan Y., Verbowski C., Wang H. and Lorch J. SubVirt: Implementing malware with virtual machines. In /Proceedings of the 2006 IEEE Symposium on Security and Privacy/ (May 21 - 24, 2006). IEEE Computer Society, Washington, DC, 314-327.

[7] Tonnesen A. Master Thesis: Implementing and extending the Optimized Link State Routing Protocol. University of Oslo. 2004. http://www.olsr.org/docs/report.pdf

[8] T. Clausen, “Securing OLSR problem statement”, Internet draft, 2005

[9] http://www.olsr.org/

Handling Security Vulnerabilities in Clustered Wireless Mesh Networks

Sadeq Ali Makram and Fahad Samad

Chair of Computer Science, Informatik 4, RWTH Aachen University
{makram,samad}@nets.rwth-aachen.de

Abstract. Secure channel assignment and communication is very important in Multi-channel Multi-Radio Wireless Mesh Networks (WMNs) and all wireless networks. The proposed schemes of dynamic channel assignment (DCA) for WMNs do not consider the security issues and expose to different attacks which influence the channel assignment. Therefore the security vulnerabilities in DCA should be addressed in order to achieve the benefits of Multi-channel Multi-Radio WMNs. In this paper, we discuss an efficient, secure channel assignment with DCA in wireless mesh network using clustering. Using public key cryptography, we have proposed an effective solution for securing DCA in a clustered WMNs.

keywords: Mesh networks, secure channel assignment, DCA

1 Introduction

Recently, wireless mesh networks (WMNs) have been in the focus of academia and industry research. This is because WMNs have several interesting characteristics such as self-organization, self-configuration, reliable services, and Internet connectivity. A WMN is a multihop wireless network which consists of *mesh routers* and *mesh clients*. Mesh routers have minimal mobility and form the backbone of a WMN which provides access to the mesh clients that can be stationary or mobile [1]. In WMNs, all nodes including clients have the ability to relay messages and behave like a router. This usually leads to efficient routing but also raises some issues because of its complex communication requirements. These also include security and privacy issues. The challenge is to have a safe transfer of messages from source to destination without any adverse interruption from relay nodes or other attackers. For efficient and secure transmission of data different topologies for these networks have been suggested. One such topology is clustering. A cluster is a group of neighbor nodes, acting in a group, having a node as cluster head, which is responsible to control that cluster. In that way, a WMN may consist of a number of clusters working together in an environment called clustered environment. To communicate with each other, the nodes need some channel frequencies to be assigned to them. Care should be taken in assigning these channels to maintain links' performance and also to reduce interference. There are many popular static and dynamic channel assignment

algorithms (DCA) used for this purpose (refer to the related work). The issue of secure channel assignment among nodes of the clusters in WMN still exists. The previously proposed schemes of DCA do not consider the security issues and expose to different attacks which influence the channel assignment. Therefore, the security vulnerabilities on DCA should be addressed in order to achieve the benefits of Multi-channel Multi-Radio WMN.

In this paper, we discuss an efficient, secure and dynamic channel assignment (DCA) in wireless mesh network using clustering. We use public key cryptography as an effective solution for securing DCA in a clustered WMNs. The trusted authority TA generates unique key pairs (private/public keys) for each node in the clustered network.

The distribution of the paper is in the following segments. section 2 summarizes related work. section 3, describes the network model and presents the security problem during channel allocation. Furthermore, this section also reviews and discusses the *Cluster Channel Assignment* (CCA) approach [2] of dynamic channel assignment for WMN using clustering.section 4 introduces the cluster-based security scheme for WMN based on public key cryptography. It also describes in detail the secure control messages for dynamic channel assignment followed by discussion about our scheme in section 5. Finally, section 6 concludes the paper.

2 Related Work

In this section, we will briefly describe the work that has been done for channel assignment problem and security issues in wireless mesh network (WMN).

2.1 Channel Assignment Approaches

Several approaches discuss channel assignment for WMNs. The main focus of these approaches is to enhance the overall network performance by reducing interference and maximizing the throughput. Raniwala et al. [3] have proposed a multi-channel and multi-hop WMN architecture with centralized channel assignment. Ramachandran et al. [4] have proposed a centralized channel assignment algorithm, that runs on a central server, which collects dynamically changing channel interference information periodically. Makram et al. [5] have proposed a distributed channel assignment based on clustering. Features of this approach include: fully distributed channel assignment, fair channel distribution for the clusters based on the number of the nodes within a cluster and re-assignment with consideration of the distance. None of these approaches consider the security aspect during the channel assignment planning. The threats like the effect of misbehaving malicious and compromised nodes, the threat of Denial of Service (DoS) attacks and the loss of confidentiality are not considered in all above mentioned algorithms [6].

2.2 Security Approaches

Lin et al. [7] have presented a collective authentication system for WMN. This system is based on threshold signature technique in which 't' out of 'n' servers are used for authentication. This system is specially useful in those areas where very large number of nodes work together under a single authority. Jin et al. [8] have proposed a group key agreement protocol for authentication of WMN. This is an improved version and more efficient then Tseng's group key agreement protocols. The protocol still lacks description about group member events like joining and leaving of nodes. Sun et al. [9] have explained an architecture providing a balance between anonymity and traceability while keeping the basic security requirements intact. The authors have suggested a solution with identity-based cryptography using bilinear pairings on elliptic curves. Xuygang et al. [10] have proposed a risk avoidance scheme using multi-path routing to assist encryption schemes mitigate the damages of security attacks. This is based on node identification. None of these approaches mentioned security for dynamic channel assignment. For this reason, we present our cluster-based security approach to secure channel assignment especially for dynamic assignment.

3 Problem Formulation and System Model

3.1 Network Model

A WMN can be represented as an undirected graph $G(V, E, K)$ known as the connectivity graph, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices in the graph that represents mesh routers, $F = \{f_1, f_2, \ldots, f_c\}$ the set of available channels, and $E = \{(v_i, u_j, f_i) | v_i, v_j \in V \wedge f_i \in F\}$ the set of wireless links between the mesh router v_i and its neighbors $v_j, \forall v_j \in N_i$ on channel f_i where N_i denotes the neighbors of node v_i. The wireless link $l_{i,j}^{f_i}$ is constructed between any two mesh routers v_i, u_j if they are located within each other's transmission range and agree on a common channel f_i. A mesh router v_i with multiple wireless network interface cards (WNICs) $R_{v_i} = \{r_1, r_2, \ldots, r_m\}$ may allocate different channels which should not exceed the number of R_{v_i} if available. F_{v_i} is the set of channels assigned to node v_i.

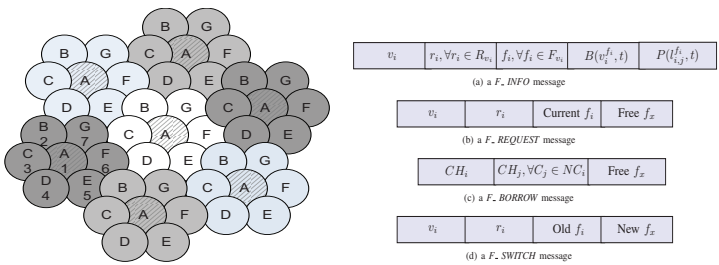
3.2 Problem Formulation

Most of the channel assignments algorithms do not consider security aspects during channel assignment and they assume that the mesh nodes are trusted nodes. Furthermore, the decision of the channel assignment for such a node is based on the information delivered from neighbor nodes especially in dynamic and distributed algorithms [11]. Since this information about the neighbors is not verified, the attacker can easily influence the channel assignment procedure and then the network's performance. Some of these attacks have been identified [6].In general, we can summarize these attacks exploiting the security vulnerabilities in two ways. First, the malicious node modifies the channel assignment of

its interfaces without informing its neighbors or switch to higher priority channel which effect the performance in term of available bandwidth. Second, the malicious node transmits wrong information to its neighbors that it has changed the channel assignment but actually it has not. So, a secure control message exchange mechanism between the nodes is required. For more details about these attacks refer to [6].

3.3 Cluster Channel Assignment

In [2], the *Cluster Channel Assignment* (CCA) algorithm is introduced, which we refer to in this paper. In CCA, the available channels in the network are equally distributed among the clusters in a way that two neighbor clusters get disjoint sets of channels as shown in Figure 1(a). $F_{C_i} = \{A \vee B, \ldots, G\}$ is the set of channels to be assigned to cluster C_i. CCA algorithm consists of



(a) An Example of CCA (b) CCA messages

Fig. 1. Cluster channel assignment (CCA)

two phases: Static phase and on-demand (dynamic) phase. In the static phase, each clusterhead assigns a default common channel to all nodes (routers) which belong to its cluster. After this initialization phase, each router periodically estimates the load of all its communication links and exchanges this information with its neighbors and clusterhead. This information contains the link status $P(\text{loss on } l_{i,j}^{f_i}$ and channel usage $B(v_{i,j}^{f_i}, t)$ which is illustrated in Figure 1(b)a. We also assume that a clusterhead has an overview of the load estimation and channel usage of the neighbor clusters, since it can exchange this information with its neighbors. If the router v_i experiences a loss rate $P(\text{loss on } l_{i,j}^{f_i}) \geq \sigma$ on a current link $l_{i,j}^{f_i}$ or the link usage is close to the link capacity on channel f_i which is not sufficient to the requested bandwidth from v_i's clients, it shifts to the on-demand phase. Then, v_i requests a free channel from the clusterhead by sending *F_REQUEST*message. In the second phase, the clusterhead detects the nodes having a high load and then tries to assign them new channels. These new channels cannot only be taken from the still unused channels of F_{C_i}, but can also

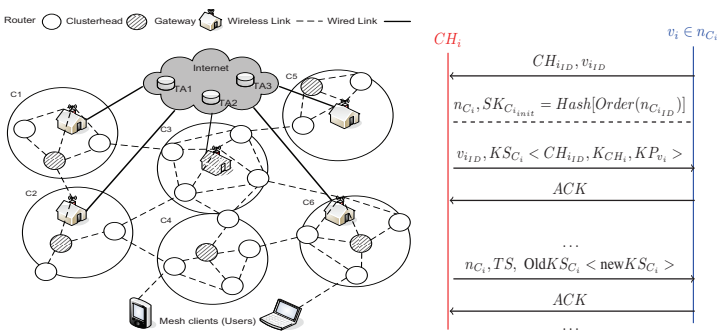
be borrowed from neighbor clusters having enough free channels. The contents of the control messages of CCA algorithm are illustrated in Figure 1(b).

After the clusterhead selects a suitable channel and suitable neighbor $u_j | u_j \in N_i$, it informs both the requested router v_i and its neighbor u_j about the new channel to switch to using a *F_SWITCH* message and executes the channel switch. It then awaits a *F_ACK* message from them acknowledging the switch. More details about this algorithm refer to [2] and [5].

4 Cluster-Based Security

Before we describe our approach in details, we state some of our assumptions:

- Clustering has already been done.
- The public key of the gateway is known by the neighbor clusterhead.
- Gateway can possibly be a clusterhead.
- There exist distributed trusted authorities (TAs) connected directly to gateways.
- A cluster may or may not have a gateway.
- The cluster without gateway will communicate with the nearest available gateway through border node.
- Each node contains a unique identifier.
- Initially (as a fist step), all communication between the nodes should be done by using a default or single channel.
- The limitation of energy consumption is not an issue since the mesh routers are placed in a fixed position and connected to the main power supply.
- The last parameter of all messages before encryption is the Hash of complete message for message integrity.



(a) SCCA Network Model (b) Initial Session Key

Fig. 2. SCCA

Table 1. Notation

Symbol	Definition
K_{v_i}	Public Key of node v_i
S_{v_i}	Private Key of node v_i
KP_{v_i}	Key Pair (public/private) of node v_i
$l_{i,j}^{f_i}$	A wireless link between node v_i and v_j on channel f_i
F_{v_i}	The set of channels assigned to node v_i
R_{v_i}	The set of WNICs of node v_i
CH_i	Clusterhead of cluster i
NC_i	Set of neighbor clusters of cluster i
n_{C_i}	Set of nodes which belong to cluster i
n_{GW_i}	Set of neighbor nodes of gateway i
SK_{C_i}	Session Key of cluster C_i
$v_{i_{ID}}$	The identity of node v_i

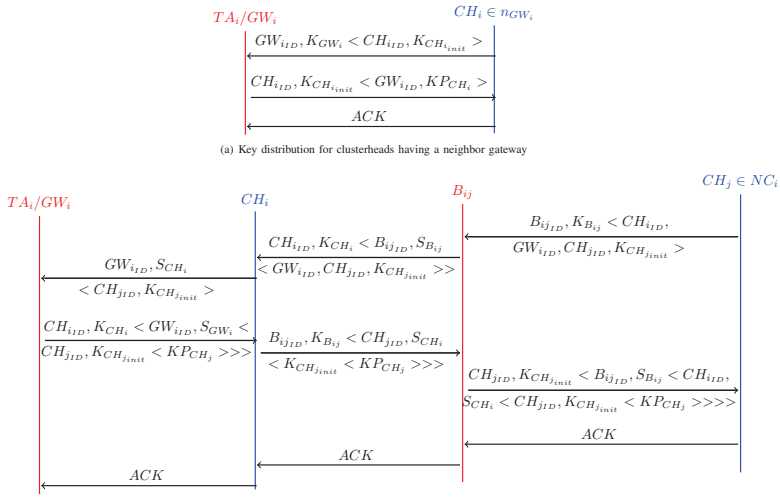
Based on these assumptions and using the terminology defined in Table 1, we discuss the *Secure Cluster Channel Assignment* (SCCA) using Figure 2(a). SCCA consists of two stages: key distribution and secure channel assignment. The key distribution also consists of two stages: in first stage, the TA generates and distributes the key pairs for the clusterheads in a secure way and then for clusters' members in the second stage.

4.1 Key distribution for a clusterhead

As a first step of key distribution for clusterheads, each clusterhead generates an initial public and private key-pair for itself. In the first case with clusters having a gateway, each clusterhead (CH_i) send its ID and its initial public key ($K_{CH_{i_{init}}}$) encrypted with the public key of the gateway (K_{GW_i}) to the gateway. Then, TA_i/GW_i verifies CH_i and generates the key pairs (KP_{CH_i}) using the identities of the CH and transfer this to CH_i as shown in Figure 3(a). For the second case with clusters not having a gateway, the CH can get its key pairs from the TA via the neighbor cluster that has a gateway and already got its key pairs. Figure 3(b) illustrates this case. The CH not having a gateway (CH_j) send its ID and $K_{CH_{j_{init}}}$ to the border node (B_{ij}) after encryption with B_{ij}'s public key ($K_{B_{ij}}$). B_{ij} relay this information to the clusterhead having a gateway (CH_i) after encryption with its private key ($S_{B_{ij}}$) and CH_i's public key. When CH_i receives this information it decrypts it with $K_{B_{ij}}$ and encrypted with its private key S_{CH_i} and relay this information to the trusted authority/gateway TA_i/GW_i. After receiving this information by TA_i/GW_i, it generates the key pairs (KP_{CH_j}) for CH_j. Then, TA_i/GW_i compiles a message with these key pairs and send it to CH_i. The same process is repeated in the other direction till the key-pair is received by the CH_j.

4.2 Key distribution for cluster members

In this section, we discuss the key distribution for the clusters' members and in the following section we explain how to secure channel assignment messages.



(a) Key distribution for clusterheads having a neighbor gateway

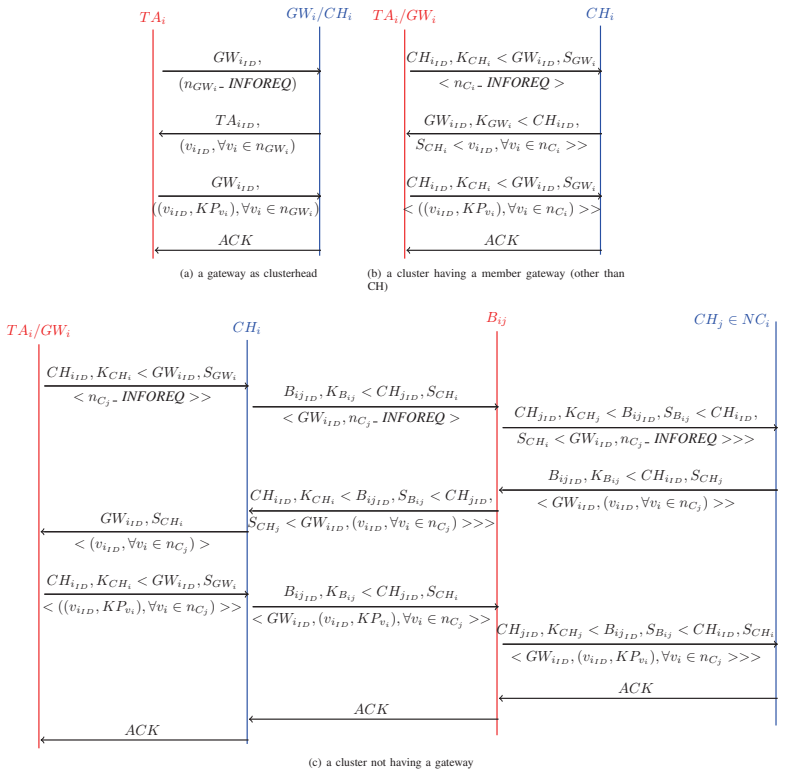
(b) Key distribution for clusters not having a gateway

Fig. 3. Key distribution for a clusterhead

For Clusters Having Gateway as Clusterhead The process starts with the respective TA_i asking the gateway about its neighbor nodes. Gateway informs the TA_i about its neighbors by sending their IDs which it received during broadcast messages having neighbor queries (clustering process). Gateway then generates an initial session key $SK_{C_{i_{init}}}$ by concatenating IDs of the neighbor nodes in ascending order (with increasing node ID values), and then taking a hash of it as in Figure 2(b) . Here we assume that the attacker has or can have information about some of the nodes but not all the nodes so the attacker cannot generate this hash (session key for key pair distribution) itself. TA_i generates the key pairs (public/private) using the identities of the mesh routers (neighbor nodes) attached to the gateway and transfer this to the gateways. Gateway using the session key KS_{C_i} encrypts these member nodes' key pairs (KP_{v_i}) and its public key K_{CH_i} and send them to the respective member nodes. This key which is used only for key distribution will only be changed if the nodes are added or deleted from the gateway/clusterhead range. Now all nodes will have their respective key pair. The member nodes send acknowledgment about receiving the keys, to the gateway (clusterhead). These messages are also shown in Figure 2(b) and in Figure 4(a). $n_{GW_i}\_INFOREQ$ is the request about member node IDs from trusted authority.

For Clusters Having a Non-Gateway Clusterhead For these types of clusters, the procedure is the same as described for gateway as clusterhead but as a last step gateway will transfer the key pairs to the respective cluster-head using its private key S_{GW_i} and the public key of the cluster-head K_{CH_i} which the

gateway already possesses. The cluster head CH_i replies with the member node IDs encrypted with its private key S_{CH_i} and public key of gateway K_{GW_i}. The gateway then sends the required key pairs KP_{v_i}. After receiving the key pairs the cluster head sends the acknowledgment ACK. This is shown in Figure 4(b).



(a) a gateway as clusterhead

(b) a cluster having a member gateway (other than CH)

(c) a cluster not having a gateway

Fig. 4. Key distribution for a cluster members

For Clusters not Having a Gateway Figure 2(a) shows an example of this case, the border node connects cluster C_3 and C_4. The gateway requests for member nodes is forwarded to the clusterhead CH_i which knows that a member node B_{ij} which is also a border node is a part of another cluster. After being queried, this border node relays the information of C_4 members (n_{C_4}) to respective TA_2 through gateway which generates the public/private key pair of

n_{C_4} and the distribution process is repeated. This is shown in the first three messages of Figure 4(c).

The cluster-head of the cluster having no gateway (e.g. C_j) sends the info of its members (n_{C_j}) encrypted with its private key (S_{CH_j}) and border node's public key $K_{B_{ij}}$ to the border node (B_{ij}). B_{ij} relay this information to CH_i after encrypting with its private key $S_{B_{ij}}$ and CH_i's public key. When CH_i receives this information it decrypts it by $K_{B_{ij}}$ and encrypted by its private key S_{CH_i} and relay this information to the trusted authority/gateway TA_i/GW_i. After receiving this information, TA_i/GW_i generates the key pairs (KP_{v_i}) for each member node $\forall v_i \in n_{C_j}$ included with this message. Then, TA_i/GW_i compiles a message with these key pairs and send it to CH_i. The same process is repeated in the opposite way till received by the CH_j. CH_j has all the information about its members and then it sends KP_{v_i} to each $v_i \in n_{C_j}$ after encrypting with its session key (KS_{C_j}). The exchange of messages is shown in Figure 4(c). In case of the cluster reconstruction, the nodes within a cluster, elect a new clusterhead and the whole process is repeated for the new clusterhead.

4.3 Secure Cluster Channel Assignment

In this section we explain how to secure the channel assignment (CCA) process for intra-cluster and inter-cluster which mentioned in subsection 3.3.

Static/default CCA After the key distribution mentioned in the above section is finished, each node in the network gets its public/private keys and the session key of the cluster that belongs to it. In this phase, the clusterhead (CH_i) sends the default channel info to all member nodes within a cluster (C_i) using its session key (SK_{C_i}). In the case of border node (B_{ij}) within two neighbor clusters C_i and C_j, it will get two distinct default channels from CH_i and CH_j. This procedure will be the same for all clusterheads.

Dynamic On-demand CCA

Channel Available in cluster: The node which has bandwidth overload or observing packet drops because of interference, would request the clusterhead for a new channel, by sending a *F_REQUEST* message. The bandwidth usage information is sent to the clusterhead by the node requiring new channel and all n_{C_i} that leads the clusterhead to have an overview of all traffic of its members nodes. This *F_INFO* message is illustrated in Figure 1(b). The clusterhead after authenticating the member node, checks whether it has a free available channel. In case of availability, it compiles the *F_SWITCH* message and sends it to the interested node. This message is encrypted with the cluster's session key. The node after allocating new channel based on the CCA algorithm, replies with an acknowledgment. The CH records this updated info (time, requested node and the new allocated channel) in its database.

Borrowing channel from neighbor cluster: If on channel request, the CH does not have any available free channel to allocate to the member node, it requests for an available free channel from its neighbor clusters. For this purpose, it sends a *F_BORROW* message to all its neighbor cluster using either border node or the cluster head (if in transmission range) to borrow a free channel frequency. This message is sent with the signature (private key) of CH_i i.e. S_{CH_i} and encrypted with the public key of the border node $K_{B_{ij}}$ as shown in Figure 5. In this figure, CH_i sends a free channel request to $CH_j \in NC_i$ or border node and then relayed to CH_j. CH_j after authenticating CH_i request, it checks for a free channel or the least used channel within its cluster. Then, it compiles a *F_INFO* message with this information and reply *F_BORROW* message. As CH_i receives all replies from its neighbors, it select the suitable free channel or at least the minimum used channel within its neighbors for the requested node based on CCA algorithm. Then, CH_i sends a *F_SWITCH* with this free channel to the requested node and updates its database.

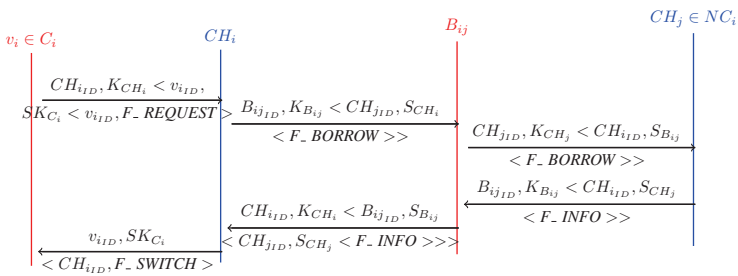


Fig. 5. Channel Borrow from neighbor cluster

5 Discussion

In this section, we talk about how our proposed solution provides sufficient security to DCA in Wireless Mesh Networks. Firstly, we discuss the common types of attacks possible during the channel assignment of wireless mesh networks and then discuss the prevention.

5.1 Link Disconnectivity Attacks and Prevention

This attack occurs if the two neighbor nodes within a cluster that want to communicate do not have one common channel link available. This can happen if the attacker gets control of one of the communicating nodes and try to deny the channel availability at that node although in reality it is there. These type

of attacks are prevented in this way that the channel assignment is done after authentication of the node and the CH maintains the channels associated with each member node in its cluster. Then, the signature attached to each message from the neighboring nodes makes sure that the valid node with valid channel is transferring the message.

5.2 False Assignment and its Prevention

If the attacker impersonates the CH and tries to issue an overloaded channel to the member nodes after new channel's demand then the communication on this channel may lead to chaos. The member node to which this node is assigned would not be able to avail full channel bandwidth as that would already be overloaded. In our model, it would not be possible as the clusterhead would also authenticate itself each time to the member nodes and the attacker will not be able to guess the corresponding session key as well as the private key of the cluster head CH.

5.3 Spam Request and its Prevention

The adversary node may try to obtain a new channel associated with it by requesting it from the CH. Since the CH maintains the complete list of the channels associated with each member node, it will simply reject the request of the adversary as the adversary will not be able to guess the private keys of the actual member node that it is impersonating as well as the session key, as this session key depends on the node IDs of all the member nodes within the CH. The malicious node can send spam request messages to the CH for applying a free channel where actually it does not need it. Since the clusterhead has an overview of the load estimation and channel usage of all its cluster members, it will simply not reply to these spam messages.

5.4 Other general security issues and attacks

Besides all these security issues that are mentioned above, our proposed approach also handles the general security issues which are not specific to channel assignment. These include active eavesdropping, information leakage and unauthorized access. Active eavesdropping will be easily detected as before message encryption, the hash value (last parameter) with each transferred message keeps the integrity and any change will be detected on the receiving side. Information leakage is not possible as all the info regarding channel assignment is encrypted. Unauthorized access requires the adversary to know the key-pair of the sender and the session key in case of communication within the cluster that are unknown to the adversary and are very difficult to be compromised.

6 Conclusion and Future Work

In this paper, a cluster-based security scheme (SCCA) is proposed to secure wireless mesh network (WMN) and specially channel assignment in WMN. The proposed SCCA scheme addresses the security vulnerabilities that exist in most of the channel assignment algorithms, specially in CCA algorithm. SCCA can provide confidential communication for cluster-based WMN and prevent the security attacks during the channel assignment. Currently, we are working on the implementation of our proposed scheme with omnet simulator and probably also on a real testbed. Moreover, we are also working on how to secure mobile clients

Acknowledgment

This work was partially supported by the German National Science Foundation (DFG) within the research excellence cluster Ultra High-Speed Mobile Information and Communication (UMIC).

References

1. Akyildiz, I.F., Wang, X., Wang, W.: Wireless mesh networks: a survey. Volume 47. (2005) 445–487
2. Makram, S.A., Günes, M., Kchiche, A., Krebs, M.: Dynamic channel assignment for wireless mesh networks using clustering. In: Proceedings of The Seventh International Conference on Networking (ICN'08). (2008) 539–544
3. Raniwala, A., Gopalan, K., cker Chiueh, T.: Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. Volume 8., New York, NY, USA, ACM (2004) 50–65
4. Ramachandran, K., Belding, E., Almeroth, K., Buddhikot, M.: Interference-aware channel assignment in multi-radio wireless mesh networks. In: INFOCOM. (2006)
5. Makram, S.A., Günes, M.: Distributed channel assignment for multi-radio wireless mesh networks. In: Proceedings of IEEE Symposium on Computers and Communications (ISCC'08). (2008) 272–277
6. Naveed, A., Kanhere, S.S.: Security vulnerabilities in channel assignment of multi-radio multi-channel wireless mesh networks. In: GLOBECOM. (2006)
7. Lin, X., Lu, R., Ho, P.H., Shen, X., Cao, Z.: A novel compromise-resilient authentication system for wireless mesh networks. (March 2007) 3541–3546
8. Jin, Z.A., Park, G.D., Yoo, K.Y.: An improved secure authenticated group key agreement protocol for wmns. In: ALPIT '08: Proceedings of the International Conference on Advanced Language Processing and Web Information Technology, IEEE Computer Society (2008) 412–417
9. Sun, J., Zhang, C., Fang, Y.: A security architecture achieving anonymity and traceability in wireless mesh networks. In: INFOCOM, IEEE (2008)
10. Xuyang, D., Mingyu, F., Xiaojun, L., Dayong, Z., Jiahao, W.: Multi-path based secure communication in wireless mesh networks. Volume 18., Elsevier, ScienceDirect (December 2007) 818–824
11. Haq, A., Naveed, A., Kanhere, S.S.: Securing channel assignment in multi-radio multi-channel wireless mesh networks. In: IEEE WCNC, IEEE (March 2007)

Protecting Receiver Privacy in Routing for Wireless Sensor Networks

Edith C.-H. Ngai[1] and Brittle K.-H. Tsoi[2]

[1] Department of Information Technology, Uppsala University, Sweden
edith.ngai@it.uu.se
[2] Department of Computing, Imperial College London, United Kingdom
khtsoi@doc.ic.ac.uk

Abstract. Many existing routing algorithms in wireless sensor networks require the destination address in the header field of the packets which may expose the location of the receiver to the attackers. In this work, we propose a random subpath routing scheme which protects receiver location privacy in geographic routing against node compromise and eavesdropping attacks. The original path from the base station to the sensor is divided into subpaths based on the selected checkpoints. Location of these checkpoints are encrypted, such that a sensor only knows the end node on its subpath. The path is further prolonged and randomized to increase the difficulty for an attacker to capture the receiver location or predict its direction. We evaluate the packet delay, security strength, and energy consumption of the proposed scheme by both analysis and simulations. The performance of our scheme can be tuned through a couple of parameters that determine the tradeoff between the protection strength and the overheads.

Keywords: wireless sensor networks, routing, security and privacy

1 Introduction

Wireless sensor network (WSN) is composed of a number of small sensing devices with limited communication range. The sensors can collect data samples from the environment and perform multihop routing to transport the data to the destination. The sensing and wireless technologies are promising to be widely deployed in a broad spectrum of civil and military applications in the near future [1].

A lot of research has been conducted to investigate data collection from large amount of sensors to the base station (BS) [1][2][3]. Different from the popular many-to-one communication pattern, we examine private message delivery from BS to individual sensor. This particular one-to-one communication pattern occurs in many mission critical applications which require high level of privacy and confidentiality. For example, soldiers equipped with sensors may be trained to provide different services, like attacking the enemy, destroying some military facilities, caring for the wounded, repairing tanks and so on. The mission, which is assigned by the BS, will be delivered through packets to the particular soldier. Both the content of the mission and the location of the soldier have to be treated as confidential before the action is taken.

In this situation, both the message secrecy and the receiver privacy are very important as the mission plan will be revealed once the message is disclosed or the receiver location is captured. The message can be encrypted with the symmetric keys to provide confidentiality [4], but there are still several ways that an attacker can capture or predict the location of the receiver. We consider two types of attackers, namely the strong attackers and the weak attackers. The strong attackers have more computation power which allows them to compromise a node and obtain its key, such that they can decrypt a packet and read the receiver location in the packet. This kind of attack is vulnerable to traditional geographic routing [5] as the location of receiver must be included in the destination field of the packet for routing. On the other hand, the weak attackers, who are not able to compromise a key, may perform eavesdropping and predict the receiver location by analyzing the network traffic.

Dummy packets injection has been proposed to provide location privacy in WSNs, but it increases the network traffic, packet delay and energy consumption [3][6][7]. Different random routing algorithms are suggested to confuse the eavesdropper from predicting the receiver location. However, randomized routing solely cannot protect the network against strong attackers who can compromise the key of a node and read the receiver location in packets of the popular geographic routing protocols in WSNs [5][8][9]. Moreover, the communication patterns and the continuous data streams in the above network models are different from the mission critical applications which involve infrequent and small amount of traffic.

To address the privacy message delivery problem for mission critical applications in WSNs, we design a novel routing scheme that can obscure the receiver location from compromised nodes or network traffic eavesdropping.

The contributions of this paper include:

- a novel routing scheme featured with random checkpoints and path prolongation for protecting receiver privacy in routing (Section 3 and 4).
- an analytical model which allows the study of protection strength, packet delay and energy consumption of the proposed scheme (Section 5).
- comparison of the overhead between the proposed scheme and the original shortest path routing scheme in terms of protection strength, packet delay and energy consumption (Section 6).

2 Related Work

Contextual privacy issues in sensor networks, especially location privacy [6][7][10][11], have been studied in recent years. The random walk based phantom flooding scheme [7] is proposed to defend against an external adversary who attempts to trace back to the data source in sensor networks and provide source location privacy of the BS. A path perturbation algorithm [12] is also proposed to cross paths in areas where at least two users meet which intends to make the attacker confuse the paths of different users. Although the random routing approach can protect the network from local adversaries who overhear and analyze the traffic passively, stronger attackers can still capture the receiver location from an encrypted packet.

Several schemes, like ConstRate and ProbRate, which introduce dummy traffic to hide the real event sources, are proposed to provide source event unobservability in the network [3][13]. They consider source nodes which generate continuous data streams and large amount of traffic, which are different from the scenario in our mission critical applications. Even though some dummy packets can be dropped on their way [13], the injected dummy traffic still increases the packet delay and consume more energy in sensor nodes.

Techniques of multpath routing and fake message injection are also introduced [2] to provide receiver privacy. However, they concentrate on the traffic-analysis attack, which determines the BS's location through the measurement of traffic rates at various locations. A recent work is proposed to protect receiver-location privacy in WSNs by providing path diversity in combination with fake packet injection [6]. It is solving a similar problem as us, but it considers only weak attackers who capture the receiver by eavesdropping and network traffic analysis. It also assumes continuous data streams which provide large amount of traffic samples and time for the attacker to trace the receiver location. On the contrary, we aim at a routing scheme that can protect the network in presence of both weak and strong attackers in the typical geographic routing for WSNs.

3 Network and Threat Models

3.1 Network Model

A wireless sensor network consists of a number of sensors deployed in an area, together with one or multiple BS(s). Each sensor has a transmission range. They forward messages from the source to the destination hop-by-hop based on the popular geographic routing protocols [5][8]. In geographic routing, a sensor node knows the destination of the packet and forwards the packet to the next hop that is located closer to the destination.

Since sensors have limited computation, storage, and communication resources, they cannot afford to use asymmetric cryptography. Instead, they use symmetric cryptographic primitives to provide data confidentially, authentication, integrity, and freshness of the message [4][14]. Each sensor i shares an unique symmetric key K_i with the BS.

We focus on private message delivery from the BS to individual sensor. The receiving node of the private message is informed to carry out some secret missions. The message therefore requires high privacy and confidentiality. Both the content of the message and the location of the receiver need to be protected.

3.2 Adversary Models

We consider two kinds of attackers in the network, namely the strong attackers and the weak attackers. Details of their attack models are presented below. Note that the BS is supposed to have strong protection capability, so it is trustworthy. This is also a common assumption in WSN security [4][14] [15].

– Strong attacker:
  Strong attackers are equipped with strong computation capability and power which allow them to attack a node more actively. An attacker may compromise a node and obtain its secret key, then it can decrypt the packet to the node and capture the location of the receiver. The attackers also understand the security protection mechanisms well. Their only goal is to capture the receiver, so they will not be discovered without interfering the proper functionality of the network.
– Weak attacker:
  Weak attackers are only equipped with some supporting devices, such as antenna, which allow them to eavesdrop the delivery of packets and perform some simple traffic analysis. However, they are not able to compromise the key of a node or decrypt the message, though they can predict the direction of the receiver based on the signals that they overheard.

3.3 Notations

We use the following notations to describe the cryptographic operations in this paper which are mainly adopted from [4].

– $M1|M2$ denotes the concatenation of messages $M1$ and $M2$.
– K_i denotes the secret (symmetric) key that is shared between node i and the BS.
– $E = \{M\}_{K_i}$ is the encryption of message M with the symmetric key shared by node i and BS.
– $\{E\}_{K_{mac}} = \{E, MAC(K_{mac}, C|E)\}$ is the signed message which contains the encrypted message E and its MAC (message authentication code) from the BS, where K_{mac} is the MAC key of BS and C is the counter value.

4 Random Subpath Routing for Private Message Delivery

4.1 Path Partition and Prolongation

The location of the receiving node is included in a packet in geographic routing protocols [5][8][9]. Each intermediate node, which receives the packet, will forward it to next hop that is located closest to the destination. Although many encryption and authentication techniques [4][14] can be applied, attackers who compromised an intermediate node can still capture the receiver as its location must be known for routing purpose.

The problem is to prevent the attackers from obtaining the location of the receiver even if they compromise an intermediate node along the path. The BS divides the original path into several sub-paths based on the selected checkpoints. It then encrypts and inserts the location of the destination nodes (or checkpoints) of the sub-paths in the packet. For instance, the original path from BS to destination d, $Path_{org} =< BS \rightarrow ... \rightarrow d >$, will become $Path_{new} =< BS \rightarrow ... \rightarrow a >< a \rightarrow ... \rightarrow b >< b \rightarrow ... \rightarrow c >< c \rightarrow ... \rightarrow d >$, where a, b, and c are the checkpoints introduced on the path, d is the final destination of the packet, $< BS \rightarrow ... \rightarrow a >$ is the subpath from BS to a. The checkpoints can be chosen to divide the original path into subpaths with equal

length as shown in Figure 1. There are two checkpoints in Figure 1(b), where the last checkpoint is also the receiver of the message.

The original packet for delivering the message from BS to receiver d is

$$\{a|\{M\}_{K_d}\}_{K_{mac}}, \tag{1}$$

where M is the secret message containing the mission details, K_d is the secret key between node d and BS, and K_{mac} is the MAC key of BS. If the MAC generated can be verified by the sensor correctly, the sensor can be assured that the message is originated from the BS.

After path partition, the new message is

$$\{a|\{\{b\}_{K_{mac}}\}_{Ka}|\{\{c\}_{K_{mac}}\}_{Kb}|\{\{d\}_{K_{mac}}\}_{Kc}|\{M\}_{K_d}\}_{K_{MAC}}, \tag{2}$$

where K_a, K_b, K_c, and K_d are the secret keys between the BS and nodes a, b, c and d respectively. Since the location of a checkpoint is encrypted by the secret key of the previous checkpoint, the intermediate nodes only know the end node on their subpaths, but not the intended receiver of the message.

Apart from that, BS can make the destination location even more confusing by prolonging the path and inserting more checkpoints after the destination node d. For instance, it may add nodes e and f after d, such that, $Path_{new} =< BS \rightarrow ... \rightarrow a >< a \rightarrow ... \rightarrow b >< b \rightarrow ... \rightarrow c >< c \rightarrow ... \rightarrow d >< d \rightarrow ... \rightarrow e >< e \rightarrow ... \rightarrow f >$. In this case, even the attacker knows that the node f is the end of the path, it still cannot tell whether it is the intended receiver of the message. Figure 1(c) shows the prolonged path with three checkpoints.

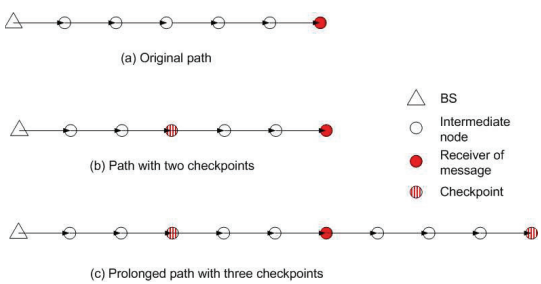


Fig. 1. Original path is prolonged and divided into subpaths.

4.2 Checkpoint Selection

The path partition and prolongation techniques prevent strong attackers from capturing the receiver location in the packet. However, the attackers can still predict the receiver

location easily if the packet from BS is routed along the shortest path towards the receiver in single direction. The adversary who is equipped with antenna or spectrum analyzer can overhear the signals and predict the receiver location. We propose a simple random checkpoint selection mechanism to handle this problem.

Instead of choosing the checkpoints along the shortest path, the sink selects the checkpoints from a larger area as shown in Figure 2. The area has a width of $H = H1 + H2$ in this example. Note that $H1$ and $H2$ are not necessarily equal, such that the attackers cannot predict where the shortest path lies based on the location of the intermediate nodes. The region is further divided into segments based on the number of checkpoints it wants to achieve. One checkpoint will be selected in each segment.

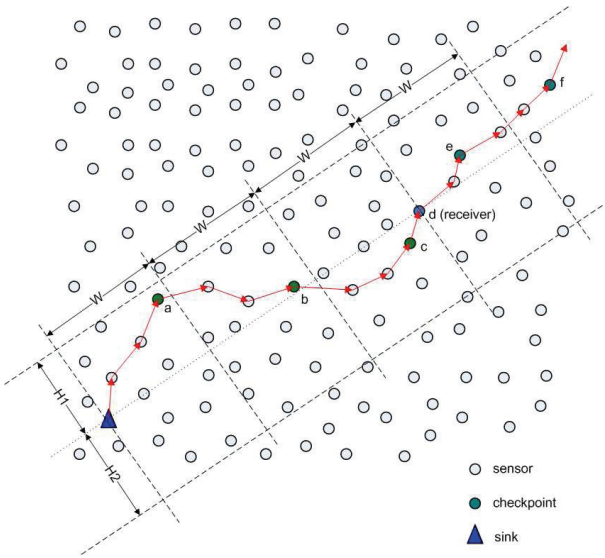


Fig. 2. Selection of the checkpoints.

5 Analytical Results

5.1 Packet Delay

A message receiver is located at a distance D measured in number of hops from the BS. Supposed that n checkpoints are selected in the segmented area. The new distance becomes $D' = D1 + D2 + ... + D_i + ... + D_n$, where D_i is the distance in number of hops from the $(i-1)^{th}$ checkpoint to the ith checkpoint. Intuitively, D' will be longer

when $H1$ and $H2$ increase as the checkpoints are spread farther away from the shortest path.

In order to consider the packet delay, we compute the average delay $\overline{Q}$ (queueing and transmission) through each hop which we will approximate using a M/M/1 queue yielding:

$$\overline{Q} = \frac{\mu^{-1}}{1-\rho}, \tag{3}$$

where $\rho = \lambda/\mu$, λ is the incoming rate of packets, and $1/\mu$ is the link transmission delay per packet plus any processing delay through a node.

The average total packet delay is then

$$E[T] = [\frac{\mu^{-1}}{1-\rho}]D'. \tag{4}$$

5.2 Protection Strength

We evaluate the strength of privacy protection against the strong and weak attackers who intend to capture the receiver location.

Strong Attackers A stronger attacker can compromise a node and read the destination node of that particular subpath from the packet. The probability that the strong attacker can capture the receiver location on a path is:

$$P_r = \sum_{i=1}^{n} k_i \hat{p}_i / D', \tag{5}$$

where n is the number of checkpoints (or subpaths), k_i is the length of subpath i in number of hops, D' is the total path length, $\hat{p}_i$ is the probability that the end node on subpath i is the intended message receiver, given that $\sum_{i=1}^{n} \hat{p}_i = 1$.

Suppose that each checkpoint has equal probability to be the intended message receiver and the number of nodes on all subpaths are the same, i.e. $k_i = k, \forall i$, where k is a constant. Then,

$$P_r = k/D' = 1/n. \tag{6}$$

If there are n strong attackers in the network, it is possible that they may work together and try to discover all the checkpoints. The probability that all checkpoints are discovered is $p_1 p_2 ... p_i ... p_n n!$, where $p_i = k_i/D'$ is the probability that the attacker captures a node on the ith subpath. However, even though the attackers can identify all checkpoints along the path, they still cannot decide which of them is the intended receiver of the message.

Weak Attackers An attacker may predict the direction of the receiver based on the packet travelling direction from BS to the overheard intermediate node. Suppose the attacker assumes shortest path in routing, the probability for the attacker to know the exact direction of the receiver is:

$$P_c = N_c/D', \tag{7}$$

where N_c is number of common nodes between the path in random subpath routing and the original shortest path routing.

If there are multiple attackers in the network, they may try to locate the region that contains all the checkpoints. It is uneasy to circle the entire area, unless there are enough attackers in the network. Furthermore, even though they can identify the area where the checkpoints are located, the probability P_s to identify the receiver is still low if the number of nodes in the segmented area is large.

$$P_s = 1/N_s, \tag{8}$$

where $N_s = WHn\sigma$ is the number of nodes in the area that the checkpoints are located and σ is the node density in the network.

5.3 Energy Consumption

The path length increases from D to D' hops in routing with checkpoints. The energy consumption can be calculated as

$$E = D'E_v, \tag{9}$$

where E_v is the energy for transmitting and receiving a packet from one node to another.

6 Simulation Results

We have conducted extensive simulations with *ns-2* [16] to evaluate our proposed random subpath routing algorithm. The network settings are summarized in Table 1 which are mainly drawn from existing works [8] [9]. The results of shortest path geographic routing [5] are also presented for comparison.

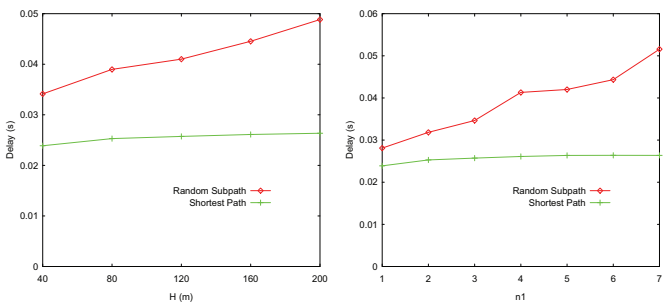
Table 1. Simulation Parameters

Network size	400m x 400m
Sensor distribution	Uniform random
No. of sensors (N)	400
Radio range	40m
Packet size	32bytes
MAC layer	IEEE 802.11
Transmit data rate	76.8kbps
Radio receiver current	9.6mA
Radio transmitter current	16.5mA
Supply voltage	3V

6.1 Packet Delay

We measure the average packet delay from the BS to the destination sensors, which are randomly generated along the diagonal of the network area. We fix the number of checkpoints from the BS to the receiver as $n1 = 5$ and the number of checkpoint in the prolonged path as $n2 = 1$. Figure 3(a) shows the average packet delay of random subpath routing varying H. The results of shortest path geographic routing, in which the packets always flow along the same path, is also shown for comparison. When H increases, the checkpoints of random subpath routing are selected in a larger segmented area with greater randomness. Its average packet delay becomes higher than that in shortest path routing due to the increased path length.

Similarly, Figure 3(b) shows the average packet delay varying $n1$ with $n2 = 1$ and $H = 120m$. The average packet delay in random subpath routing again is higher than shortest path routing as its path is prolonged with more checkpoints.

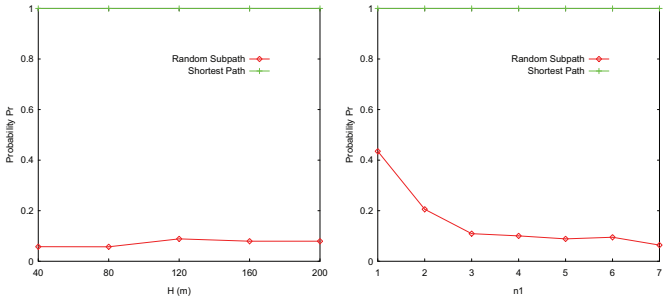


(a) (b)

Fig. 3. Packet delay (a) varying H (b) varying $n1$.

6.2 Protection Strength

Figures 4(a) and (b) show the probability P_r for an attacker to obtain the receiver location after capturing a node on a path. Since shortest path routing keeps the receiver location open in the packet header, the attacker can always read it after capturing any of the intermediate nodes. On the contrary, random subpath routing only discloses the next checkpoint on a subpath, so the attacker has much lower probability to obtain the location of the intended message receiver. The results also show that the attacker has slightly higher probability to know the receiver location when H increases. The reason is that the increased path length allows the attacker to have higher chance to compromise any nodes on the subpath. On the other hand, the attacker has lower probability to discover the intended receiver when the number of checkpoints increases.



(a) (b)

Fig. 4. Probability P_r (a) varying H (b) varying $n1$.

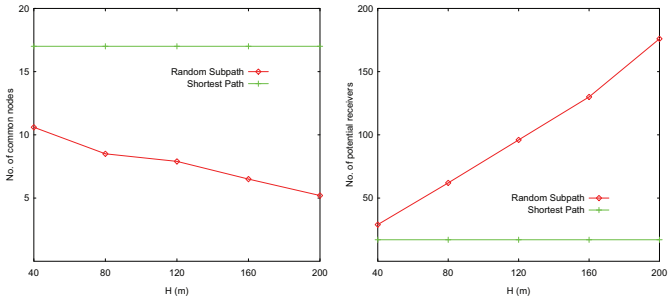
An attacker may predict the direction of the receiver location easily if the path in random subpath routing is close to the shortest path in geographic routing. We then evaluate the privacy protection strength by counting the number of common nodes between the paths in the two schemes. Figure 5(a) shows that random subpath routing reduces the number of common nodes effectively when H increases. The curve of shortest path routing is also plotted to illustrate that the attacker can predict the receiver direction from any nodes along the path.

Multiple attackers may also predict the location of the receiver by overhearing the traffic at multiple locations. Figure 5(b) shows the number of nodes in the area where the receiver may be located. Random subpath routing can protect the receiver effectively as the intermediate nodes are spread on a larger area. The greater number of nodes in the potential area makes the prediction of receiver location more difficult. Since the packets in shortest path routing follow only one direction, the receiver is limited to the nodes along the shortest path. Thus, its protection strength is much weaker.

6.3 Energy Consumption

Figure 6 shows the average energy consumption for delivering a packet in the two schemes. Random subpath routing consumes more energy than shortest path routing. It is because more intermediate nodes are involved in transmitting and receiving the packet. The energy consumption of random subpath routing also increases with H and $n1$ as the path becomes longer.

Overall, the simulation results demonstrate that random subpath routing can provide higher privacy protection strength, but it also increases the packet delay and the energy consumption. The parameters n and H which determine the number of checkpoints and the randomness of the path have been taken account for the tradeoff between the security strength and efficiency.



(a) (b)

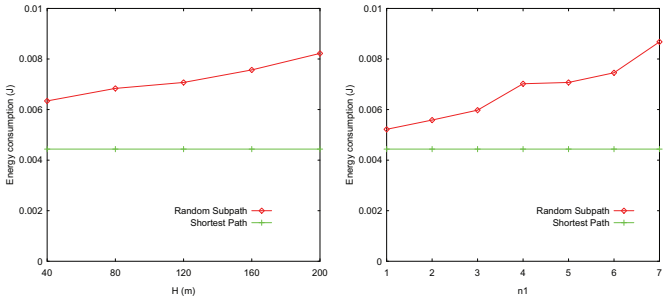
Fig. 5. (a) Number of common nodes N_c on path (b) Number of nodes N_s in the segmented area.

7 Conclusions

In this paper, we have proposed a random subpath routing scheme to protect receiver privacy from the BS to sensors in presence of node compromise and eavesdropping attacks in WSNs. The prolonged and randomized path with encrypted checkpoints prevents the attackers from capturing the receiver location in the vulnerable geographic routing protocols. The random selection of the checkpoints further increases the difficulty for an adversary to predict the direction of the receiver. We have evaluated our scheme by both analysis and simulations. The results demonstrated the security strength and effectiveness of the proposed scheme on protecting receiver privacy. Moreover, there is a tradeoff between the protection strength and the overheads on delay and energy consumption which can be controlled by the number of checkpoints, the length of the prolonged path and the randomness in checkpoint selection. In the future, we will extend our work to handle more communication patterns and investigate other attack models such as the insider attacks.

References

1. Akyildiz, I.F., Su, W., Sandarasubramaniam, T.: Wireless sensor networks: a survey. Computer Networks **38**(5) (2002) 393–422
2. Deng, J., Han, R., Mishra, S.: Countermeasures against trafic analysis attacks in wireless sensor networks. In: Proc. of IEEE/CreateNet International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm). (2005)
3. Yang, Y., Shao, M., Zhu, S., Urgaonkar, B., Cao, G.: Towards event source unobservability with minimum network traffic in sensor networks. In: Proc. of ACM WiSec, Alexandria, Virginia, USA (Apr 2008)
4. Perrig, A., Szewczyk, R., Tygar, D., Wen, V., Cullar, D.: Spins: security protocols for sensor networks. Wireless Communications **8**(5) (2002) 521–534



(a) (b)

Fig. 6. Energy consumption (a) varying H (b) varying $n1$.

5. Karp, B., Kung, H.: GPSR: Greedy perimeter stateless routing for wireless networks. In: Proc. of ACM Mobicom, Boston, Massachusetts, U.S. (2000)
6. Jian, Y., Chen, S., Zhang, Z., Zhang, L.: Protecting receiver-location privacy in wireless sensor networks. In: Proc. of IEEE Infocom. (2007) 1955–1963
7. Kamat, P., Zhang, Y., Trappe, W., Ozturk, C.: Enhancing source-location privacy in sensor network routing. In: Proc. of IEEE ICDCS, Columbus, Ohio, USA (Jun 2005)
8. He, T., Stankovic, J., Lu, C., Abdelzaher, T.: SPEED: a real-time routing protocol for sensor networks. In: Proc. of IEEE ICDCS, Providence, RI, U.S. (May 2003) 46–55
9. Felemban, E., Lee, C.G., Ekici, E.: MMSPEED: multipath multi-speed protocol for QoS guarantee of reliability and timeliness in wireless sensor networks. IEEE Trans. on Mobile Computing **5**(6) (Jun 2006) 738–754
10. Gruteser, M., Schelle, G., Jain, A., Han, R., Grunwald, D.: Privacy-aware location sensor netwroks. In: Proc. of USENIX Workshop on Hot Topics in Operation Systems (HotOS IX). (2003)
11. Al-Muhtadi, J., Campbell, R., Kapadia, A., Mickunas, M.D., Yi, S.: Routing through the mist: privacy preserving communication in ubiquitous computing environment. In: Proc. of IEEE ICDCS. (2002)
12. Hoh, B., Gruteser, M.: Protecting location privacy through path confusion. In: Proc. of IEEE/CreateNet International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm). (2005)
13. Shao, M., Yang, Y., Zhu, S., Cao, G.: Towards statistically strong source anonymity for sensor networks. In: Proc. of IEEE Infocom. (2008)
14. Eschenaur, L., Gligor, V.: A key-management scheme for distributed sensor networks. In: Proc. of the 9th ACM Conference on Computer and Communication Security. (2002)
15. Karlof, C., Wagner, D.: Secure routing in sensro networks: attacks and countermeasures. In: Proc. of the 1st IEEE Workshop on Sensor Network Protocols and Applications. (May 2003) 1–15
16. Fall, K., Varadhan, K.: The ns manual. (Dec 2003) http://www.isi.edu/nsnam/ns.

Protocols for Distributed AAA Framework in Mobile Ad-hoc Networks

Sondes Larafa and Maryline Laurent-Maknavicious

CNRS Samovar UMR 5157, TELECOM & Management SudParis, 9 rue Charles Fourier, 91011 EVRY, FRANCE
sondes.larafa@it-sudparis.eu Maryline.Maknavicius@it-sudparis.eu

Abstract. Ad-hoc networks are subject to malicious attacks given their wireless nature and high dynamicity. Various security issues have been raised so far, especially access control. In a previous work, we focused on a light and distributed AAA framework using a six message-AAA protocol, and an access token mechanism. In this article, we optimize this AAA protocol through the definition of two new AVPs. Besides to resist to spoofing and replay attacks, we design a two-way protocol that allows destination nodes to check the access authorization validity of their neighbors.

Keywords: Ad-hoc networks, distributed AAA infrastructure, access phase, access token, SEND

1 Introduction

Ad-hoc networks are wireless networks able to self-configure with no administrator's assistance. They are known as infrastructure-less networks, i.e. with no central network entity for supporting packet routing. Ad-hoc networks might be very dynamic. A mobile node joins an ad-hoc network simply by connecting to the nearest already connected nodes. Once a mobile node is connected, it has three functions: transmitting and receiving data, in addition to routing.

Ad-hoc networks are very useful for supporting military and rescue operations because they are simple to set up and remain operational as long as there are enough nodes to relay traffic. They are likely to play an important role in the future networks by extending the operators networks coverage. This would enable other users to get access even if they can not directly reach the networks. A crucial prerequisite for this, however, is the availability of suitable authentication, authorization and charging mechanisms to ensure revenues for operators [1].

AAA (Authentication, Authorization and Accounting) infrastructures provide these functions. They especially ensure security by applying access control. In the wireless, dynamic and infrastructure-less context of ad-hoc networks, we are concerned by a distributed AAA infrastructure. We gave a detailed description of its design in [2] where we proposed a six-way protocol for mutual

authentication, and an access token to be used during the access phase after authentication and authorization phase.

In the second section of this document we give an overview of our distributed AAA infrastructure compared to centralized AAA infrastructures. Then in the third section we review our six-way authentication protocol for which we give an optimization through the definition of two new AVPs. The fourth section deals with the access token usage to secure the access phase. We demonstrate that it is vulnerable to spoofing and replay attacks if we just rely on SEND protocol for its validation process. That is why we introduce a two-way protocol to be executed between each newly authenticated node and any other selected destination node during the access phase.

2 Distributed AAA infrastructure for ad-hoc networks

2.1 Existing Centralized AAA infrastructure

AAA infrastructures are classically used by network operators and service providers to control the access to their networks and services, and also to perform accounting operations for next charging their subscribers. These ones are first authenticated then granted access to certain resources.

Figure 1 shows a AAA infrastructure composed typically of a Client Node (CN - the subscriber), the AAA server performing the authentication, and an access point which relays the access requests from the CN to the AAA server. The access point is also an Enforcement Point (EP) [1] which filters the traffic of non authenticated CNs.

A classical AAA infrastructure refers to the two following protocols:

- An access protocol, like PANA, that supports the communications between the CN and the access point for the authentication and authorization. As such, the CN integrates a PANA client and the access point a PANA authenticator.
- A AAA protocol that supports AAA exchanges between the AAA client within the access point and the AAA server.

PANA and AAA messages carry the same EAP (Extensible Authentication Protocol) messages [3]. Depending on the authentication method used, a fixed number of EAP messages transport authentication and authorization material for session establishment between the CN and the access point, thus between the CN and the network.

Because lightness of ad-hoc networks solutions is very important, we conceived a distributed AAA infrastructure that employs only AAA protocols (cf. section 2.2).

AAA exchanges cover three phases:

[1] The EP is responsible for enforcing policies with respect to authentication of subscribers, authorization to access and services, accounting and mobility, etc

– Authentication and authorization phase: EAP messages carry the authentication and authorization elements. An EAP Success message is sent to the CN if the authentication was successful and a session is established between the CN and the access point.
– Access phase: CN sends traffic through the access point. The EP does not filter this traffic.
– Accounting phase: the access point sends CN's consumption information to the AAA server.

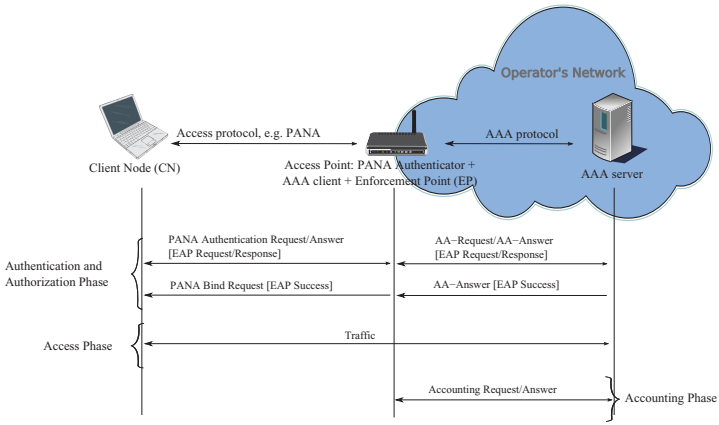


Fig. 1. Centralized AAA Infrastructure

2.2 Distributed AAA Infrastructure as defined in [2]

Introduction of AAA functionalities into ad-hoc networks is needed, as underlined in the introduction (cf. section 1). However it is inadequate to exploit centralized AAA frameworks given the decentralized nature of ad-hoc networks. That is why we designed a distributed AAA infrastructure [2] that has the following features.

n ($n \geq 1$) AAA servers (AAA_1, AAA_2,..., AAA_n), rather than one centralized AAA server, ensure the AAA service. Each CN is a AAA client so it does not need to contact an access point to join the AAA servers. This saves the exchanges between the access point and the CNs. Moreover when a CN (i.e. a AAA client) has been correctly authenticated, it acquire the functions of an Enforcement Point. This means that it becomes responsible for examining the arriving traffic and for filtering it if the generator node was not authorized.

During authentication and authorization phase, a CN contacts at least t ($1 \leq t \leq n$) AAA servers in order to be authenticated correctly. t is the threshold

number required by [2] based on the principle of Shamir Secret Sharing [4]. In [2], we adapted the ISO [9798-3] three-way authentication protocol [5] to our distributed AAA service thanks to the principle of Shoup's signature shares [6]. The resulted authentication protocol makes use of public key certificates that can be initialized into CNs by a third party (e.g. a Service Provider).

Mutual authentication occurs between a CN and the AAA servers. The CN authenticates itself first in order to avoid overloading the servers with heavy ciphering and deciphering operations during the first exchange. This improves resistance to denial of service attacks.

At the end of the authentication and authorization phase, AAA servers send an access token to the CN. This token is necessary during the access phase because it proves that the CN was successfully authenticated by the AAA service and authorized to use the ad-hoc network.

Since at least t AAA servers are required for CN's authentication, and because our authentication protocol consists of six exchanges, at least $6 \cdot t$ messages are necessary to achieve the authentication of one CN. It is obvious that the number of authentication messages increases faster than t given the coefficient 6. Optimization of the number of the protocol exchanges is then a necessity.

3 Distributed AAA Protocol Optimization

3.1 Distributed AAA Protocol as defined in [2]

Figure 2 depicts our six way-protocol as we defined it in [2]. The AAA service consists of n servers ($n \geq 3$), and the threshold number t is equal to 3. A joining node (JN - a CN) wishes to join the network. It obtains the list of the AAA servers available as explained in [2]. Then it initiates AAA exchanges with 3 of them e.g. AAA_1, AAA_2, and AAA_3. It sends its identity in the first exchange. The AAA servers reply in the second exchange with a random number R_{AAA} chosen jointly by them.

The third and fourth exchanges are inspired from the ISO three way-protocol since we distributed it using Shoup's principle. In the third exchange, JN signs with its private key the random R_{AAA} together with some data, namely a random number R_{JN} that it generates and the AAA service identity ID_{AAA}. AAA servers examine the validity of this signature. Then each server generates a piece of the AAA service signature and sends it in the fourth exchange. JN combines these pieces of signature according to Shoup's principle [6]. A signature computed by an entity A and sent to an entity B authenticates A towards B if B establishes this signature integrity. So if JN's signature and AAA service's signature are valid, the mutual authentication between JN and the AAA service is achieved successfully.

AAA servers reply if they consider that the JN authentication was successful. Similarly JN triggers the fifth exchange if it succeeds to authenticate the AAA service. The sixth exchange carries the access token of the JN: AAA servers agree on a deadline T_{JN} after which JN will have no longer access to the network and

will have to re-authenticate itself. Each server concatenates T_{JN} with the IP address of the JN and signs the result following Shoup's principle (like it does in the fourth exchange). The resulted piece of signature is concatenated to the deadline T_{JN} (cf. Fig. 2). Please have a look on the section 4 for further details about the access token and its use during the access phase.

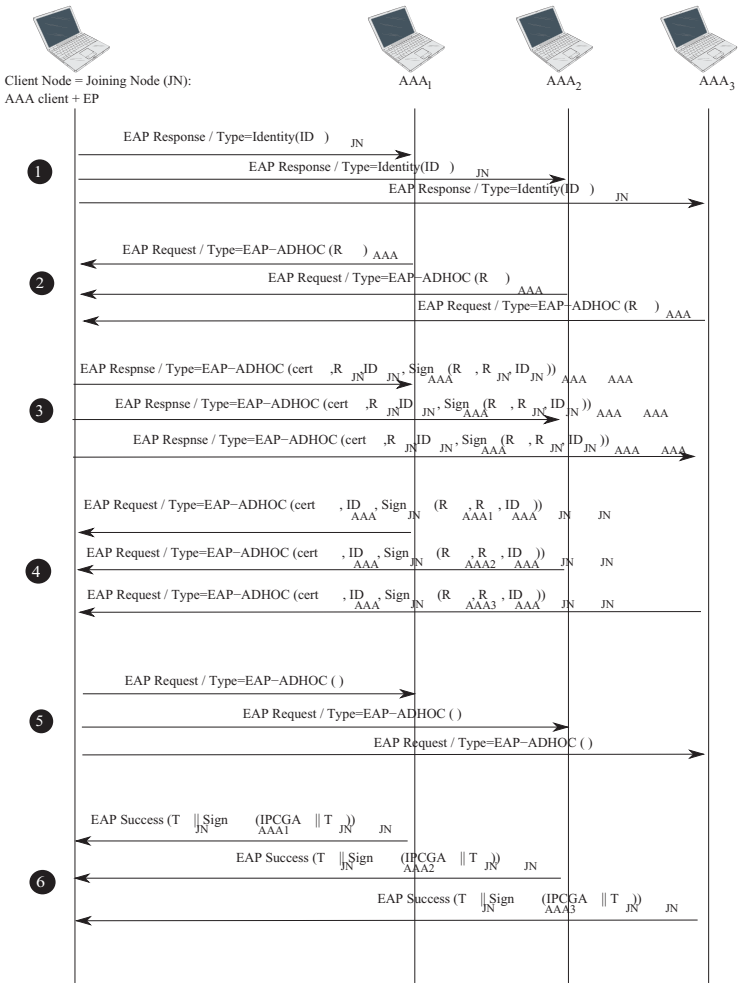


Fig. 2. AAA protocol in distributed AAA infrastructures [2]

We named EAP-ADHOC the distributed authentication method that we designed based on ISO [9798-3] norm.

3.2 Optimization

To optimize the protocol described in this section, the three last exchanges are replaced by one exchange (cf. Fig. 3). The fifth exchange is in fact an empty message where the JN informs the AAA service that the mutual authentication was successful. It induces the access token sending by the AAA service. Actually the success of JN's authentication is a sufficient reason for AAA servers to send the access token. Then it is up to the JN to trust or not the information sent by the AAA service.

Now in the optimization, the last message is an EAP Success which informs the JN that its authentication was successful. It carries the authentication information of the AAA service and the access token of the JN. Both are encapsulated into two new AVPs (Attribute Value Pair): AAA Authentication AVP encapsulates the authentication information and the Access Token AVP encapsulates the access token (cf. section 3.3.).

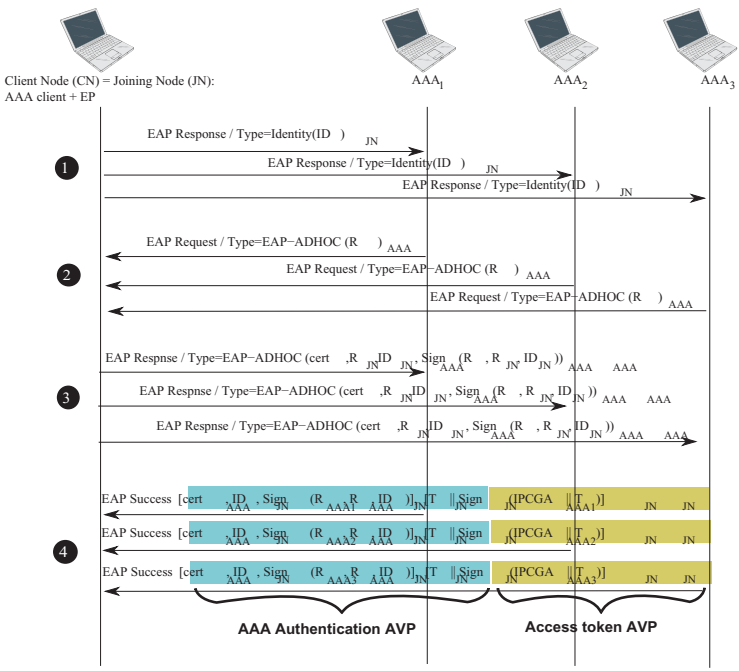


Fig. 3. Optimized AAA protocol in distributed AAA infrastructures

3.3 New AVPs

AVPs are tuples $< attribute\ name, value >$ that carry specific authentication, accounting, authorization, routing and security information as well as configuration details for the AAA exchanges. For example all EAP messages for JN's authentication shown in the figure 3 are transported in EAP-Payload AVPs. However there is no AVP expected to transport AAA service authentication information at the same time as the EAP Success message in the EAP-Payload AVP (cf. the last exchange in the optimized protocol, section 3.2).

AAA protocols such as Radius [7] and Diameter [8] support creation of new attribute value pairs for the new applications needs. So we define the AAA authentication AVP to carry the AAA service authentication information. Similarly there is no special AVP for access token transportation. So we define an Access Token AVP to carry the access token of the JN. The content of these two AVPs is detailed in the next two paragraphs.

AAA Authentication AVP The data field of the AAA Authentication AVP contains:

- $cert_{AAA}$: the X.509v3 certificate of the AAA service
- ID_{JN}: the identity of the JN that it has sent in the first exchange
- $Sign_{AAA_i}(R_{AAA}, R_{JN}, ID_{JN})$: the piece of the AAA service signature computed by the server number i on the random numbers R_{AAA} and R_{JN} together with the identity of the JN. Its length is equal to the AAA service private key length. According to NIST recommendations 1024 bits RSA private key length is enough for most of the applications until 2010 [9]. But this maybe insufficient for communicating critical data.

These information actually form the authentication material of the AAA service that are sent in the fourth exchange of the non-optimized protocol (cf. section 3.1).

Access Token AVP The data field of the Access Token AVP contains:

- T_{JN}: the access token deadline, it specifies the expiration time of the access token as the number of seconds since midnight on 1st January 1970
- $Sign_{AAA_i}(IP_{CGA_{JN}}||T_{JN})$: the piece of the AAA service signature computed by the server number i on the concatenation of the IP address and the access token deadline of the JN. The IP address is a Cryptographically Generated Address (cf. section 4.1)

When the CN receives at least t Access Token AVPs from t AAA servers, it computes $Sign_{AAA}(IP_{CGA_{JN}}||T_{JN})$ [6] then obtains its access token: $AT_{JN} = T_{JN}||Sign_{AAA}(IP_{CGA_{JN}}||T_{JN})$ (cf. Fig. 4). This access token is like a passport for JN. Henceforward it is added to JN's traffic during the access phase, and JN's neighbors check its validity to establish if the JN was authorized or not by the AAA service. Figure 5 illustrates a JN sending a message M to a destination

Dest. This message includes AT_{JN} and has $IP_{CGA_{JN}}$ as a source IP address. IN, R_1 and R_2 are relay nodes and operate as EPs. They examine AT_{JN} before relaying M. In the reality the only node that can verify AT_{JN} is the immediate neighbor IN (cf. section 4).

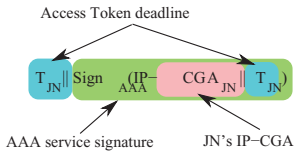


Fig. 4. Access Token content ($IP\text{-}CGA_{JN} = IP_{CGA_{JN}}$)

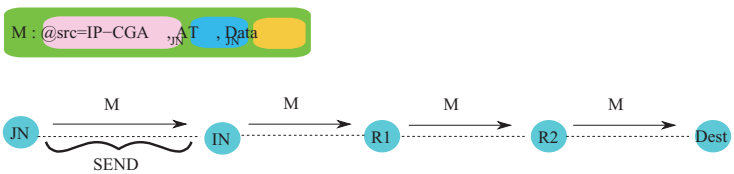


Fig. 5. Access Token Adding to JN's Messages

4 The Access Token Usage to secure the Access Phase

Unlike the solution proposed in [10], we make use of an access token to control and secure the access to the ad-hoc network. Ad-hoc nodes need not keep a list of authorized nodes in their caches. AAA servers need not broadcast a message each time a new node is authorized to access the network, either.

Authenticated and authorized nodes must rather add their access tokens to their messages in the beginning of the access phase to prove their legitimacy. You can refer to section 3 for the access token content and the way it is obtained by the JNs.

In this section we give an overview of CGA and SEND for better understanding of their role in the validity check of the access token during the access phase.

4.1 Brief Introduction to CGA and SEND

Cryptographically Generated Address (CGA) [11] strongly links the public key of a CN to its IP address. It is computed using CGA parameters that include:

– the node's public key
– a modifier: an integer that avoids collision by introducing randomness
– collision count: an integer equal to 0, 1 or 2

Once a CGA is computed by a JN, it becomes its IP address or IP-CGA. CGA parameters are necessary for the JN's neighbors to verify the CGA address. For this reason SEND transports the CGA in the IP source address and the CGA parameters in the CGA option of the same packet (cf. the paragraph below).

SEND [12] is a secure version of the Neighbor Discovery Protocol (NDP). NDP enables a JN to discover its neighboring nodes or determine if they are still reachable by mainly soliciting them to advertise their link-layer addresses. So JN sends Neighbor Solicitation messages to its immediate neighbor IN from which it receives Neighbor Advertisement messages as a reply (cf. Fig. 6).

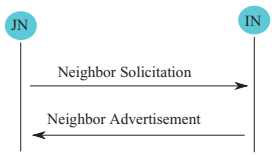


Fig. 6. Neighbor Discovery Protocol

SEND uses CGAs (cf. the previous paragraph) and includes the CGA option, the RSA option and the Timestamp and Nonce options in NDP packets to secure NDP. The CGA option contains the CGA parameters. The RSA option contains an RSA signature generated on all the other options, the packet source and destination IP-CGAs, and the packet data.

After SEND exchanges, the JN and its neighbors record the CGAs of the neighboring entities and the necessary parameters for verification in their SEND caches.

4.2 Access Token Validity Check by the Immediate Neighbors

Immediate neighbors (INs) of the JN receive JN's access token AT_{JN} (cf. Fig. 4) in the traffic of JN (cf. Fig. 5). They check the validity of AT_{JN} relying on the information recorded in their caches filled in during SEND exchanges (cf. section 4.1). The first step is to validate JN's IP-CGA after SEND exchanges by means of the CGA parameters.

The second step is to check if the IP-CGA received in the traffic is equal to that validated using above. The third step consists in concatenating the IP-CGA of the received traffic with the expiration time of validity T_{JN} in AT_{JN} and in verifying the consistency of this concatenation with the signature $Sign_{AAA}(IP_{CGA_{JN}}||T_{JN})$ in AT_{JN}. This verification can be processed thanks

to the public key of the AAA service. This public key was previously registered by the INs after their authentication. In case of successful verification, INs add AT_{JN} and T_{JN} in their caches entry for JN. Henceforward INs no longer proceed with the previous check until the expiration of T_{JN}. They simply check if the access token received is equal to that in their caches and that the deadline T_{JN} has not expired yet.

4.3 Access Token Validity Check by the Destination

In the subsection 4.2 we demonstrated that the validity check of the access token is based on SEND and CGA. Consequently only INs can do the check. It is important that remote destination nodes can do the check, too, in order to detect malicious nodes attacks like spoofing and replaying attacks. So we need a protocol like SEND that carries CGA parameters beyond the immediate neighborhood of JN. That protocol should never carry the access token without securing it by a signature, a hash or a ciphering. The two-way protocol depicted in figure 7 meets these two requirements.

In the first exchange with the destination node Dest, the JN sends its CGA parameters namely the modifier, the public key (PK) and the collision count for its IP-CGA verification. In addition to these parameters, it sends a nonce that identifies this message and the corresponding response sent later by Dest. It sends also a sequence number in order to avoid message replaying attacks.

The sequence number is equivalent to the timestamp sent in the Timestamp option of SEND packets. If a mechanism exists to correctly synchronize ad-hoc nodes, JN can send a timestamp rather than a sequence number.

g, p, and A_{JN} are Diffie Hellman parameters [13] that JN sends as well, thus asking the destination to compute a Diffie Hellman shared key K. Finally AT_{JN} is also included in the first exchange in order to prove that the JN was authenticated and authorized by the AAA service.

The RSA signature of the JN is computed on all these elements plus the source and destination IP addresses of the message, respectively $IP_{CGA_{JN}}$ and $IP_{CGA_{Dest}}$. Thanks to this signature the access token is protected from spoofing attacks. No other node can indeed send the signed CGA parameters for $IP_{CGA_{JN}}$ and AT_{JN} validation (cf. section 4.1).

Now Dest can validate AT_{JN} following the same procedure described in section 4.2. It first authenticates the JN thanks to the RSA signature and establishes, so, that it is effectively the owner of the public key received. Then the validity of $IP_{CGA_{JN}}$ and AT_{JN} can be checked.

Dest moreover computes the Diffie Hellman shared key K. Let's d (resp. j) be the secret Diffie Hellman number of Dest (resp. JN), then $A_{Dest} = g^d mod\ p$ (cf. Fig. 7) and $K = A_{JN}^d mod\ p = A_{Dest}^j mod\ p$.

If all these operations were successful, Dest responds with a similar message containing the same nonce of JN's message. JN achieves the same verifications on Dest's elements. If these operations were successful, JN and Dest have been successfully and mutually authenticated. Both have also proved that they were successfully authenticated by the AAA service and so authorized to access the

network. In addition they have established a Diffie Hellman shared key K that they can use in their later communications. Hence they no longer need to send their access tokens until one of the deadlines T_{JN} T_{Dest} expires.

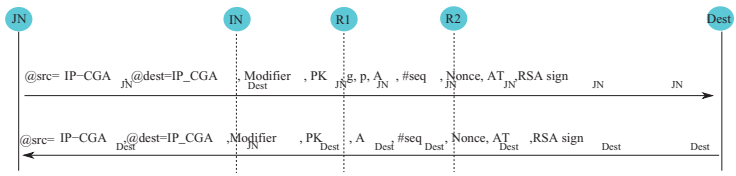


Fig. 7. Protocol for source to destination access token checking

4.4 Robustness to Classical attacks

Signing the concatenation of $IP_{CGA_{JN}}$ and T_{JN} with the AAA service's private key (cf. section 3.3) preserves the IP-CGA from spoofing. It preserves at the same time the access token from spoofing. If we place ourselves in the case where just the access token is attached to JN's traffic during the access phase (cf. Fig. 5), this preservation can unfortunately be guaranteed only in the immediate neighborhood of JN since only immediate neighbors can execute SEND (cf. section 4.1).

INs can check the validity of the access token and so detect an access token spoofing attack. However destination nodes can not detect a spoofing attack without SEND. Figure 8 illustrates an example of spoofing. The node R_1 pretends that it is relaying one of JN's messages but in fact it has generated this message itself. It has placed into it the $IP_{CGA_{JN}}$ as a source IP address and AT_{JN} as the attached access token. Dest can not receive SEND messages from JN, so there are no CGA parameters for JN in Dest's cache. Dest can't verify the validity of AT_{JN} sent by R_1 and accepts the message as if it was sent by JN. Consequently it does not detect R_1 as a malicious node.

The two-way protocol described in section 4.3 solves this problem. It transfers CGA parameters for the AT_{JN} validation and protects these elements with a RSA signature. Replay attacks are also avoided by means of this protocol because it employs sequence numbers: Dest detects that a message was replayed by comparing its sequence number to those received in the previous messages.

5 Conclusions and perspectives

In this paper we investigated an optimized version of the AAA protocol described in [2]. Furthermore we demonstrated how to secure the access phase by means of a special form of the access token associated to the use of SEND, CGA, and a two-way protocol that we conceived in the section 4.3.

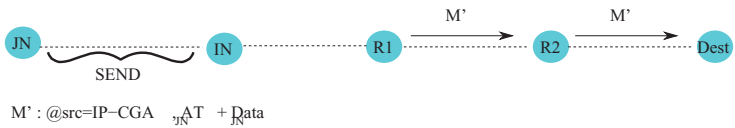


Fig. 8. Access Token Spoofing

The number of AAA exchanges and their length may be a difficult point in the implementations. We are now evaluating the overhead induced to make sure this authentication and access control protocol is usable for network access control.

Acknowledgment

We are thankful to ANR (Agence Nationale de la Recherche) for financially supporting the project TLCOM MobiSEND.

References

1. Nikolov, M.: Exploiting social and mobile ad hoc networking to achieve ubiquitous connectivity (2008) `http://developer.symbian.com/main/documentation/technologies/future_technology_ideas/milen_nikolov.jsp`.
2. Larafa, S., Maknavicius, M., Chaouchi, H.: Light and Distributed AAA Scheme for Mobile Ad-hoc Networks. First Workshop on Security of Autonomous and Spontaneous Networks, SETOP 2008, Loctudy, France (october 2008)
3. Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., Levkowetz, H.: Extensible Authentication Protocol (EAP). RFC 3748 (June 2004)
4. Shamir, A.: How to Share a Secret. Communications of the ACM (1979)
5. : ISO [9798-3] `http://www.iso.org/iso/fr/search.htm?qt=9798-3&published=on&active_tab=standards`.
6. Shoup, V.: Practical Threshold Signatures. Theory and Application of Cryptographic Techniques (2000)
7. Rigney, C., Willens, S., Rubens, A., Rubens, A.: Remote authentication dial in user service (radius). RFC 2865 (June 2000)
8. Calhoun, P., Loughney, J., Guttman, E., Zorn, G., Arkko, J.: Diameter Base Protocol. RFC 3588 (September 2003)
9. Keylength: `http://www.keylength.com/en/4`.
10. Khakpour, A., Laurent-Maknavicius, M., Chaouchi, H.: WATCHMAN: An Overlay Distributed AAA Architecture for Mobile Ad hoc Networks. The Third International Conference on Availability, Reliability and Security (ARES 2008), IEEE Computer Society, Barcelona, Spain (March 2008)
11. Aura, T.: Cryptographically Generated Addresses (CGA). RFC 3972 (March 2005)
12. Arkko, J., Kempf, J., Zill, B., Nikander, P.: SEcure Neighbor Discovery (SEND). RFC 3971 (March 2005)
13. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Transactions on Information Theory 22, pages 644 to 654 (1976) `http://www.rsa.com/rsalabs/node.asp?id=2248`.