Secure Negotiation for Manual Authentication Protocols

Milica Milutinovic[1], Roel Peeters[2], and Bart De Decker[1]

[1] K.U.Leuven, Dept. of Computer Science, DistriNet/SecAnon,
firstname.lastname@cs.kuleuven.be,
[2] K.U.Leuven, Dept. of Electrical Engineering - ESAT, COSIC,
firstname.lastname@esat.kuleuven.be,

Abstract. In this paper we propose a protocol that allows users operating mobile wireless personal devices to agree on a manual authentication protocol to use in a fair and secure way in order to bootstrap secure communication. Our protocol also has adjustable level of security and a variant of it is applicable to low-end devices with constrained user interfaces.

Keywords: Ad hoc Association, Fair Negotiation, Manual Authentication

1 Introduction

Mobile ad hoc networks, which are formed by spontaneous peer-to-peer associations between devices with no prior context, are brought into the spotlight as they provide an efficient solution for communication of mobile devices. However, the flexibility of ad hoc networks introduces a new dimension in securing this communication process.

Since the associations are often established in a constrained environment, the existence of a Trusted Third Party (TTP), a Public Key Infrastructure (PKI) or on-line Certificate Authorities (CAs) cannot be assumed. In addition, the association is spontaneous, which means that devices do not have a prior security context. Therefore, the use of traditional means for establishing secure communication is very limited. A possible alternative is the deployment of an authenticated Out-Of-Band (OOB) channel for key exchange or establishment. The OOB channel is considered to have a low bandwidth and is used for transferring only a small amount of authenticated data.

User involvement is unavoidable in establishing such an auxiliary channel. Protocols in which a user is considered to be an authenticated channel are denoted as manual authentication protocols. These OOB channels are established using human senses, such as sight, hearing or touch. By employing the user-controlled channel, one can bootstrap secure communication between devices, assuming that the security can be reduced to authenticated key agreement. The OOB channel is therefore used to authenticate the data exchanged over the main, insecure channel or to establish an authenticated shared secret. During the last

decade, many manual authentication techniques have been proposed, differing in the type of user involvement and required devices' interfaces.

To the best of our knowledge, this is a first work that focuses on secure and fair negotiation between previously unfamiliar devices for which manual authentication protocol to use. Since devices can differ in capabilities and available interfaces, it is necessary for them to agree on a manual authentication method prior to the association. In addition, users operating the associating devices can have different preferences regarding the type of their involvement and the desired level of security. Therefore, it is necessary to ensure fairness, i.e. that the preferences of both users are equally valued. Secure negotiation is also important, as an adversary could perform a MITM attack in order to persuade devices in agreeing on a less secure authentication method or a method that uses a specific medium as an OOB channel, which he has the ability to jam. This paper seeks to address these issues. It describes a protocol that allows associating devices to agree on a manual authentication method in a fair and secure way. An evaluation of the claimed properties is also provided.

2 Manual authentication protocols

The communication between personal mobile devices is usually performed over the wireless link since many devices are already equipped with appropriate transceivers and no additional equipment is required. However, wireless communication is inherently vulnerable to attacks. Since the receiving device cannot be assured of the sender's identity, a Man-In-The-Middle (MITM) attack a serious security threat.

In order to protect from eavesdropping and prevent an attacker from obtaining sensitive information exchanged over the wireless link, devices can exchange encrypted messages. Even though this ensures confidentiality of the exchanged data, security reduces to secure key establishment. There is no straightforward way to establish or exchange keys in a secure way, since devices cannot be assumed to have a prior trust relationship or share a common point of trust, such as a Trusted Third Party (TTP), a Public Key Infrastructure (PKI) or a Certificate Authority (CA).

In order to set up secure key establishment, we have to consider MITM attacks, where the attacker does not only obtain sensitive data exchanged between the legitimate parties, but can also modify them unobserved. For example, let us assume that devices are using each other's public keys for data encryption. Recalling that there are no TTPs and that the devices don't have a prior context, it means that they cannot check whether or not they hold a legitimate public key. Furthermore, the alternative of establishing shared symmetric keys (i.e. Diffie-Hellman key exchange protocol [3]) is also prone to attacks, as the adversary can intercept the exchanged messages in such a way that it establishes a secret key with each of the devices. Since the attack can be unobservable by the legitimate communicating partners, complete communication can be performed over the adversary who can read, modify, create or drop messages and still remain undetected.

In order to tackle these problems, an auxiliary low-bandwidth channel is used for exchanging authenticated data. It is intended for authenticated exchange of key derivation parameters, public keys or secret data that is subsequently used for key authentication. However, the inherent properties of the ad hoc mobile networks require user involvement for establishing such a channel. The idea is to make use of human sensory capabilities as a means of authenticating transferred data. Therefore, those protocols are noted as manual authentication protocols.

This idea was put forward by Stajano and Anderson [16]. They investigated secure association of devices with low computational capabilities with trusted devices, using physical contact, as a location-limited channel, for exchange of secret keys. Authenticity of data is provided as there is no ambiguity about which devices are associating.

The first manual authentication protocol was proposed by Balfanz et al. [1]. In their protocol, devices exchange their public keys over the insecure wireless channel and hashes of those public keys over an authentic OOB channel, in order to bootstrap secure communication. Devices would check the authenticity of the received keys by checking whether the received hash values correspond to the received keys. With this approach, the wireless channel can be considered to be under complete control of an adversary and only messages sent over the OOB channel would need to be authentic. Their suggested candidates for location-limited, OOB channels are contact, infrared signals and sound. Later approaches are based on these ideas.

2.1 Examples of manual authentication protocols

Strings Some of the early approaches introduced protocols based on short numerical strings. Examples are the four MANA protocols [4] and SAS-based protocols [17], [7]. Authentication of devices is achieved by manually transferring short data strings from one device to the other, entering strings on both devices or by manually comparing strings that are output of the two devices.

Images Values such as public keys or their hashes are coded into images compared by the user [10], or barcodes which devices with cameras can capture and verify [9]. Since these protocols require devices to have high resolution displays, other approaches employ simple LEDs for emitting visual patterns to be compared [12].

Audio Audio is considered to be a pervasive interface and is therefore a good candidate for associating low end devices [14], [5], [11]. Authenticity of the channel lies in the fact that users can easily verify the source of the signal.

Location These protocols rely on using RF signals or ultrasound for measuring the distance of the communicating partner and therefore provide assurance that the communication is carried out between legitimate devices [6], [2], [13].

Movement Some recent approaches introduce movement imposed by the user as a means to create a shared secret. Imposing movement on associating devices by shaking them together or simultaneous button presses in order to create shared secret data was discussed in [8] and [15].

It is obvious that different techniques require different device interfaces and user interaction. In addition, they provide different security levels. Some of the

techniques are also not applicable to all user groups such as users with impaired hearing or vision. The choice of the appropriate method depends on the context and the users controlling the devices at the time of association.

3 The negotiation protocol

Major differences between the auxiliary channels introduced in the previous section are in terms of required device interfaces, ranging from widespread interfaces like a single button or audio to very uncommon ultrasonic transceivers. Therefore, devices need to agree on a manual authentication method to use in order to bootstrap secure communication. In this section we present a protocol that provides secure and fair negotiation on which manual authentication protocol to use.

When two devices want to associate securely they have to perform a pairing procedure which typically consists of three phases. In the first, discovery phase, devices exchange their identifiers. The second phase comprises of a pairing protocol where the devices establish keys that will be used to secure subsequent communication. In the final phase, the entities that exchanged messages in the pairing protocol are authenticated to each other. In mobile ad hoc networks, the devices authenticate each other using a manual authentication protocol. We propose an extension of the procedure above: an additional phase, executed before the authentication phase, in which the devices run the negotiation protocol.

The negotiation protocol consists of several phases. In the first phase the user is prompted to create a list of preferred manual authentication methods. In the second phase both devices commit to these lists and subsequently reveal them. In the final phase the manual authentication method is selected and the users can verify the correctness of the protocol. An overview of the protocol is given in Fig. 1. In the next section, we evaluate the fairness and security properties of the proposed protocol. We will now discuss each phase in more detail.

3.1 List creation

Each device prompts the user what actions he is willing to perform. Examples of the offered options are: 'Compare Strings', 'Transfer Strings', 'Compare Audio Sequences' or 'Align Devices'. The user grades the actions he is willing to perform, according to his preferences. This will result in an ordered list of possible manual authentication methods ($list_{\mathrm{A}}$ and $list_{\mathrm{B}}$, respectively). If a device has only limited capabilities and no display, it is assumed that it has a predetermined list of possible methods, as the user cannot be prompted about the desired methods.

3.2 List exchange

Each device will first commit to its ordered list. After receiving the commitment from the other device, it will reveal its list. We will now discuss these two steps.

Committing to these ordered lists prevents the party that receives the list first from adjusting its own according to the other party's preferences. The ordered

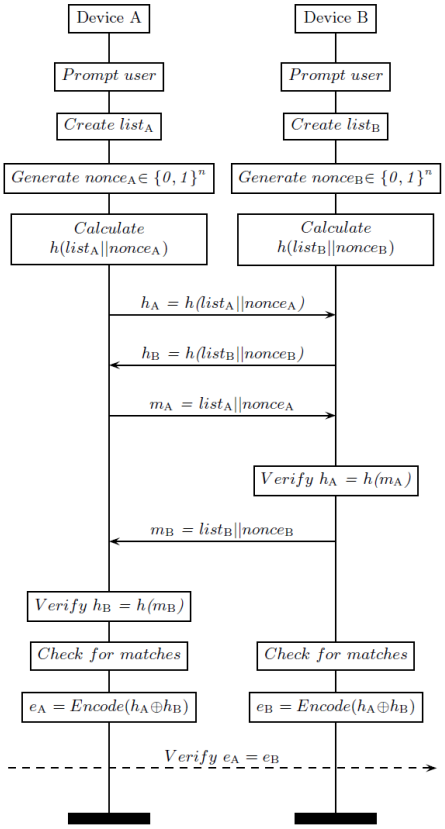

Fig. 1. Negotiation protocol

lists are appended with a random nonce ($nonce_A$ and $nonce_B$, respectively). These concatenated values are hashed using a hash function h and devices exchange the hash values, thereby committing to the lists without revealing them. Since the number of possible ordered lists is limited, appending a nonce to the list is important. This will prevent the other party from deriving the actual list from its hash value by creating look-up tables.

Following the exchange of commitments, each device sends its list and nonce in clear. After receiving the list and the nonce from the other device, they check whether the received bitstrings indeed correspond to the received commitments. If this check fails, the device will abort the protocol and alert the user.

3.3 Method selection

For each common method on the two lists, the grades are aggregated and the method with the greatest sum is selected. If there are multiple such methods,

the protocol providing the best security-usability properties is chosen. Therefore, there should be a universal agreement on the prioritization of methods.

Additionally, the user can verify the correctness of the protocol, as the complete exchange of messages is performed over the wireless and insecure channel. For this purpose, devices calculate a bitwise sum of the received and generated commitment. The resulting bitstring is coded for the users to compare (e_{A} and e_{B}, respectively).

If one of the devices does not have the means to perform the verification due to a lack of a interfaces, this step is not performed. In this case, the benefit of the protocol lies in the fact that an adversary cannot guess the chosen methods before committing to his own, reducing his success probability.

As an additional security measure, a device that received a list with only one method from the other device, will alert the user and reveal the list. This will discourage cheating parties from selecting only their most preferred method in order to persuade the communicating party to agree on it. If one method was indeed the only possibility, it will be confirmed by the sending user.

4 Evaluation

In this section we will discuss and evaluate the fairness and security properties of the proposed negotiation protocol.

4.1 Fairness

We define fairness as the property that the preferences of all parties are equally respected. Informally speaking, this makes it impossible for one party to persuade the other in accepting the authentication method it prefers. For example, to persuade the other device to accept its preferred method, the device could state that this is the only possible method. Since its list is revealed, this option can be ruled out as a possibility.

Another option is to create a list of methods that would have only one match with the received list. In order to do so, the device would have to guess the list of the other device before creating its own. This is prevented by having the devices first commit to their lists before revealing them. As already discussed before, appending a nonce to the list will prevent a cheating party from creating a look-up table for the limited number of possible lists. Since the size of the random nonce, n, is chosen to be sufficiently large, the brute force search is considered to be infeasible in this case.

Another important parameter is the length of the hash function output. For an h-bit hash value, the collision probability (two different inputs having the same hash value) is $2^{-h/2}$. In this protocol, if a cheating party manages to find two lists that would hash to the same value, it would be able to choose which one to reveal after receiving the other party's choices. Therefore, the size of the hash value should be sufficiently large, so that a cheating party would have a negligible probability of finding a collision.

4.2 Security

We define security against an adversary that succeeds in having the two devices successfully complete the protocol on different inputs. More specifically, for $list_{\mathrm{A}}$ the list of preferences sent by device A and $list'_{\mathrm{A}}$ the list of preferences of device A as received by device B (and vice versa), the probability that devices A and B complete the protocol successfully for $list_{\mathrm{A}} \neq list'_{\mathrm{A}}$ or $list_{\mathrm{B}} \neq list'_{\mathrm{B}}$ should be negligible. Informally speaking, the protocol should be secure against Man-In-The-Middle (MITM) attacks.

There are two motives for a MITM attack. Firstly, the adversary can try to persuade the devices to agree on a less secure authentication method, which would increase his probability of a successful attack. Secondly, if he has the means to jam one type of OOB channel, he would persuade the parties to agree on the authentication method using that specific channel. That would give him an advantage to perform a Denial of Service (DoS) attack and disable the authentication.

Assuming that the main wireless channel is under complete control of an adversary, we can evaluate the probability of a successful attack. We will describe the scenario where the attacker replaces the ordered lists of device A by its own list. Since he can modify exchanged messages unobserved, instead of forwarding h_{A} to device B, he sends hash value of his own list and a nonce, h'_{A}. He then receives h_{B} from device B and forwards h'_{B} to device A. If the users compare the bitwise sums of hash values, $h'_{\mathrm{B}} = h_{\mathrm{B}} \oplus h_{\mathrm{A}} \oplus h'_{\mathrm{A}}$ needs to hold. This means that the attacker needs to find a message m'_{B} such that it hashes to h'_{B}. This means that he needs to find a pre-image for the hash function. For an h-bit hash value, the probability of finding a pre-image is 2^{-h}.

Finally, an an important feature of the protocol is the fact that every protocol execution is acknowledged by the user, making it impossible for an adversary to unobservably start multiple protocol instances in order to learn user's preferences and succeed in the attack.

5 Conclusion and future work

In this paper, we proposed and analysed a fair and secure protocol for negotiating a manual authentication method as part of bootstrapping secure communication in mobile ad hoc networks. The proposed protocol does not have any special interface requirements and provides adjustable levels of security and usability. Users are allowed to choose the size of the encoded verification value to compare, thereby choosing the desired security level.

For the future work, a possible improvement of the protocol could provide more flexibility by allowing the devices to download an XML description of the authentication protocols, specifying the required interfaces and the security level they provide, from a server and update it regularly. Each device would adjust the list of actions offered to the user according to this file and the interface/hardware capabilities. Thereby, the devices would not need to agree on a predetermined logic by which authentication methods are chosen if there is a collision of grades.

References

1. D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *NDSS*, 2002.
2. M. Cagalj, S. Capkun, and J.-P. Hubaux. Key agreement in peer-to-peer wireless networks. In *Proceedings of the IEEE Special Issue on Security and Cryptography*, volume 94, 2006.
3. W. Diffie and M. E. Hellman. New directions in cryptography. In *IEEE Transactions on Information Theory*, volume 22, pages 644–654, November 1976.
4. C. Gehrmann, C. J. Mitchell, and K. Nyberg. Manual authentication for wireless devices. *RSA Cryptobytes*, 7(1):29–37, Spring 2004.
5. M. T. Goodrich, M. Sirivianos, J. Solis, C. Soriente, G. Tsudik, and E. Uzun. Using audio in secure device pairing. *Int. J. Secur. Netw.*, 4:57–68, February 2009.
6. T. Kindberg and K. Zhang. Validating and securing spontaneous associations between wireless devices. In *Information Security*, volume 2851 of *Lecture Notes in Computer Science*, pages 44–53. Springer Berlin / Heidelberg.
7. S. Laur and K. Nyberg. Efficient mutual data authentication using manually authenticated strings. In *Cryptology and Network Security*, volume 4301 of *Lecture Notes in Computer Science*, pages 90–107. Springer Berlin / Heidelberg.
8. R. Mayrhofer and H. Gellersen. Shake well before use: Authentication based on accelerometer data. In *In Pervasive*, pages 144–161. Springer, 2007.
9. J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. *Security and Privacy, IEEE Symposium on*, 0:110–124, 2005.
10. A. Perrig and D. Song. Hash visualization: a new technique to improve real-world security. In *In International Workshop on Cryptographic Techniques and E-Commerce*, pages 131–138, 1999.
11. R. Prasad and N. Saxena. Efficient device pairing using human-comparable synchronized audiovisual patterns. In *Applied Cryptography and Network Security*, volume 5037 of *Lecture Notes in Computer Science*, pages 328–345. Springer Berlin / Heidelberg.
12. N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan. Secure device pairing based on a visual channel. In *In 2006 IEEE Symposium on Security and Privacy*, pages 306–313, 2006.
13. D. Singelee and B. Preneel. Location verification using secure distance bounding protocols. In *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, November 2005.
14. C. Soriente, G. Tsudik, and E. Uzun. Hapadep: Human-assisted pure audio device pairing. In T.-C. Wu, C.-L. Lei, V. Rijmen, and D.-T. Lee, editors, *Information Security*, volume 5222 of *Lecture Notes in Computer Science*, pages 385–400. Springer Berlin / Heidelberg.
15. C. Soriente, G. Tsudik, and E. Uzun. Beda: Button-enabled device pairing. Cryptology ePrint Archive, Report 2007/246, 2007.
16. F. Stajano and R. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Security Protocols, 7th International Workshop Proceedings, Lecture Notes in Computer Science*, pages 172–194. Springer-Verlag, 1999.
17. S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. In *Advances in Cryptology CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 309–326. Springer Berlin / Heidelberg.