Private Authentication

Martín Abadi

Computer Science Department
University of California at Santa Cruz[**]
abadi@cs.ucsc.edu

Abstract. Frequently, communication between two principals reveals their identities and presence to third parties. These privacy breaches can occur even if security protocols are in use; indeed, they may even be caused by security protocols. However, with some care, security protocols can provide authentication for principals that wish to communicate while protecting them from monitoring by third parties. This paper discusses the problem of private authentication and presents two protocols for private authentication of mobile principals. In particular, our protocols allow two mobile principals to communicate when they meet at a location if they wish to do so, without the danger of tracking by third parties. The protocols do not make the (dubious) assumption that the principals share a long-term secret or that they get help from an infrastructure of ubiquitous on-line authorities.

1 Privacy, authenticity, and mobility

Although privacy may coexist with communication, it often does not, and there is an intrinsic tension between them. Often, effective communication between two principals requires that they reveal their identities to each other. Still, they may wish to reveal nothing to others. Third parties should not be able to infer the identities of the two principals, and to monitor their movements and their communication patterns. For better or for worse, they often can. In particular, a mobile principal may advertise its presence at a location in order to discover and to communicate with certain other principals at the location, thus revealing its presence also to third parties.

Authentication protocols may help in addressing these privacy breaches, as follows. When a principal A wishes to communicate with a principal B, and is willing to disclose its identity and presence to B but not to other principals, A might demand that B prove its identity before revealing anything. An authentication protocol can provide this proof. It can also serve to establish a secure channel for subsequent communication between A and B.

However, authentication protocols are not an immediate solution, and they can in fact be part of the problem. Privacy is not one of the explicit goals of

[**] This work was started at Bell Labs Research, Lucent Technologies, and at InterTrust's Strategic Technologies and Architectural Research Laboratory, and is partly supported by the National Science Foundation under Grant CCR-0208800.

common authentication protocols. These protocols often send names and credentials in cleartext, allowing any eavesdropper to see them. An eavesdropper may also learn substantial information from encrypted packets, even without knowing the corresponding decryption keys; for example, the packets may contain key identifiers that link them to other packets and to certain principals. Furthermore, in the course of authentication, a principal may reveal its identity to its interlocutor before knowing the interlocutor's identity with certainty. If A and B wish to communicate but each wants to protect its identity from third parties, who should reveal and prove theirs first?

This last difficulty is more significant in peer-to-peer communication than in client-server communication, although the desire for privacy appears in both settings.

- In client-server systems, the identity of servers is seldom protected. However, the identity of clients is not too hard to protect, and this is often deemed worthwhile. For example, in the SSL protocol [14], a client can first establish an "anonymous" connection, then authenticate with the protection of this connection, communicating its identity only in encrypted form. An eavesdropper can still obtain some addressing information, but this information may be of limited value if the client resides behind a firewall and a proxy. (Similarly, the Skeme protocol [19] provides support for protecting the identity of the initiator A of a protocol session, but not the identity of the interlocutor B.)
- The symmetry of peer-to-peer communication makes it less plausible that one of the parties in an exchange would be willing to volunteer its identity first. Privacy may nevertheless be attractive. In particular, mobile principals may want to communicate with nearby peers without allowing others to monitor them (cf. Bluetooth [7] and its weaknesses [18]). Thus, privacy seems more problematic and potentially more interesting in the fluid setting of mobile, peer-to-peer communication.

This paper gives a definition of a privacy property (informally). This property implies that each principal may reveal and prove its identity to certain other principals, and hide it from the rest. The definition applies even if all parties are peers and have such privacy requirements.

Standard authentication protocols do not satisfy the privacy property. However, we show two protocols that do, and undoubtedly there are others (to the extent that informally described protocols can satisfy informally defined properties). In our protocols, a session between two principals A and B consists of messages encrypted under public keys and under session keys in such a way that only A and B discover each other's identity. The protocols differ from standard protocols by the absence of cleartext identity information. More subtly, they rely on some mild but non-trivial assumptions on the underlying cryptographic primitives. One of the protocols also includes a subtle "decoy" message in order to thwart certain active attacks.

Our protocols do not assume that the principals A and B have a long-term shared secret. Neither do they require an infrastructure of on-line trusted third parties, or suppose that the world is organized into domains and that each principal has a home domain. In this respect, the protocols contrast with previous ones for related purposes (see for example [4, 6, 23, 30] and section 5). Because of their weak infrastructure needs, the protocols are consistent with ad hoc networking.

As an example, consider a mobile principal A that communicates with others when they are in the same (physical or virtual) location. In order to establish connections, A might constantly broadcast "hello, I am A, does anyone want to talk?". An eavesdropper could then detect A's presence at a particular location. An eavesdropper could even monitor A's movements without much difficulty, given sensors at sufficiently many locations. Our protocols are applicable in this scenario, and are in fact designed with this scenario in mind. Suppose that two principals A and B arrive anonymously at a location. Although A and B may know of each other in advance, they need not have a long-term shared key. Furthermore, neither may be certain a priori that the other one is present at this location. If they wish to communicate with one another, our protocols will enable them to do it, without the danger of being monitored by others.

The next section defines and discusses the privacy property sketched above. Section 3 presents the assumptions on which our protocols rely. Section 4 develops the two protocols and some optimizations and extensions. Section 5 discusses some related problems and related work (including, in particular, work on message untraceability). Section 6 concludes.

This paper does not include a formal analysis for the protocols presented. However, formalizing the protocols is mostly a routine exercise (for example, using the spi calculus [1] or the inductive method [25]). Reasoning about their authenticity and secrecy properties, although harder, is also fairly routine by now. More challenging is defining a compelling formal specification of the privacy property. Such a specification should account for any "out-of-band" knowledge of attackers, of the kind discussed in section 3. In this respect, placing private authentication in the concrete context of a system may be helpful. We regard these as interesting subjects for further work. Recently, several researchers who read drafts of this paper (Vitaly Shmatikov and Dominic Hughes, Hubert Comon and Véronique Cortier, and Cédric Fournet) have made progress on these subjects. Their ideas should be applicable to other systems with privacy goals, beyond the protocols of this paper.

2 The problem

More specifically, suppose that a principal A is willing to engage in communication with some set of other principals S_A (which may change over time), and that A is willing to reveal and even prove its identity to these principals. This proof may be required, for instance if A wishes to make a sensitive request from each of these principals, or if these principals would reveal some sensitive data only to A. The problem is to enable A to authenticate to principals in S_A without requiring A to compromise its privacy by revealing its identity or S_A more broadly:

1. A should be able to prove its identity to principals in S_A, and to establish authenticated and private communication channels with them.
2. A should not have to indicate its identity (and presence) to any principal outside S_A.
3. Although an individual principal may deduce whether it is in S_A from A's willingness to communicate, A should not have to reveal anything more about S_A.

Goal 1 is common; many cryptographic protocols and security infrastructures have been designed with this goal in mind.

Goal 2 is less common. As discussed above, it is seldom met with standard protocols, but it seems attractive. When C is a principal outside S_A, this goal implies that A should not have to prove its identity to C, but it also means that A should not have to give substantial hints of its identity to C.

We could consider strengthening goal 2 by saying that A should have to reveal its identity only to principals $B \in S_A$ such that $A \in S_B$, in other words, to principals with which A can actually communicate. However, we take the view that S_B is under $B's$ control, so B could let $A \in S_B$, or pretend that this is the case, in order to learn A's identity. At any rate, this variant seems achievable, with some additional cost; it may deserve study.

Goal 3 concerns a further privacy guarantee. Like goal 2, it is somewhat unusual, seldom met with standard techniques, but attractive from a privacy perspective. It might be relaxed slightly, in particular allowing A to reveal the approximate size of S_A.

Note that A may be willing to engage in anonymous communication with some set of principals in addition to S_A. We assume that A is programmed and configured so that it does not spuriously reveal its identity (or other private data) to those other principals accidentally. This assumption is non-trivial: in actual systems, principals may well reveal and even broadcast their names unnecessarily.

3 Assumptions

This section introduces the assumptions on which our protocols rely. They generally concern communication and cryptography, and the power of the adversary in these respects. (Menezes et al. [22] give the necessary background in cryptography; we rely only on elementary concepts.) Although the assumptions may not hold in many real systems, they are realistic enough to be implementable, and advantageously simple.

3.1 Communication

We assume that messages do not automatically reveal the identity of their senders and receivers—for example, by mentioning them in headers. When the location of the sender of a message can be obtained, for example, by triangulation, this assumption implies that the location does not reveal the sender's identity. This assumption also entails some difficulties in routing messages. Techniques for message untraceability (see for example [10, 26, 27] and section 5) suggest some sophisticated solutions. Focusing on a relatively simple but important case, we envision that all messages are broadcast within some small area, such as a room or a building.

We aim to protect against an adversary that can intercept any message sent on a public channel (within the small area under consideration or elsewhere). In addition, the adversary is active: it can send any message that it can compute. Thus, the adversary is essentially the standard adversary for security protocols, as described, for example, by Needham and Schroeder [24].

We pretend that the adversary has no "out-of-band" information about the principals with which it interacts, that is, no information beyond that provided by the protocols themselves. This pretense is somewhat unrealistic, but it is a convenient simplification, as the following scenario illustrates. Suppose that three principals A, B, and C are at the same location, alone. Suppose further that A and B are willing to communicate with one another, and that A is willing to communicate with C, but B is not. Therefore, the presence of B should remain hidden from C. However, suppose that C suspects that $S_A = \{B, C\}$, or that A even tells C that $S_A = \{B, C\}$. When C sees traffic between A and someone else, C may correctly deduce that B is present. Our simplification excludes this troublesome but artificial scenario.

3.2 Cryptography

We also assume that each principal A has a public key K_A and a corresponding private key K_A^{-1}, and that the association between principals and public keys is known. This association can be implemented with the help of a mostly-off-line certification authority. In this case, some additional care is required: fetching certificates and other interactions with the certification authority should not compromise privacy goals. Alternatively, the association is trivial if we name principals by their public keys, for example as in SPKI [12]. Similarly, it is also trivial if we use ordinary principal names as public keys, with an identity-based cryptosystem [31]. Therefore, we may basically treat public keys as principal names.

When K^{-1} is a private key, we write $\{M\}_{K^{-1}}$ for M signed using K^{-1}, in such a way that M can be extracted from $\{M\}_{K^{-1}}$ and the signature verified using the corresponding public key K. As usual, we assume that signatures are unforgeable. Similarly,[1] when K is a public key, we write $\{M\}_K$ for the encryption of M using K. We expect some properties of the encryption scheme:

[1] These notations are concise and fairly memorable, but perhaps somewhat misleading. In particular, they imply that the same key pair is used for both public-key signatures and encryptions, and that the underlying algorithms are similar for both kinds of operations (as in the RSA cryptosystem). We do not need to assume these properties.

1. Only a principal that knows the corresponding private key K^{-1} should be able to understand a message encrypted under a public key K.
2. Furthermore, decrypting a message with a private key K^{-1} should succeed only if the message was encrypted under the corresponding public key K, and the success or failure of a decryption should be evident to the principal who performs it.
3. Finally, encryption should be which-key concealing [3, 5, 8], in the following sense. Someone who sees a message encrypted under a public key K should not be able to tell that it is under K without knowledge of the corresponding private key K^{-1}, even with knowledge of K or other messages under K. Similarly, someone who sees several messages encrypted under a public key K should not be able to tell that they are under the same key without knowledge of the corresponding private key K^{-1}.

Property 1 is essential and standard. Properties 2 and 3 are not entirely standard. They are not implied by standard computational specifications of encryption (e.g., [15]) but appear in formal models (e.g., [1]). Property 2 can be implemented by including some checkable redundancy in encrypted messages, without compromising secrecy properties. It is not essential, but we find it convenient, particularly for the second protocol and its enhancements. Property 3 is satisfied with standard cryptosystems based on the discrete-logarithm problem [5, 8], but it excludes implementations that tag all encryptions with key identifiers. Although the rigorous study of this property is relatively recent, it seems to be implicitly assumed in earlier work; for example, it seems to be necessary for the desired anonymity properties of the Skeme protocol [19].

4 Two protocols

This section shows two protocols that address the goals of section 2. It also discusses some variants of the protocols.

The two protocols are based on standard primitives and techniques (in particular on public-key cryptography), and resemble standard protocols. The first protocol uses digital signatures and requires that principals have loosely synchronized clocks. The second protocol uses only encryption and avoids the synchronization requirement, at the cost of an extra message. The second protocol draws attention to difficulties in achieving privacy against an active adversary.

Undoubtedly, other protocols satisfy the goals of section 2. In particular, these goals seem relatively easy to satisfy when all principals confide in on-line authentication servers. However, the existence of ubiquitous trusted servers may not be a reasonable assumption. The protocols of this section do not rely on such trusted third parties.

4.1 First protocol

In the first protocol, when a principal A wishes to talk to another principal $B \in S_A$, they proceed as follows:

- A generates fresh key material K and a timestamp T, and sends out

  $$\text{“hello”}, \{\text{“hello”}, K_A, \{K_A, K_B, K, T\}_{K_A^{-1}}\}_{K_B}$$

  The key material may simply be a session key, for subsequent communication; it may also consist of several session keys and identifiers for those keys. The signature means that the principal with public key K_A (that is, A) says that it has generated the key material K for communicating with the principal with public key K_B (that is, B) near time T. The timestamp protects against replay attacks.
- Upon receipt of any message that consists of “hello” and (apparently) a ciphertext, the recipient B decrypts the second component using its private key. If the decryption yields a key K_A and a signed statement of the form $\{K_A, K_B, K, T\}_{K_A^{-1}}$, then B extracts K_A and K, verifies the signature using K_A, and checks the timestamp T against its clock. If the plaintext is not of the expected form, or if $A \notin S_B$, then B does nothing.
- A and B may use K for encrypting subsequent messages. Each of these messages may be tagged with a key identifier, derived from K but independent of A and B. When A or B receives a tagged message, the key identifier suggests the use of K for decrypting the message.

This protocol is based on the Denning-Sacco public-key protocol and its corrected version [2, 11]. Noticeably, however, this protocol does not include any identities in cleartext. In addition, the protocol requires stronger assumptions on encryption, specifically that public-key encryption under K_B be which-key concealing. This property is needed so that A's encrypted message does not reveal the identity of its (intended) recipient B.

When A wishes to communicate with several principals $B_1, \ldots, B_n$ at the same time (for example, when A arrives at a new location), A may simply start n instances of the protocol in parallel, sending different key material to each of B_1, $\ldots$, B_n. Those of $B_1, \ldots, B_n$ who are present and willing to communicate with A will be able to do so using the key material. (Section 4.3 describes optimizations of the second protocol for this situation.)

4.2 Second protocol

In the second protocol, when a principal A wishes to talk to another principal $B \in S_A$, they proceed as follows:

- A generates a fresh, unpredictable nonce N_A, and sends out

  $$\text{“hello”}, \{\text{“hello”}, N_A, K_A\}_{K_B}$$

  (In security protocols, nonces are quantities generated for the purpose of being recent; they are typically used in challenge-response exchanges.)

- Upon receipt of any message that consists of "hello" and (apparently) a ciphertext, the recipient B tries to decrypt the second component using its private key. If the decryption succeeds, then B extracts the corresponding nonce N_A and key K_A, checks that $A \in S_B$, generates a fresh, unpredictable nonce N_B, and sends out

  $$\text{“ack”}, \{\text{“ack”}, N_A, N_B, K_B\}_{K_A}$$

  If the decryption fails, if the plaintext is not of the expected form, or if $A \notin S_B$, then B sends out a "decoy" message. This message should basically look like B's other message. In particular, it may have the form

  $$\text{“ack”}, \{N\}_K$$

  where N is a fresh nonce (with padding, as needed) and only B knows K^{-1}, or it may be indistinguishable from a message of this form.
- Upon receipt of a message that consists of "ack" and (apparently) a ciphertext, A tries to decrypt the second component using its private key. If the decryption succeeds, then A extracts the corresponding nonces N_A and N_B and key K_B, and checks that it has recently sent N_A encrypted under K_B. If the decryption or the checks fail, then A does nothing.
- Subsequently, A and B may use N_A and N_B as shared secrets. In particular, they may compute one or more session keys by concatenating and hashing the nonces. They may also derive key identifiers, much as in the first protocol.

In summary, the message flow of a successful exchange is:

$$\begin{array}{l} A \rightarrow B : \text{“hello”}, \{\text{“hello”}, N_A, K_A\}_{K_B} \\ B \rightarrow A : \text{“ack”}, \{\text{“ack”}, N_A, N_B, K_B\}_{K_A} \end{array}$$

Section 4.3 describes variants of this basic pattern, for example (as mentioned above) for the case where A wishes to communicate with n principals $B_1, \ldots, B_n$.

This protocol has some similarities with the Needham-Schroeder public-key protocol [24] and others [19, 20]. However, like the first protocol, this one does not include any identities in cleartext, and again that is not quite enough for privacy. As in the first protocol, public-key encryption should be which-key concealing so that encrypted messages do not reveal the identities of their (intended) recipients. Furthermore, the delicate use of the decoy message is important:

- B's decoy message is unfortunately necessary in order to prevent an attack where a malicious principal $C \notin S_B$ computes and sends

  $$\text{“hello”}, \{\text{“hello”}, N_C, K_A\}_{K_B}$$

  and then deduces B's presence and $A \in S_B$ by noticing a response. In order to prevent this attack, the decoy message should look to C like it has the form "ack", $\{\text{“ack”}, N_C, N_B, K_B\}_{K_A}$.

- B's response to A when $A \notin S_B$ should look as though B was someone else, lest A infer B's presence. Since B sends a decoy message when its decryption fails, it should also send one when $A \notin S_B$. For this purpose, the decoy message should look to A like one that some unknown principal would send in response to A's message.

The decoy message "ack", $\{N\}_K$ is intended to address both of these requirements.

4.3 Efficiency considerations

Both protocols can be rather inefficient in some respects. These inefficiencies are largely unavoidable consequences of the goals of private authentication.

- A generates its message and sends it before having any indication that B is present and willing to communicate. In other situations, A might have first engaged in a lightweight handshake with B, sending the names A and B and waiting for an acknowledgment. Alternatively, both A and B might have broadcast their names and their interest in communicating with nearby principals. Here, these preliminary messages are in conflict with the privacy goals, even though they do not absolutely prove the presence of A and B to an eavesdropper. Some compromises may be possible; for example, A and B may publish some bits of information about their identities if those bits are not deemed too sensitive. In addition, in the second protocol, A may precompute its message.
- Following the protocols, B may examine many messages that were encrypted under the public keys of other principals. This examination may be costly, perhaps opening the door to a denial-of-service attack against B. In other situations, A might have included the name B, the key K_B, or some identifier for K_B in clear in its message, as a hint for B. Here, again, the optimization is in conflict with the privacy goals, and some compromises may be possible.

The second protocol introduces some further inefficiencies, but those can be addressed as follows:

- In the second protocol, A may process many acknowledgments that were encrypted under the public keys of other principals. This problem can be solved through the use of a connection identifier: A can create a fresh identifier I, send it to B, and B can return I in clear as a hint that A should decrypt its message:

  $$\begin{array}{l} A \rightarrow B : \text{“hello”},\, I,\, \{\text{“hello”},\, N_A,\, K_A\}_{K_B} \\ B \rightarrow A : \text{“ack”},\, I,\, \{\text{“ack”},\, N_A,\, N_B,\, K_B\}_{K_A} \end{array}$$

  The identifier I should also appear in B's decoy message. Third parties may deduce that the messages are linked, because I is outside the encryptions, but cannot relate the messages to A and B.

- Suppose that A wishes to communicate with several principals, $B_1, \ldots, B_n$. It could initiate n instances of the protocol. However, combining the messages from all the instances can be faster. In particular, although each of $B_1, \ldots, B_n$ should receive a different nonce, they can all share a connection identifier. Moreover, when K_A is long, its public-key encryption may be implemented as a public-key encryption of a shorter symmetric key K plus an encryption of K_A using K; the key K and the latter encryption may be the same for $B_1, \ldots, B_n$. Thus, A may send:

  $$\begin{array}{l}\text{"hello"}, I, \{K_A\}_K, \{\text{"hello"}, H(K_A),\ N_{A1},\ K\}_{K_{B_1}}, \ldots, \\ \qquad\qquad\qquad\qquad \{\text{"hello"}, H(K_A),\ N_{An},\ K\}_{K_{B_n}}\end{array}$$

  where H is a one-way hash function. Most importantly, the need for decoy messages is drastically reduced. A principal that plays the role of B need not produce n true or decoy acknowledgments, but only one. Specifically, B should reply to a ciphertext encrypted under K_B, if A included one in its message, and send a decoy message otherwise. This last optimization depends on our assumption that B can recognize whether a ciphertext was produced by encryption under K_B.

With these and other improvements, both protocols are practical enough in certain systems, although they do not scale well. Suppose that principals wish to communicate with few other principals at a time, and that any one message reaches few principals, for instance because messages are broadcast within small locations; then it should be possible for principals that come into contact to establish private, authenticated connections (or fail to do so) within seconds. What is "few"? A simple calculation indicates that 10 is few, and maybe 100 is few, but 1000 is probably not few. Typically, the limiting performance factor will be public-key cryptography, rather than communications: each public-key operation takes a few milliseconds or tens of milliseconds in software on modern processors (e.g., [21]). Perhaps the development of custom cryptographic techniques (flavors of broadcast encryption) can lead to further efficiency gains.

4.4 Groups

In the problem described above, the set of principals S_A and S_B with which A and B wish to communicate, respectively, are essentially presented as sets of public keys. In variants of the problem, S_A, S_B, or both may be presented in other ways. The protocols can be extended to some situations where a principal wants to deal with others not because of their identities but because of their attributes or memberships in groups, such as "ACME printers" or "Italians". These extensions are not all completely satisfactory.

- Suppose that B is willing to communicate with any principal in a certain group, without having a full list of those principals. However, let us still assume that S_A is presented as a set of public keys. In this case, we can extend our protocols without much trouble: A can include certificates in its encrypted message to B, proving its membership in groups.

- Suppose that, instead, A wants to communicate with any principal in a certain group, and S_B is presented as a set of public keys. The roles in the protocols may be reversed to handle this case.
- However, the protocols do not address the case in which neither S_A nor S_B is presented as a set of public keys, for example when both are presented as groups. Introducing group keys may reduce this case to familiar ones, but group keys can be harder to manage and protect.

5 Related problems and related work

The questions treated here are broadly related to traffic analysis, and how to prevent it. This subject is not new, of course. In particular, work on message untraceability has dealt with the question of hiding (unlinking) the origins and destinations of messages (e.g., [10, 26, 27]). It has produced techniques that allow a principal A to send messages to a principal B in such a way that an adversary may know the identities of A and B and their locations, but not that they are communicating with one another. Those techniques address how to route a message from A to B without leaking information. In the case of cellular networks, those techniques can be adapted to hide the locations of principals [13, 28]. In contrast, here we envision that all messages are broadcast within a location, simplifying routing issues, and focus on hiding the identities of principals that meet and communicate at the location. Other interesting work on untraceability in mobile networks has addressed some important authentication problems under substantial infrastructure assumptions, for instance that each principal has a home domain and that an authentication server runs in each domain [4, 23, 30]. That work focuses on the interaction between a mobile client and an authentication server of a domain that the client visits, typically with some privacy guarantees for the former but not for the latter. In contrast, we do not rely on those infrastructure assumptions and we focus on the interaction between two mobile principals with potentially similar privacy requirements.

There has been other research on various aspects of security in systems with mobility (e.g., [9, 32, 33] in addition to [4, 6, 13, 18, 23, 30], cited above). Some of that work touches on privacy issues. In particular, the work of Jakobsson and Wetzel points out some privacy problems in Bluetooth. The protocols of this paper are designed to address such problems.

The questions treated here are also related to the delicate balance between privacy and authenticity in other contexts. This balance plays an important role in electronic cash systems (e.g., [16]). It can also appear in traditional access control. Specifically, suppose that A makes a request to B, and that A is member of a group that appears in the access control list that B consults for the request. In order to conceal its identity, A might use a ring signature [29] for the request, establishing that the request is from a member of the group without letting B discover that A produced the signature. However, it may not be obvious to A that showing its membership could help, and B may not wish to publish the access control list. Furthermore, A may not wish to show all its memberships to B.

Thus, there is a conflict between privacy and authenticity in the communication between A and B. No third parties need be involved. In contrast, we do not guarantee the privacy of A and B with respect to each other, and focus on protecting them against third parties.

Designated verifier proofs address another trade-off between confidentiality and authenticity [17]. They allow a principal A to construct a proof that will convince only a designated principal B. For instance, only B may be convinced of A's identity. Designated verifier proofs differ from the protocols of this paper in their set-up and applications (e.g., for fair exchange). Moreover, in general, they may leak information about A and B to third parties, without necessarily convincing them. Therefore, at least in general, they need not provide a solution to the problem of private authentication treated in this paper.

6 Conclusion

Security protocols can contribute to the tension between communication and privacy, but they can also help resolve it. In this paper, we construct two protocols that allow principals to authenticate with chosen interlocutors while hiding their identities from others. In particular, the protocols allow mobile principals to communicate when they meet, without being monitored by third parties. The protocols resemble standard ones, but interestingly they rely on some non-standard assumptions and messages to pursue non-standard objectives. As virtually all protocols, however, they are only meaningful in the context of larger systems. They are part of a growing suite of technical and non-technical approaches to privacy.

Acknowledgments

Markus Jakobsson and Mike Reiter provided encouragement and useful references. The access-control scenario sketched in section 5 arose in the context of SPKI [12] during discussions with Carl Ellison, Alan Kotok, and Andrew Palka in 1997. Discussions with Vitaly Shmatikov were helpful in thinking about section 4.3 (though this section does not report on Vitaly's ideas). Mike Burrows suggested an interesting variant of the second protocol with timestamps and without decoy messages. Cédric Fournet pointed out a flaw in a preliminary version of a variant of the second protocol. Hugo Krawczyk confirmed points related to the Skeme protocol. Anand Desai made available information on his unpublished work. Mary Baker suggested many improvements in the presentation of this paper.

References

1. Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 148(1):1–70, January 1999. An extended version appeared as Digital Equipment Corporation Systems Research Center report No. 149, January 1998.

2. Martín Abadi and Roger Needham. Prudent engineering practice for cryptographic protocols. *IEEE Transactions on Software Engineering*, 22(1):6–15, January 1996.
3. Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (The computational soundness of formal encryption). In *Proceedings of the First IFIP International Conference on Theoretical Computer Science*, volume 1872 of *Lecture Notes in Computer Science*, pages 3–22. Springer-Verlag, August 2000.
4. Giuseppe Ateniese, Amir Herzberg, Hugo Krawczyk, and Gene Tsudik. On traveling incognito. *Computer Networks*, 31(8):871–884, 1999.
5. Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Anonymous encryption. Unpublished manuscript, 2000.
6. V. Bharghavan and C. V. Ramamoorthy. Security issues in mobile communications. In *Proceedings of the Second International Symposium on Autonomous Decentralized Systems*, pages 19–24, 1995.
7. Specification of the Bluetooth system (core, v1.0b). On the Web at `http://www.bluetooth.com`, December 1, 1999.
8. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology—EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer-Verlag, 2001.
9. Luca Cardelli. Mobility and security. In F.L. Bauer and R. Steinbrueggen, editors, *Foundations of Secure Computation*, NATO Science Series, pages 1–37. IOS Press, 2000. Volume for the 20th International Summer School on Foundations of Secure Computation, held in Marktoberdorf, Germany (1999).
10. David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the Association for Computing Machinery*, 24(2):84–88, February 1981.
11. Dorothy E. Denning and Giovanni Maria Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24(7):533–535, August 1981.
12. C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. SPKI certificate theory. On the Web at `http://www.ietf.cnri.reston.va.us/rfc/rfc2693.txt`, September 1999.
13. Hannes Federrath, Anja Jerichow, and Andreas Pfitzmann. MIXes in mobile communication systems: Location management with privacy. In Ross J. Anderson, editor, *Information hiding: First international workshop*, volume 1174 of *Lecture Notes in Computer Science*, pages 121–135. Springer-Verlag, 1996.
14. Alan O. Freier, Philip Karlton, and Paul C. Kocher. The SSL protocol: Version 3.0. On the Web at `http://home.netscape.com/newsref/std/SSL.html`, March 1996.
15. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, April 1984.
16. Markus Jakobsson. *Privacy vs. Authenticity.* PhD thesis, University of California, San Diego, 1997.
17. Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated verifier proofs and their applications. In Ueli Maurer, editor, *Advances in Cryptology—EUROCRYPT 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 143–154. Springer-Verlag, 1996.
18. Markus Jakobsson and Susanne Wetzel. Security weaknesses in Bluetooth. In *Topics in Cryptology - CT-RSA 2001, Proceedings of the Cryptographer's Track at RSA Conference 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 176–191. Springer-Verlag, 2001.

19. Hugo Krawczyk. SKEME: A versatile secure key exchange mechanism for internet. In *Proceedings of the Internet Society Symposium on Network and Distributed Systems Security*, February 1996. Available at `http://bilbo.isu.edu/sndss/sndss96.html`.
20. Butler Lampson, Martín Abadi, Michael Burrows, and Edward Wobber. Authentication in distributed systems: Theory and practice. *ACM Transactions on Computer Systems*, 10(4):265–310, November 1992.
21. Arjen K. Lenstra and Eric R. Verheul. The XTR public key system. In Mihir Bellare, editor, *Advances in Cryptology—CRYPT0 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 1–19. Springer-Verlag, 2000.
22. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1996.
23. Refik Molva, Didier Samfat, and Gene Tsudik. Authentication of mobile users. *IEEE Network*, 8(2):26–35, March/April 1994.
24. Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999, December 1978.
25. L. C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6(1–2):85–128, 1998.
26. Andreas Pfitzmann and Michael Waidner. Networks without user observability. *Computers and Security*, 6(2):158–166, April 1987.
27. Charles Rackoff and Daniel R. Simon. Cryptographic defense against traffic analysis. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on the Theory of Computing*, pages 672–681, 1993.
28. Michael G. Reed, Paul F. Syverson, and David M. Goldschlag. Protocols using anonymous connections: Mobile applications. In B. Christianson, B. Crispo, M. Lomas, and M. Roe, editors, *Security Protocols: 5th International Workshop*, volume 1361 of *Lecture Notes in Computer Science*, pages 13–23. Springer-Verlag, 1997.
29. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology—ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer-Verlag, 2001.
30. Didier Samfat, Refik Molva, and N. Asokan. Untraceability in mobile networks. In *Proceedings of the First Annual International Conference on Mobile Computing and Networking (MobiCom 1995)*, pages 26–36, 1995.
31. Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology—CRYPTO 84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer-Verlag, 1984.
32. Alex C. Snoeren and Hari Balakrishnan. An end-to-end approach to host mobility. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 155–166, 2000.
33. Yongguang Zhang and Wenke Lee. Intrusion detection in wireless ad-hoc networks. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 275–283, 2000.