YET ANOTHER SECURE DISTANCE-BOUNDING PROTOCOL

Ventzislav Nikov and Marc Vauclair
NXP Semiconductors, Leuven, Belgium

Keywords: Distance-Bounding protocols, Relay Attacks, Mafia-Fraud.

Abstract: Distance-bounding protocols have been proposed by Brands and Chaum in 1993 in order to detect *relay attacks*, also known as *mafia fraud*. Although the idea has been introduced fifteen years ago, only recently distance-bounding protocols attracted the attention of the researchers.

In this paper, a new secure distance-bounding protocol is presented. It is self-contained and composable with other protocols for example for authentication or key-negotiation. It allows periodically execution and achieves better use of the communication channels by exchanging authenticated nonces. The proposed protocol becomes suitable for wider class of devices, since the resource requirements to the prover are relaxed.

1 INTRODUCTION

Consider a model in which a party known as *verifier V* is interested in learning the proximity to a second party known as *prover P*. The prover can be a trusted device (e.g. the trust can be assured by the tamper-resistance of the device) or un-trusted. In both cases, the prover is surrounded by an un-trusted environment. Many practical situations motivate this model, e.g. RFID, content protection systems (digital rights management systems), ad-hoc wireless networks, sensor networks etc., see (Anderson, 2001; L. Bussard, 2005; S. Brands, 1993; S. Capkun, 2006; Desmedt, 1988; G. Hancke, 2005; J. Reid, 2007; N. Sastry, 2003; D. Singelee, 2005; B. Waters, 2003). To motivate our research we will describe some of them.

Almost all existing RFID authentication schemes (tag/reader) are vulnerable to mafia attacks, because of their inability to estimate the distance to the tag. Such attacks are usually identified by the signal strength but a resourceful adversary can easily thwarts this. Several positioning and distance measuring techniques for ad-hoc networks or wireless location-based access control has been proposed, but most of them consider just non-adversarial settings and rely on signal strength or (signal) noise analysis.

Location-based access control is based on the rule **bigger distance implies distrust**, i.e. a device which refuses to respond to the distance estimation request or which appears to be not close enough is simply denied access. So we assume that the goal of a malicious device is to be localized in a place other than its true location so it participates in the protocol but tries to mislead the verifiers. For example in DRM the content owner (known as source) can refuse to deploy it to the sink (the receiver) if he is too far from the source.

Distance-bounding protocols provide secure proximity control, i.e. they prevent the so-called *distance fraud* attacks. Brands and Chaum (S. Brands, 1993) were the first who proposed a secure solution for this problem. The *distance-bounding* protocols measure the delays between the sending out of a challenge and the reception of the response. To be feasible, this approach requires nearly no computation during each challenge-response operation. Two types of attacks against such secure protocols are considered in the literature: *mafia fraud* or *mafia attack* (Desmedt, 1988) (or *Mig-in-the-middle* (Anderson, 2001)) and *terrorist fraud* (Desmedt, 1988). In the mafia fraud attack, the attacker does not perform any cryptographic operations based on the security protocol, and only forwards the challenges and the responses between the honest prover and the honest verifier. While in the terrorist fraud attack the prover is not honest and he collaborates with the attacker.

In this paper, we present an efficient secure distance-bounding protocol suitable for wide range of devices. Note that all protocols measure the round-trip time which is twice the one-way propagation time plus the processing delay at the prover. But for heterogeneous systems the processing delay can vary a lot, e.g. an RFID tag versus a PC. Our contribution can be summarized as follows:

Our distance-bounding protocol is self-contained and can be combined with other protocols (e.g. au-

Nikov V. and Vauclair M. (2008).
YET ANOTHER SECURE DISTANCE-BOUNDING PROTOCOL.
In *Proceedings of the International Conference on Security and Cryptography*, pages 218-221
DOI: 10.5220/0001926402180221

thentication protocols), i.e. it can be plugged into any protocol provided the assumptions are satisfied. Recall that Brands and Chaum introduced the rapid bit exchange techniques in order to measure the round-trip time. Most of the known protocols use the idea of rapid bit exchange. But this approach has a drawback as pointed out in (D. Singelee, 2005) that most of the today used communication channels have a bandwidth much bigger than a bit. Having this in mind we extend the approach of Waters and Felten, so that the authenticated nonces are sent out and hence the overall efficiency of the protocol is improved. A combination of symmetric cryptographic algorithms with preprocessing reduces the resource requirements for the participants especially for the prover. In addition, our protocol does not require the prover to generate any random data which diminishes the requirements to the devices and hence makes the protocol suitable for wide range of devices. Because we measure the proximity to a mobile device the proposed protocol allows periodical execution.

The paper is organized as follows: Section 2 describes some (due to lack of space) of the known distance-bounding protocols. A new distance-bounding protocol is proposed in Section 3. We conclude in Section 4.

2 RELATED WORK

For all protocols described in this section a security parameter k is chosen, the security levels for the protocols can be described as a function of k. All protocols have a phase of “rapid exchange”, i.e. the verifier starts its timer and sends a challenge to which the prover replies, upon receiving the reply the verifier stops its timer. In most of the protocols the challenge and the reply are bits and then this rapid exchange is repeated k times. There are also few examples in which the challenge and the reply are strings of length derived from k and the rapid exchange is executed just ones.

Brands and Chaum (S. Brands, 1993) are the pioneers of distance-bounding protocols, they have designed several protocols secure against mafia fraud: At the beginning the prover and the verifier randomly generate $m_i \in_R \{0,1\}$ and $\alpha_i \in_R \{0,1\}$ for $i = 1,\ldots,k$. The prover also commits to $m_1|\ldots|m_k$. After this the phase of k rapid bit exchanges starts: the prover replies with $\beta_i = \alpha_i \oplus m_i$ to the received α_i. The prover signs α_i and β_i, but he also sends the opening of the commitment to $m_1|\ldots|m_k$. The verifier verifies the received signature and using the opened commitment checks whether $\beta_i = \alpha_i \oplus m_i$ for $i = 1,\ldots,k$.

Waters and Felten (B. Waters, 2003) have designed the following proximity-proving protocol: The prover randomly generates *start* and *reply* and sends $Enc(K_V^{enc}; start, reply, Sign(K_P^{sign}; ID))$ to the verifier. The verifier decrypts and extracts the nonces then checks the signature. Next he generates a random *echo* and the rapid exchange starts - he sends $(start, echo)$ to the prover, after verifying that the first part is the nonce *start* the prover replies with $(reply, echo)$. The verifier verifies that the received message consists of two parts: *reply* and *echo*.

Due to the lack of space we will point just to one more related work. As it has been pointed out by Singelee and Preneel (D. Singelee, 2005), all known protocols except Waters, Felten and Capkun, Hubaux are using the idea of Brands and Chaum to measure the proximity by a rapid bit exchange. In order to measure the round trip time with accuracy special hardware is required. Moreover most of the today used communication channels have a bandwidth much bigger than a bit. Another observation made in (D. Singelee, 2005) is that any protocol secure against mafia attacks can be made secure against terrorist attacks when *trusted hardware* is used.

3 THE NEW PROTOCOL

Design Principles. There should be as few resource demands as possible on both parties but especially on the prover. Since our real goal is to enable proximity control for a large class of devices we would like to limit the computation power and hardware resources necessary to participate in such protocol to minimum. This requirement excludes many cryptographic solutions like public-key encryption and signature, commitment schemes, zero-knowledge protocols. In addition in order to make the distance-bounding protocol applicable for a wide range of environments (e.g. from low power RFID tags and embedded devices, to PCs) additional constrains like time, energy and computations of the protocol should be taken into account.

The setup requirement should be minimal. To participate in the protocol both parties must share one or two common secrets. How these secrets are set up is out of scope for the considered protocol, for example they could be derived from an authentication protocol executed beforehand or distributed via separate protocol or build in during the production phase of the devices.

As pointed out most of the today used communication channels have a bandwidth much bigger than a bit. Moreover even when the protocol specifies that a bit is sent because of the communication packeting this bit is encapsulated to much bigger packet which is then sent over the communication medium. This observation shows that the standard cryptographic approach of rapid bit exchange is not efficient in practice. Here we don't consider the time needed for the message to pass through the communication stack.

It is desirable for the proposed protocol that all computations are performed in a preprocessing stage, i.e., when the time is measured the processing delay to be negligible e.g. it only amounts at the comparison of two numbers. The protocol should be easily composable with other protocols and because the goal is to measure the proximity to mobile devices the proposed protocol should allow secure periodical execution.

Threat Model. The protocol should be secure against both cheating verifier and mafia attacks. The reason why we don't consider terrorists attacks is twofold, first as noted by using trusted hardware this attack is prevented (as this is the case for DRM). Secondly, the only way a cryptographic proximity protocol to be secure against terrorists attacks is to force the prover to give away to the terrorists his private secrets (private-asymmetric or shared-symmetric key). In some cases this will prevent such an attack from a potential cheater. But depending on the application terrorists attacks are still a possible threat (e.g. DRM or sensor networks cases - if the device has been compromised by an attacker).

Since we would like the prover to belong to a large class of devices some of them with very limited resources, e.g. RFID, it is reasonable to assume that he either has no source of randomness or it is limited and thus insecure (predictable). Note that all existing protocols are subject to mafia attacks if the prover has insecure random number generator.

The Protocol. The purpose of the protocol is to prove to the verifier that the prover is within a given distance, without using any source of randomness. We assume that the prover P and the verifier V share some common secrets. Namely, a distance-authentication key and denoted by K and a seed by R, both with a fixed length $\tilde{k}$. These two secrets may be derived from the authentication protocol executed in advance by both parties, if the parties have already established a secure authenticated channel, or distributed via other means. Thus we separate the phase when the distance is measured from the phase when the authentication takes place. A pseudo-random function is used for the calculation of the verifier's challenge and prover's response. Since we don't consider the terrorist attacks the attacker has no access to the shared key. We will denote by $h(s;m)$ the pseudo-random function which has as inputs a secret key s and a public string m and outputs a string with a fixed length $\tilde{k}$, computationally indistinguishable from a uniformly random string. In practice, h can be HMAC or AES.

Recall that a goal when designing the protocol was to allow the periodical execution of it once a secure channel is established between the parties. In other words, the protocol can be executed several times at unspecified time intervals in order to ensure that the communicating parties are still in the same proximity. We assume that in case distance-bounding protocol fails then the secure authenticated channel is terminated.

Thus in the preprocessing stage both parties compute two sequences say $\{a_i\}$ and $\{b_i\}$ (for $i = j+1,...,j+k$), where j is a counter known for both parties (initialized to zero when the protocol is executed for the first time). For example, by using dedicated h and the seed R, i.e. $a_i = h(R;i|V)$ and $b_i = h(R;i|P)$. Note that the sequences $\{a_i\}$ and $\{b_i\}$ may also be public and not pseudo-random. For example, a recurrence relation like the Fibonacci sequence (a_i is $i+5$-th Fibonacci number and $b_i = a_i + 1$) can be used. We stress here that the sequences $\{a_i\}$ and $\{b_i\}$ should satisfy the following condition: the probabilities $Prob(a_i = b_j), Prob(b_i = b_j)$ and $Prob(a_i = a_j)$ are negligible. In case the sequences are public the seed R is either made public (thus anybody can compute $a_i = h(R;i|V)$ and $b_i = h(R;i|P)$) or the seed is not used in the computation (in the recurrence relation case). By using these sequences we avoid the need of random source for the prover. Our protocol is described below.

1. Let's assume that the prover P and verifier V share a common secret (distance-authentication key K) and another common secret (seed R) both with a fixed length $\tilde{k}$.
2. Let k be a security parameter and j be a counter known for both parties (initialized to zero when the protocol is executed for the first time). In the preprocessing stage both parties first compute fixed parts of the two sequences $\{a_i\}$ and $\{b_i\}$ (for $i = j+1,...,j+k$).
3. The second step in the preprocessing stage for both parties is to compute the tags $ma_i = h(K;a_i)$ and $mb_i = (K;b_i)$ (for $i = j+1,...,j+k$).
4. The interactive stage starts with the verifier choosing at random $i \in_R [j+1, j+k]$. Then V starts his timer and sends the tuple i, ma_i to P.
5. The prover P compares the received value ma_i with his pre-computed one and if they are the same returns the tuple i, mb_i to V.

6. Upon receiving the reply the verifier stops his timer. Then, he compares the received values i and mb_i with his pre-computed one and if the comparison is ok, computes the round-trip time.
7. Both P and V increase the counter j with k.

Now we will show that the described protocol is secure against distance and mafia frauds. In order to mount a distance fraud attack the prover must respond to the challenge ma_i in advance (i.e. before getting the challenge). Hence he should choose at random one of his possible replies mb_i for $i = j+1,...,j+k$ and send it out. The probability that prover's guess for i is correct is $1/k$ (since the verifier chooses i uniformly at random), but the repetition of the protocol will make the probability of the prover's correct guess negligible.

Consider an attacker in the maffia fraud setting. The attacker can run the distance-bounding protocol, pretending to be either prover or verifier, with a legitimate verifier or prover respectively. But then he should choose i and guess mb_i or respectively ma_i which he sends out. The probability that his guess of mb_i or ma_i is correct is negligible since these are the tags produced from a pseudo-random function and the attacker doesn't know the distance-authentication key. The probability that a random guess of mb_i or ma_i to be correct is $2^{-\tilde{k}}$, the same as guessing the distance-authentication key K. The attacker also can't use any of the previously exchanged authenticated nonces since by design the probabilities of $Prob(a_i = b_j), Prob(b_i = b_j)$ and $Prob(a_i = a_j)$ are negligible and hence the probabilities of $Prob(ma_i = mb_j), Prob(mb_i = mb_j)$ and $Prob(ma_i = ma_j)$ are negligible again because of the pseudo-randomness of the used function and the choice of the security parameter $\tilde{k}$.

We stress here that the protocol can be made secure against terrorist attacks when trusted hardware is used. Note that the trusted hardware also prevents the distance fraud. Hence from a practical point of view, the prevention from mafia fraud is more important than the prevention from the distance fraud.

4 CONCLUSIONS

This paper presents a new secure distance-bounding protocol. It is self-contained and composable with other protocols for example authentication or key-negotiation. The protocol allows periodical execution, which is in accordance with the nature of the measuring proximity to mobile devices. Better usage of the communication channels is achieved by exchanging authenticated nonces, which also improves the overall efficiency of the protocol. Since the resource requirements to the prover are relaxed the proposed protocol is suitable for wider class of devices.

ACKNOWLEDGEMENTS

This work has been done during the design and development of Display Port Content Protection system. The authors would like to thank Michael Epstein and Raymond Krasinski for the valuable discussions.

REFERENCES

Anderson, R. (2001). Security engineering: A guide to building dependable distributed systems. John Wiley and Sons.

B. Waters, E. F. (2003). Secure, private proofs of location. Princeton Computer Science TR-667-03.

D. Singelee, B. P. (2005). Location verification using secure distance bounding protocols. IEEE Computer Society pp. 834-840.

Desmedt, Y. (1988). Major security problems with "unforgeable" (feige)- at- shamir proofs of identity and how to overcome them. SecuriCom'88.

G. Hancke, M. K. (2005). An rfid distance bounding protocol. IEEE SecureComm pp. 67–73.

J. Reid, J. Neito, T. T. B. S. (2007). Detecting relay attacks with timing based protocols. ACM ASIACCS pp. 204–213.

L. Bussard, W. B. (2005). Distance-bounding proof of knowledge to avoid real-time attacks. In *IFIP/SEC*.

N. Sastry, U. Shankar, D. W. (2003). Secure verification of location claims. ACM Workshop on Wireless Security pp. 48–61.

S. Brands, D. C. (1993). Distance-bounding protocols. In *EUROCRYPT'93*. LNCS 765 pp. 344–359.

S. Capkun, J.-P. H. (2006). Secure positioning in wireless networks. IEEE Selected Areas in Communications vol.24, no.2, pp. 221-232.