Distance-Bounding Protocols: Are You Close Enough?

Christos Dimitrakakis and Aikaterini Mitrokotsa | Chalmers University of Technology

Distance-bounding protocols can offer protection against attacks on access control systems that require users to both verify their credentials and prove their location. However, tradeoffs among accuracy, cost, and privacy are necessary.

Many access control problems implicitly require access-seeking parties to not only verify their credentials but also prove their location. For instance, in an automobile's electronic lock and ignition system, a driver carries a token that activates the lock and the ignition only if the token's validity and proximity can be verified. Verifying both is necessary; otherwise, the system becomes vulnerable to man-in-the-middle attacks. Although proximity can be guaranteed through signal fading, it can be easily circumvented through active relays. For example, an adversary could simply use the Internet to relay messages between a token in a user's house to the vehicle. Many systems are susceptible to this type of attack, as shown by an extensive study of 10 car models from eight manufacturers in which the attack succeeded in all cases.[1]

In addition to car systems, this type of attack can target bank cards, mobile phones that use near-field technology, and proximity cards such as those used to access buildings. Subtler attacks on wireless ad hoc networks, which rely on location information for routing, are also possible. In this article, we address how and to what extent we can verify the distance of an authenticating party.

A token's validity can be verified with standard cryptographic authentication protocols, but the problem of establishing proximity remains. The simplest approach is to rely on signal attenuation: as distance increases, the received signal strength decreases.

In this article, we briefly explain the principles behind distance-bounding (DB) authentication protocols. Distance bounding creates inherent tradeoffs among verification accuracy, protocol cost, and location privacy. This is of greatest importance in constrained settings, because many wireless devices are limited in power or bandwidth, and distance verification imposes additional limits on computation.

Distance Bounding

DB protocols attempt to simultaneously verify the credentials and the proximity of a *prover* (the party to be authenticated). These protocols are used in settings in which adversaries might want to fool a *verifier* (the authentication system) into accepting a distant prover.

DB protocols are real-time challenge–response protocols. This means that each time i that the protocol runs, verifier V issues a challenge c_i to prover P. P then issues a response r_i, which is the result of applying a function f_x to the challenges, where x is a secret key. How can attackers defeat this protocol without knowing the secret?

A simple example is a child playing simultaneously against two chess grandmasters.[2] By simply relaying the moves of the first grandmaster P to the second grandmaster V, and vice versa, the child can win one of the games. The fraud is detected by prespecifying when each grandmaster should respond, and then checking the timing of the fraudster's moves.[2] Consequently,

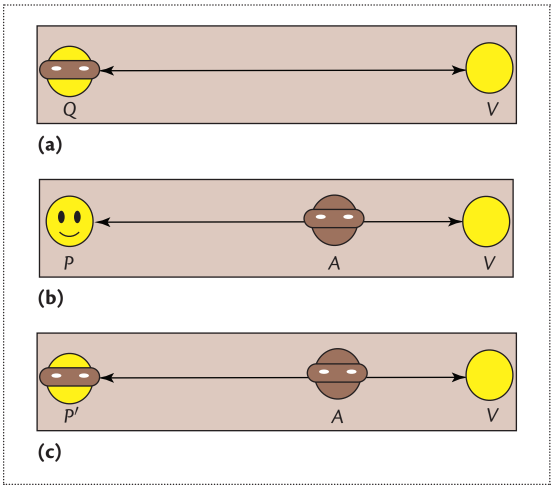

Figure 1. Types of relay attacks: (a) distance fraud, (b) mafia fraud, and (c) terrorist fraud. A masked yellow figure indicates an adversary that knows the secret *x*. A masked brown figure indicates an adversary that doesn't know the secret.

although malicious parties could simply relay the messages from V to P to be authenticated, we can apply the same principle, namely timing the fraudsters' responses. In this case, we merely want to bound their distance from the verifier; we only wish to ensure that their responses are received within a certain time limit.

However, this isn't the only type of relay attack. There are three categories, depending on the nature of the attacker (see Figure 1):

- *Distance fraud*. A legitimate prover P knows the secret x and wants to prove to V that it's close when it's actually far away.
- *Mafia fraud*. A legitimate prover P is far away from legitimate verifier V as well as an adversary A. A relays the challenges of V to P and relays P's responses back to V. Thus, both the verifier and prover think they're talking to each other. The attack succeeds if the responses are correct and V doesn't realize that P is far away.
- *Terrorist fraud*. This attack differs slightly from mafia fraud. Prover P' is a willing accomplice and collaborates with A. A great example compares A to a terrorist who wants to cross the border.[2] Prover P' helps terrorist A answer the questions of an immigration officer V. The formal restriction in this setting is that P' never reveals the secret x to A.

Distance-Bounding Protocols

Current DB protocols rely on the fact that transmission times are fundamentally bounded by the speed of light and on the difficulty of forging an authenticated message. Because the speed of light is known, measuring response delay lets us bound the distance of a verifier. The responses r_i can be made hard to forge by making secrets difficult to guess.

DB protocols were first applied in the wireless communication setting to prevent relay attacks against ATM systems. In particular, Stefan Brands and David Chaum wanted to verify the distance between a legitimate ATM and a user's smart card.[3] Many DB protocols have been proposed since then, but they generally comprise three phases: initialization, distance bounding, and verification. In the first phase, the communicating parties establish a session key. During the second phase, challenges and responses are exchanged. In the final phase, the responses' timing and validity are checked. Figure 2 outlines these phases.

Initialization Phase

In most DB protocols, the prover and verifier share a common secret x. However, if the responses r_i depend only on the challenges c_i and the key x, then every time the protocol is repeated, attackers can learn the correct response for every challenge, thereby enabling a replay attack. For this reason, we generate a secret session key α for each session that's used to calculate the correct responses in the distance-bounding phase. To create a session key α, the prover and verifier both generate random values, called *nonces*, which are fed to a pseudorandom function (PRF). The session key is the output of the PRF.

Distance-Bounding Phase

The actual distance bounding occurs in this phase, which lasts n rounds. The number of rounds n is a security parameter. In the ith round, V issues challenge c_t and observes and stores response r_t as well as the time difference Δ_i between when the challenge was issued and when the response was received. The computations during this phase are extremely simple; to minimize the computational delay, the challenges and responses are single bits.

Verification Phase

V checks the correctness of the received responses r_i and calculates an upper bound on the distance of the prover P based on the response delay Δ_i of each challenge c_i. Because information can't travel faster than light, the distance between P and V is upper bounded by $C\Delta_{max}/2$, where C is the speed of light and Δ_{max} is the maximum allowed delay between sending out bit c_i and receiving bit r_i.

Finally, verifier V indicates whether prover P is authenticated ($\text{Out}_V = 1$) or not ($\text{Out}_V = 0$). If the

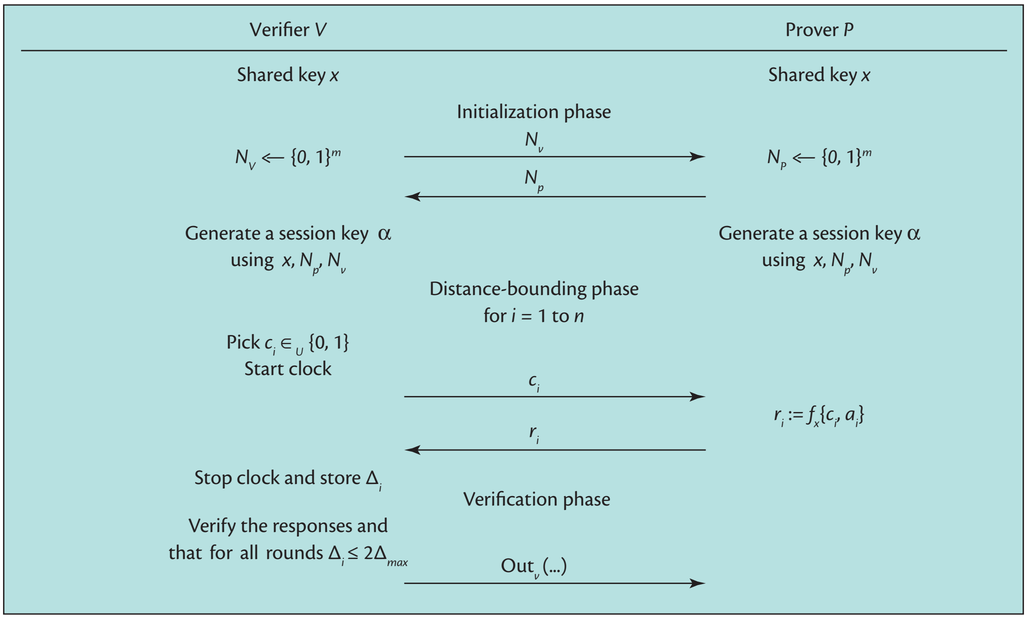
Verifier V
Prover P
Shared key x
Shared key x
Initialization phase
$N_v \leftarrow \{0,1\}^m$
N_v
$N_p \leftarrow \{0,1\}^m$
N_p
Generate a session key α using x, N_p, N_v
Generate a session key α using x, N_p, N_v
Distance-bounding phase
for $i = 1$ to n
Pick $c_i \in_U \{0,1\}$
Start clock
c_i
$r_i := f_x\{c_i, a_i\}$
r_i
Stop clock and store Δ_i
Verification phase
Verify the responses and that for all rounds $\Delta_i \leq 2\Delta_{max}$
$\mathrm{Out}_v(...)$

Figure 2. Overview of distance-bounding protocol phases: initialization, distance bounding, and verification. (The arrows denote random, uniform sampling.) In the initialization phase, the parameters of the protocol are agreed on. In the distance-bounding phase, quick messages are exchanged to guarantee that the distance is sufficiently small. In the verification phase, the messages are verified with a signature scheme.

number of incorrect responses is low and all messages are received within the allowed delay, the protocol succeeds.

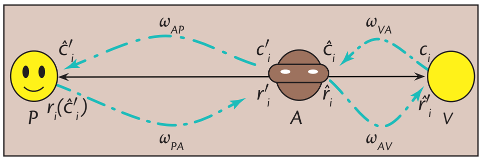

Figure 3. A mafia fraud attack under noisy conditions. Attackers communicate with a legitimate prover and a verifier through a noisy channel. Attackers first try to guess the challenges they will receive by communicating with the prover. For any correct guesses, they give the correct response; otherwise, they reply randomly. However, the channel noise makes this attack harder to carry out.

Noise and Cost Constraints

Due to timing constraints, the protocol's distance-bounding phase is performed without any error correction. As such, some of the responses might be incorrect due to noise, even though *P* knows the shared secret. (An analysis of the effect of noise can be found in "Location Leakage in Distance Bounding: Why Location Privacy Does Not Work."[4]) Thus, a tolerance threshold *t* for erroneous responses is necessary. Higher thresholds decrease the probability that honest provers will be rejected but increase the probability that adversaries will be accepted. In either case, we can improve accuracy by increasing the number of rounds *n*.

Unfortunately, each round carries a cost—especially for the prover—which is usually resource constrained owing to low power or bandwidth. We can't increase the number of rounds arbitrarily to achieve better accuracy. How should we then optimally design our protocol?

In "Expected Loss Bounds for Authentication in Constrained Channels," authors Dimitrakakis and Mitrokotsa, along with Serge Vaudenay, formalized the problem in a decision-theoretic framework.[5] An optimal solution can be found by specifying a cost not only for the transmissions but also for the two different types of authentication errors: accepting adversaries and rejecting legitimate provers. Although the problem is presented generally for authentication in noisy conditions, DB protocols are the most representative example of such authentication protocols. For this category, Dimitrakakis and his colleagues also provided a performance analysis of optimally tuned DB protocols for mafia fraud attacks.[5]

To illustrate the process, consider Figure 3, which depicts a mafia fraud attack. The communication is performed under noisy conditions; we use $\omega_{A,B}$ to denote

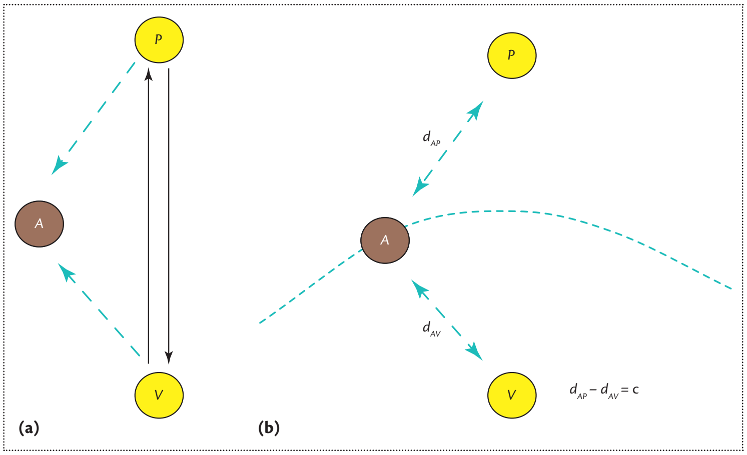

Figure 4. Information leakage regarding the locations of the prover and verifier: (a) passive eavesdropping and (b) hyperbole of the attacker's location. Potential locations for the attacker relative to the prover and verifier are constrained within the hyperbole.

the noise in the communication channel between parties A and B. We use c_i to denote the challenge sent during the ith round of the rapid bit exchange and r_i to denote the correct response. An attacker might not wait for challenge c_i but instead choose a challenge $\hat{c}_i$ and send it to legitimate prover P. The prover will finally receive a challenge $\hat{c}'_i$, which might differ from the one sent by the adversary due to the noise $\omega_{A,B}$. The same applies for the transmitted responses r_i, $\hat{r}_i$, and $\hat{r}'_i$.

Location Privacy Constraints

Rasmussen and Srđjan Čapkun noted that DB protocols might leak more information than just the fact that the prover is within the maximum allowed distance from the verifier.[6] They noticed that when a passive adversary A observes the communication between prover P and verifier V running a DB protocol, A can determine the distance between P and V by observing the arrival times of their exchanged messages. Passive adversaries might also infer their own location relative to those of P and V. Potential locations for attackers relative to the prover and verifier are constrained on a hyperbole, as Figure 4 depicts.

Although Rasmussen and Čapkun (RC) were the first to address the problem of location privacy in DB protocols, their protocol is vulnerable to mafia fraud attacks as well as collision attacks due to nonce repetition.[7,8] Nevertheless, privacy can be achieved with the location-private DB protocol proposed in "Mafia Fraud Attack against the RC Distance-Bounding Protocol," which improves on the RC protocol.[7]

Mitrokotsa and her colleagues investigated the extent to which location privacy is achievable for both *omniscient* and *limited* adversaries.[4] Omniscient adversaries can measure the transmitted messages' signal strength and observe all messages' sending and arrival times. Limited adversaries are aware only of the time at which they receive messages from other participants.

Location privacy is information-theoretically impossible for either type of adversary.[4] In particular, adversaries limited to passive observation might break the location privacy of any polynomial-time DB protocol given unbounded computation time. However, Mitrokotsa and her colleagues showed that carefully chosen parameters enable computational, provable location privacy in practice. They proved that location privacy against limited adversaries minimally requires that the prover and verifier running a DB protocol introduce exponential delays between receiving and sending messages and provide lower bounds for these delays.[4]

Nonce Space Size

As we mentioned, DB protocols use nonces to avoid replay attacks. However, constrained devices such as those used in RFID systems might not be able to use very large nonce spaces. Consequently, collisions might be relatively frequent, enabling replay attacks. In fact, in many DB protocols, attackers can recover the shared secret between a prover and a verifier after some protocol repetitions. More precisely, Mitrokotsa and her colleagues showed that the number of repetitions required is only logarithmic in key length and mainly depends on the length of the nonce.[9] Consequently, nonce space size is extremely important for these protocols. Nevertheless, it's possible to calculate experimental and theoretical bounds that practitioners can use to select appropriate security parameters, including nonce and secret key length, depending on the application scenario. In any case, there exists a countermeasure that can reduce the probability of a successful attack in this setting.

Provable Security of DB Protocols

An important question regarding DB protocols is whether their security can be proved formally. Unfortunately, only a few existing protocols are provably secure in a formal sense. Ioana Boureanu and her colleagues were the first to propose a family of provably secure DB protocols called SKI.[9,10] They've shown that SKI and its variants are provably secure against distance fraud and generalized versions of mafia and terrorist fraud, even under noisy conditions. SKI is designed to be resistant to cases in which the shared secret x is reused outside the computation of the session key. Marc Fischlin and Cristina Onete proposed another provably secure DB protocol in "Subtle Kinks in Distance-Bounding: An Analysis of Prominent Protocols."[11]

Vaudenay performed a thorough comparison of the two protocols:[12] The main advantages of the Fischlin-Onete DB protocol are its employment of binary challenges, its resilience to noise at a level of 1/4, and its reliance on standard PRF security.[11] Its main drawbacks include its low resistance to mafia fraud, $B(n, \tau, 3/4)$, and nonuniform security for distance fraud. The SKI protocol's main advantages include its uniform security for distance fraud and its better resistance, $B(n, \tau, 2/3)$, against man-in-the-middle attacks—that is, a generalization of mafia fraud attacks. SKI's main drawbacks include its use of nonbinary challenges, its resilience to noise at a level of 16, and its reliance on nonstandard PRF security.

Distance-bounding protocols are a necessary countermeasure to the emerging threat of relay attacks, especially with the advent of RFID and near-field technology in credit cards and mobile phones. They're also important for applications such as wireless ad hoc networks, where network nodes must be reasonably sure of their neighboring nodes' proximity to perform network routing.

One of the greatest challenges in DB protocol implementation is that computation in the distance-bounding phase must be extremely fast. The RC protocol had a radio-frequency circuit with nanosecond latency;[13] more recently, G.P. Hancke proposed a system that achieves low-latency distance bounding through ultra-wideband pulses.[14] Because DB protocols must interact in the physical layer, dedicated hardware is necessary for practical implementation. Consequently, widespread deployment of such protocols must await manufacturer endorsement.

We believe that the time for wide adoption of DB protocols has arrived. The field is now mature, and provably secure protocols exist. Anything would be an improvement on the current state of affairs, which offers no protection against any of these attack modes. ■

References

1. A. Francillon, B. Danev, and S. Čapkun, "Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars," *Proc. 18th Ann. Network and Distributed System Security Symp.* (NDSS 11), 2011; http://s3.eurecom.fr/docs/ndss11_francillon.pdf.
2. T. Beth and Y. Desmedt, "Identification Tokens—or: Solving the Chess Grandmaster Problem," *Advances in Cryptography* (CRYPTO 90), 1990, pp. 169–177.
3. S. Brands and D. Chaum, "Distance-Bounding Protocols (Extended Abstract)," *Advances in Cryptology—EUROCRYPT 93*, LNCS 765, Springer, 1994, pp. 344–359.
4. A. Mitrokotsa, C. Onete, and S. Vaudenay, "Location Leakage in Distance Bounding: Why Location Privacy Does Not Work," *Computers and Security*, vol. 45, 2014, pp. 199–209.
5. C. Dimitrakakis, A. Mitrokotsa, and S. Vaudenay, "Expected Loss Bounds for Authentication in Constrained Channels," *Proc. IEEE INFOCOM*, 2012, pp. 478–485.
6. K. Rasmussen and S. Čapkun, "Location Privacy of Distance Bounding," *Proc. 15th ACM Conf. Computer and Comm. Security* (CCS 08), 2008, pp. 149–160.
7. A. Mitrokotsa, C. Onete, and S. Vaudenay, "Mafia Fraud Attack against the RC Distance-Bounding Protocol," *Proc. 2012 IEEE Int'l Conf. RFID-Technologies and Applications* (RFID-TA 12), 2012, pp. 74–79.
8. J.-P. Aumasson, A. Mitrokotsa, and P. Peris-Lopez, "A Note on a Privacy-Preserving Distance-Bounding Protocol," *Proc. 13th Int'l Conf. Information and Comm. Security* (ICICS 11), 2011, pp. 78–92.
9. A. Mitrokotsa et al., "On Selecting the Nonce Length in Distance-Bounding Protocols," *Computer J.*, 4 Apr. 2013; doi: 10.1093/comjnl/bxt033.
10. I. Boureanu, A. Mitrokotsa, and S. Vaudenay, "Secure and Lightweight Distance-Bounding," *Lightweight Cryptography for Security and Privacy*, LNCS 8162, Springer, 2013, pp. 97–113.
11. M. Fischlin and C. Onete, "Subtle Kinks in Distance-Bounding: An Analysis of Prominent Protocols," *Proc. 6th ACM Conf. Security and Privacy in Wireless and Mobile Networks* (WiSec 13), 2013, pp. 195–206.
12. S. Vaudenay, "On Modeling Terrorist Frauds," *Provable Security*, LNCS 8209, Springer, 2013, pp. 1–20.
13. K.B. Rasmussen and S. Čapkun, "Realization of RF Distance Bounding," *Proc. 19th USENIX Conf. Security* (USENIX Security 10), 2010, p. 25.
14. G.P. Hancke, "Design of a Secure Distance-Bounding Channel for RFID," *J. Network and Computer Applications*, vol. 34, no. 3, 2011, pp. 877–887.

Christos Dimitrakakis is a lecturer at Chalmers University of Technology. His research interests include decision theory, reinforcement learning, statistics, and security and privacy. Dimitrakakis received a PhD in machine learning from the École Polytechnique Fédérale de Lausanne, Switzerland. Contact him at chrdimi@chalmers.se.

Aikaterini Mitrokotsa is an assistant professor in the Department of Computer Science and Engineering at Chalmers University of Technology. Her research interests include information and network security, privacy preservation, machine learning for security, and applied cryptography. Mitrokotsa received a PhD in network security from the University of Piraeus, Greece. She's an associate editor of *IEEE Communications Letters* and the *Computers and Security Journal*. Contact her at aikmitr@chalmers.se.