Protocol Analysis for Concrete Environments

Dieter Gollmann

TU Hamburg-Harburg,
Hamburg, Germany
diego@tu-harburg.de

Abstract. For protocol analysis, we have to capture the protocol specification, the security goals of the protocol, and the communications environment it is expected to run in. In the research literature, the emphasis is usually on verification techniques and on the modelling of security properties, while in most cases the default for the communications environment is an unstructured network totally controlled by the attacker. This paper will argue that for the analysis of the kind of protocols developed today, more specific models of the communications network are required. To support this argument, a number of recently proposed security protocols with novel features will be briefly discussed.

1 Introduction

The design of security protocols has a reputation of being 'difficult and error prone'. While this difficulty is sometimes exaggerated, there is certainly a case for a proper formal analysis of widely deployed security protocols. When analyzing a protocol we are given a description of the protocol, a set of security goals, and a model of the underlying communications system. We then check whether the protocol meets its desired goals.

The research community has discussed conventions for describing protocols, and there exist proposals for protocol description languages such as CAPSL[1]. There has been a change in conventions from defining protocols as sequences of messages exchanged, to defining protocols as collections of interacting processes. The latter approach has the advantage that all checks a protocol participant makes before moving to the next stage are made explicit.

Equally, the importance of distinguishing security properties in all their nuances is widely acknowledged and there has been much research on this topic. Current 'application level' definitions for the properties of cryptographic primitives and mechanisms can be found in [9]. A discussion of various authentication properties is given in [5]. There has also been much work in the field of cryptographic theory on teasing out subtle differences in the security properties cryptographic mechanisms might be asked to fulfil.

[1] For a CAPSL tutorial see http://www.csl.sri.com/users/millen/capsl/

R. Moreno Díaz et al. (Eds.): EUROCAST 2005, LNCS 3643, pp. 365–372, 2005.

2 The Dolev-Yao Model

In contrast, less effort has been spent on modelling the communications environment. Indeed, protocol analysis often tries to assume as little as possible about the communications system and gives all messages to the adversary for delivery. This approach is often presented as analysis in the Dolev-Yao model [4]. The model makes two independent assumptions:

- Cryptography is 'perfect'. The adversary does not try to exploit any weakness in the underlying cryptographic algorithms but only algebraic properties of cryptographic operators and interactions between protocol messages.
- The adversary can observe and manipulate all messages exchanged in a protocol run and can itself start protocol runs.

The second assumption was already stated by Needham and Schroeder [10]:

> We assume that the intruder can interpose a computer in all communication paths, and thus can alter or copy parts of messages, replay messages, or emit false material. While this may seem an extreme view, it is the only safe one when designing authentication protocols.

Analysing protocols in a setting as general as possible is, however, not necessarily a route to higher security. Protocols may make use of features of the particular environment they were designed for so showing that a protocol does not meet is goal in a more general setting is useful side-information but should not be automatically classified as an attack.

3 Agility

We can analyze protocols that should meet well established security requirements and use established security primitives. Typical examples are the protection of message confidentiality through encryption, the protection of message integrity through message authentication codes or digital signatures, and the establishment of a security association between two peer nodes. Today, these mechanisms are found in networks at the IP layer (IPsec) at the transport layer (SSL) and now also at the web services layer. When dealing with established goals and mechanisms, security goals, assumptions about the environment, and standard cryptographic primitives can be integral parts of the methodology.

As an example, the BAN logic of authentication [3] assumes that attackers are outsiders and this is reflected even in its axioms, in particular in its message meaning rule for shared secrets. This rule says that if principal A receives a message containing a secret shared with principal B, A can conclude that the message came from B. However, a dishonest principal B might pass the secret to a third party and thus potentially deceive A about the source of messages. If assumptions like this are hard-coded into the verification methodology, changes about goals, primitives, and environment would require some redesign of the methodology.

There is a second direction in protocol analysis, viz the study of protocols that should meet novel requirements. In this case, we need *agile* methodologies where specific adversaries (rather than the general Dolev-Yao adversary) and new security requirements can be defined conveniently. Note that in most cases new protocols are designed because new requirements have emerged, so that traditional security assumptions have to be adjusted. For illustration, we briefly sketch four specific scenarios, together with observations on how established security assumptions may change.

4 Mobile IPv6 Binding Updates

In mobile IP, each host has a home address (HoA) at its home network and can always be reached via this address. Moreover, the mobile node has a secure tunnel to its home network. When a mobile node moves to a new location, it might tell its correspondent node that it has moved to a new care-of-address (CoA). The correspondent node could then update its binding cache that links the home address and care-of-address of the mobile node. If the correspondent node cannot check that the binding updates it receives are factually correct, an attacker could spread misinformation about the location of other nodes (can be prevented by authenticating the origins of update requests) or could lie about its own location (authentication is of no help) as part of denial-of-service attacks that flood the victim with data the attacker had requested for itself (bombing attacks).

The binding update protocol for mobile IPv6 [2, 7] works as follows (figure 1). The mobile node starts by sending a *Home Test Init* message (HoTI) via the home network and a *Care-of Test Init* message (CoTI) directly to the correspondent. The correspondent replies to both requests independently. A *Home Test* (HoT) message containing a 64-bit home keygen token K_0 and a home nonce index i is sent to the mobile node via the mobile's home address. A *Care-of Test* (CoT) message containing a 64-bit care-of keygen token K_1 and a care-of nonce index j is sent directly to the claimed current location[2]. The mobile node uses both keygen tokens to compute a binding key

$$K_{bm} := \text{SHA1}(\text{home keygen token}||\text{care-of keygen token}),$$

and the *Binding Update* (BU) authenticated by a 96-bit MAC

$$\text{HoA}, i, j, \text{HMAC\_SHA1}(K_{bm}, \text{CoA}||\text{CN}||\text{BU})\_96.$$

This protocol does not rely on the secrecy of cryptographic keys but on *return routability*. The correspondent checks that it receives a confirmation from

[2] Nonces are used to make the protocol stateless for the correspondent. The keygen tokens are derived from a long term node key and nonces. The mobile node returns the indices in its final message allowing the correspondent node to look up the nonces and recalculate the keys.

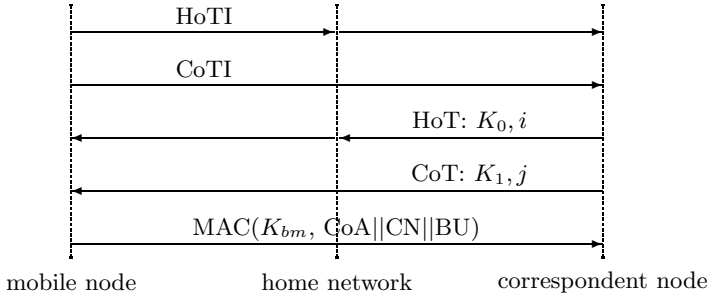

Fig. 1. Mobile IUPv6 binding update protocol

the advertised location. The threat model assumes that messages over the fixed Internet are considered secure or can be protected otherwise. Hence, keys K_0 and K_1 may be sent in the clear. These keys could also be interpreted as challenges (nonces) that bind identity to location through the binding key K_{bm}. In communications security the term authentication typically refers to the corroboration of a link between an identity of some kind and an aspect of the communications model, like a message or a session [5]. In this interpretation, binding update protocols provide *location authentication.*

5 Middleboxes

Protocols for the 'real' Internet have to consider so-called middleboxes like Network Address Translators (NATs) and firewalls. Protocols like HIP (Host Identity Protocol) provide mobile nodes with identifiers above the IP layer that do not change when nodes move and the IP address changes, and maintain a mapping between the identifier and the IP address. However, when the protocol has to traverse middleboxes several problems can arise. For example, a node may be behind a NAT so its true address is not visible to its peer so the middlebox may have to act as a proxy and provide the mapping between identifier and actual address. There may also be problems with firewalls that permit traffic only in one direction as the protocols updating the address often include messages in both directions.

As a further problem, a node may tell its firewall to let packets from its correspondent pass, but when the correspondent changes its location the firewall rules have to be updated. Then, schemes for protecting the instructions to the firewall have to be implemented. Issues of this kind are discussed, e.g. in [11]. These examples should illustrate why a simple 'Alice & Bob' model of communications is no longer appropriate when designing, and analysing security protocols at the IP layer.

6 Multi-layered Protocols

Multi-layered authentication protocols try to derive security properties at a higher protocol layer from guarantees given at a lower layer. For example, the

variants of EAP [1] use identifiers at different layers. A principal thus can be known by distinct identities at each layer. Hence, security analysis also has to check the binding that is intended (or not intended, when privacy is a goal) between the different identifiers of a single principal. An example for the pitfalls one has to be aware of when dealing with this issue is reported in [8]. Again, we note that the Alice & Bob view of the world is too simplistic.

7 Sensor Networks

The Canvas protocol [12] provides data integrity in a sensor network by relying on independent witnesses but does not provide data origin authentication at the same time. We will give a slightly abbreviated version of the discussion in [6]. Let us assume that nodes can communicate with their direct neighbours and have information about nodes in their vicinity, but no means for authenticating arbitrary nodes in the network. I.e., there is nothing like a public key infrastructure. Nodes share secret keys with nodes that are one or two hops away. Message authentication codes protect the integrity of messages transmitted between nodes that have shared keys.

Nodes can inject new messages into the network and forward messages they receive. We assume an algorithm exists for routing message sin the network. We want to achieve *data integrity*. Forwarded messages cannot be manipulated or inserted[3]:

Definition 1. *Data integrity is the property whereby data has not been altered in an unauthorized manner since the time it was created, transmitted, or stored by an authorized source [9].*

Defence against the creation of messages with bad content is a separate issue that is not being addressed here.

In this network, the creator of a message cannot vouch for its integrity as nodes further away would not share a key with the originator. The Canvas protocol thus uses interwoven authentication paths for data integrity. Forwarding of messages works as follows. Let K_{xy} denote a symmetric key shared by nodes X and Y and let A, B, C, D, denote nodes in the network. A message m is forwarded from B to C as follows (figure 2):

$$B \rightarrow C : m, A, B, D, h(K_{ac}, m), h(K_{bc}, m), h(K_{bd}, m)$$

Node A had forwarded the message to B, nominated C as the next node, and included the authenticator $h(K_{ac}, m)$. In turn, B nominates D as the next node and constructs authenticators $h(K_{bc}, m)$ and $h(K_{bd}, m)$. The recipient C checks the two authenticators $h(K_{ac}, m)$ and $h(K_{bc}, m)$, and discards m if authentication fails.

[3] Source [12] refers to message authentication, but in [9] this term appears as a synonym for data origin authentication.

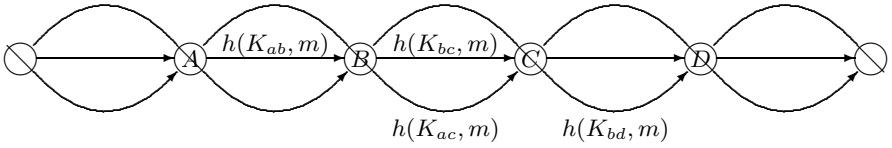

Fig. 2. The Canvas protocol

Obviously, if A and B collude they can modify m without being detected by C. However, it can be shown that the protocol achieves its goal if no two adversarial nodes are direct neighbours [12]. This observation contradicts a view widely held in communications security that *data integrity* and *data origin authentication* are equivalent properties, see e.g. [9, page 359].

Definition 2. *Data origin authentication (message authentication) is a type of authentication whereby a party is corroborated as the source of specified data created at some time in the past [9].*

By definition, data origin authentication includes data integrity. Conversely, in a communications system where the sender's identity (address) is an integral part of a message, a message with a forged sender address must not be accepted as genuine. To check the integrity of a message we would also have to verify its origin. Moreover, if messages pass through a network that is controlled by the adversary, we can only rely on evidence provided by the sender to verify that a message has not been altered in transit. For both reasons data integrity includes data origin authentication, but only under the specific assumptions we have made about the communications system.

As a final twist to this discussion, we show that there exists an attack if we adjust assumptions about the adversary. Adversarial nodes still cannot be direct neighbours but they may agree a-priori on a strategy for modifying messages and know their respective routing strategies. Two adversarial nodes A and C separated by a honest node B can collude to change a forwarded message m to $\tilde{m}$. The attack in figure 3 targets a node that can be reached in one hop from one of the adversarial nodes and in two hops from the other.

1. Adversary A forwards message m to B, naming C as the next node and including $h(K_{ae}, \tilde{m})$ in place of the authenticator $h(K_{ac}, m)$; E has to be a node that can be reached in one hop from C and in two hops from A.
2. Node B successfully checks the authenticators for m, names D as the next node, and forwards $h(K_{ae}, \tilde{m})$ unchecked.
3. Adversary C receives m from B, changes it according to the pre-arranged strategy to $\tilde{m}$, generates authenticators for the modified message, and forwards those together with $h(K_{ae}, \tilde{m})$ to E.
4. Node E receives the modified message $\tilde{m}$ with valid authenticators from A and C and accepts it as genuine.

This attack could be prevented if E knows about valid routes in the network. By assumption, A and C are not direct neighbours so messages could not arrive

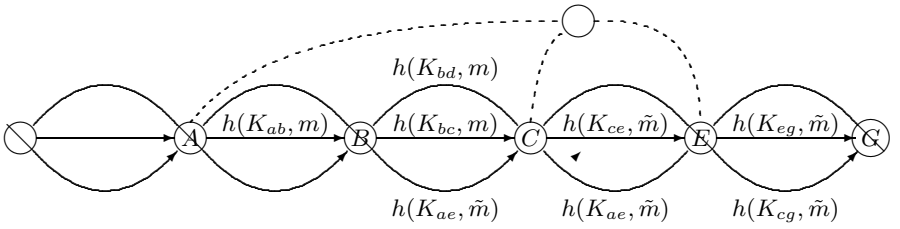

Fig. 3. An 'attack' on the Canvas protocol; dotted lines indicate unused links

along the route $A \to C \to E$. However, this would constitute yet another change in assumptions. So far, nodes were only storing keys for some neighbours but had no further information about the network topology.

8 Conclusion

We have given examples that have introduced location and return routability as new aspects that have to be captured in protocol analysis. We have pointed to issues that arise when parties have to communicate via middleboxes so that traffic identifiers change along a route. We have mentioned layered protocols and the problems of matching identifiers at different layers of the protocol stack. In the final example, security was relying on the fact that adversarial nodes are sufficiently isolated so that they cannot violate message integrity.

For the analysis of such protocols, we need methodologies that allow us to capture relevant aspects of the communications environment. We have to be able to specify which nodes are honest, and which communication links are not controlled by the adversary. We might have to accommodate new security properties, and maybe even new axioms for location-based arguments like return routability. For the analysis of such protocols, we may also have to change the way we think about security. Axioms in the logical derivation systems used may only hold under certain assumptions, and even the familiar security terminology may implicitly reflect assumptions about the communications system. The major challenge here is to check evaluation methodologies for traces of the Dolve-Yao model so that we understand which aspects of the methodology depend on its assumptions.

References

1. B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. *Extensible Authentication Protocol (EAP)*, June 2004. RFC 3775.
2. Tuomas Aura, Michael Roe, and Jari Arkko. Security of Internet location management. In *Proceedings of the 18th Annual Computer Security Applications Conference*, pages 78–87, December 2002.
3. Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication. *DEC Systems Research Center*, Report 39, revised February 22 1990.

4. Danny Dolev and Andrew C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, IT-29(2):198–208, March 1983.
5. Dieter Gollmann. Authentication by correspondence. *IEEE Journal on Selected Areas in Communications*, 21(1):88–95, January 2003.
6. Dieter Gollmann. Challenges in protocol design and analysis. In J.D. Tygar D.T. Lee S.P. Shieh, editor, *Computer Security in the 21st Century*, pages 7–22. Springer, 2005.
7. D. Johnson, C. Perkins, and J. Arkko. *Mobility Support in IPv6*, June 2004. RFC 3775.
8. Catherine Meadows and Dusko Pavlovic. Deriving, attacking and defending the gdoi protocol. In *Proceedings ESORICS 2004, LNCS 3193*, pages 53–72. Springer Verlag, 2004.
9. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, FA, 1997.
10. Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21:993–999, 1978.
11. Hannes Tschofenig, Andrei Gurtov, Aarthi Nagarajan, Murugaraj Shanmugam, and Jukka Ylitalo. Traversing middleboxes with the host identity protocol. In *Proceedings of ACISP 2005*, July 2005.
12. Harald Vogt. Integrity preservation for communication in sensor networks. Technical Report 434, ETH Zürich, February 2004.