AVISPA

www.avispa-project.org

IST-2001-39252

Automated Validation of Internet Security Protocols and Applications

Deliverable D6.1: List of selected problems

Abstract

This document presents the list of candidate protocols and problems for AVISPA. A problem is given by both a protocol and a security property the protocol should satisfy. (A protocol and a set of properties is a multiple problem). Our list contains a total of 384 security problems and 79 protocols, mostly from the IETF, divided into 33 groups. Since the desired security properties are often not explicitly stated in the original documents describing the protocols, the proposed properties may be revised after further feedback from the IETF or other organizations. This document also assesses the coverage of the proposed set of protocols.

Deliverable details

Deliverable version: *v1.0*
Person-months required: *3*
Date of delivery: *12.12.2003*
Due on: *30.11.2003*
Classification: *public*
Total pages: *55*

Project details

Start date: *January 1st, 2003*
Duration: *30 months*
Project Coordinator: *Alessandro Armando*
Partners: *Università di Genova, INRIA Lorraine, ETH Zürich, Siemens AG*

Project funded by the European Community under the *Information Society Technologies* Programme (1998-2002)

Contents

1 Introduction

The goal of this deliverable is to select candidate protocols and to present their security properties (goals), (on average, more than 4 properties per protocol). We assess the coverage and relevance of the resulting problem set, with comments of IETF representatives being taken into account.

Our list of candidate protocols includes 33 groups, 79 protocols and 384 security problems.

This list of candidate protocols is the basis for the success criteria for the project, namely,

- to specify at least 80 security problems from 20 groups,
- for at least 60 of these 80 problems, including at least one problem from each of the first seven groups (in the AVISPA proposal called the "Main Protocols"), to verify that the protocol satisfies the desired security property (at least in a certain configuration, with a bounded number of sessions) or to find a counterexample demonstrating that the property is violated, and
- to perform the verification of each problem in less than 1 hour of CPU time.

Not all IETF protocols are equally important for the proper operation of the Internet or for the secure support of applications; nor are all protocols suitable for the kind of formal verification that we plan to perform, either because their strength or weakness depends on the particulars of the cryptographic algorithms used, on policy-based operation, or on their performance. Some protocols are just "containers" that pass further unspecified authentication information. Some protocols have so many different layers, exceptions, configurations, and message exchanges that a reasonable abstraction and formalization is beyond the scope of our proposal.

Within these constraints, the purpose of this deliverable is to evaluate the protocols and select the candiate problems for formal specification. They are given by both the protocol description and a set of security properties the protocol should satisfy. In general, a protocol is designed to achieve a multitude of security goals, e.g. secrecy of session key k, or authentication of a peer in role A with strong agreement on nonce n. We therefore introduce the notion of a *security problem*, which is given by a protocol paired with a security property.

Taking into account that in many cases the desired security properties are not explicitly stated in IETF documents, work on this deliverable included

both thorough analysis of the protocols and the initiation of a discussion process with IETF representatives, with the selection being based on their comments on coverage and relevance to problem proposals by AVISPA.

Our list of 33 groups and 79 protocols covers most of the recent and ongoing IETF activities on security and security-sensitive applications. In specifying the IETF protocols and their associated problems, we will need to perform a certain amount of abstraction and simplification. Some of this *may* be necessary to deal with the limitations of our tools (e.g., with respect to the number of concurrent sessions, or agents, or on some data types), and some because certain features of the protocol (such as the cipher suites used, techniques for negotiating them, policy issues, or strength against some denial of service attacks) will be outside the scope of our analysis. Besides this, most of the IETF protocols are not written in a language that is readily translatable to a formal specification. We expect that in many cases there will be a need to discuss the protocols with their developers in order to properly interpret the intended meaning of the (proposed) standards. These discussions are conducted in continuation of the process initiated in this deliverable.

IETF group descriptions are found in:

`http://search.ietf.org/html.charters/`

and the current drafts can be found in:

`http://search.ietf.org/internet-drafts/`.

2 Coverage and Relevance Assessment

Having taken into consideration the feedback of IETF representatives, it is possible to assess the coverage of the proposed set of protocols by examining the individual charters of the working groups of the IETF and by reviewing the Internet Architecture Board documents on the Security of the Internet, and the Requests for Comments devoted completely to security, namely [32, 159, 33, 160, 174].

Our set of candidate protocols includes almost all protocols recommended by the Internet Architecture Board (IAB) during a meeting on 3-5 March 1997 in Murray Hill, NJ (reported in [32]). The IAB recommendations were:

- **"Core Security Mechanisms"**: The following mechanisms were designated as core in [32]:
  - **IPsec**, see Section 4.17.
  - **ISAKMP/Oakley**, see Section 4.17.

- **DNSsec**, see Section 4.12.
- **Security/Multipart**, [72], is the preferred way to add secured sections to MIME-encapsulated email, see Section 4.3 for a similar protocol, SIP-SMIME.
- **Signed keys in the DNS**, see Section 4.12.
- **X.509v3**, is the IETF standard for certificate profiles. It is not really a protocol, but the description of format and contents of certificates, together with their intended use. This is beyond the scope of AVISPA.
- **TLS**, see Section 4.25.

- **"Useful but not Core Mechanisms"**:
  - **AFT/SOCKS**, provides secured firewall traversal see [105, 104]. The security is based on GSSAPI (which is beyond our scope) or username/password (which is insecure). Initially, other methods were proposed (based on challenge-response and on the Extensible Authentication Protocol (EAP), see Section 4.13), but those methods have been abandoned. Thus, we do not consider AFT/SOCKS.
  - **RADIUS**, see Protocol IPv6-RADIUS in Section 4.18.
  - **Firewalls** are beyond the scope of AVISPA.
  - **GSS-API** is a framework for negotiating security mechanisms and its security properties are those of the negotiated mechanisms.
  - **PGP**, the key distribution of PGP is insecure, and the sending of encrypted or signed messages (in this case mail) using public key infrastructure is secure.
  - **Kerberos**, see Section 4.19.
  - **PKIX-CMP** (formerly PKIX-3), ([10]) is the suite of Certificate Management Protocols for the IETF version of the X.509 Public Key Infrastructure (PKI). It defines protocol messages for all relevant aspects of certificate creation and management.
  - **the various forms of per-hop authentication (OSPF, RSVP, RIPv2)**, see Section 4.5.
  - **APOP**, ([127], see Section 4.10).
  - **OTP**, see Section 4.14.

- **S/MIME**, see Section 4.3, for a quite similar protocol, the Protocol SIP-SMIME.
- **SSH**, see Section 4.23.
- **PFKey**, see [116] is a generic key management API that can be used not only for IP Security, but also for other network security services. This is beyond the scope of AVISPA.
- **IPsec API** (see [175]) provides a set of facilities which an IPsec implementation should provide to applications to allow them to both observe and influence how IPsec protects their communications. This is beyond the scope of AVISPA.
- **SASL** is a framework for negotiating security mechanisms and its security properties are those of the negotiated mechanisms.
- **CRAM**, see Protocol CRAM-MD5 in Section 4.10.
- **CHAP**, see Protocol ChapV2 in Section 4.21.

The list is five years old (it was published as RFC in April 1998); an update would be of interest. For instance, perhaps today Security/Multipart would not be seen as "core" security mechanism, but rather CMS (Cryptographic Message Syntax, see [85, 86, 196]).

We will not model any of the protocols considered as "not useful" or unacceptable in [32], in particular any protocol where plaintext passwords are sent over unencrypted channels.

For an overview of the security mechanisms for the Internet see [33]. Our protocols also cover most of them. While there is partial overlap between this list and the previous one from [32], we include both lists for easy reference to the sources.

The mechanisms described in [33] are:

- **One-Time Passwords**, see Section 4.14.
- **HMAC**, is a generic security mechanism, (a hash) used in many other protocols. It is not a protocol in the sense of AVISPA.
- **IPsec**, see Section 4.17.
- **TLS**, see Section 4.25.
- **SASL** is a framework for negotiating an authentication and encryption mechanism to be used over a TCP stream. The properties are those of the negotiated mechanism.

- **GSS-API**, like SASL, is a framework for negotiating an security mechanisms and its security properties are those of the negotiated mechanism.
- **DNSSEC**, see Section 4.12.
- **Security/Multipart**, see Section 4.20.
- **Digital Signatures** are a generic security mechanism used in many other protocols and by itself, is secure.
- **OpenPGP and S/MIME**, see Section 4.3.
- **Firewalls and Topology** are beyond our scope, but related methods ("Return Routability", in which some "regions" of the Internet are assumed to be secure) are discussed in Section 4.1.
- **Kerberos**, see Section 4.19.
- **SSH**, see Section 4.23.

Our protocols also cover most of the recommended authentication mechanisms for the Internet described in [159]:

- **One-Time Password Systems**
  - **S/Key**, see Section 4.14
  - **OTP**, see Section 4.14
  - **SecureID** is a simple proprietary protocol, based on a smart card, that produces a stream of passwords to be used only once. The sender and the receiver are synchronized. The strengths and weaknesses of the protocol are well known.
- **Challenge-Response Systems**
  - **APOP**, ([127]), see Section 4.10.
  - **ACAP** [132], very close to [100], discussed in Section 4.10 (Protocol CRAM-MD5).
  - **HTTP Digest** [69], is part of the SIP security, discussed in Section 4.3.
  - **AKA**, see Section 6.1.
  - **CRAM-MD5**, see Section 4.10.

  - **Kerberos**, see Section 4.19.
  - **SIM** is an insecure protocol, similar, but much simpler than Protocol AKA of Section 6.1.

- **"Zero Knowledge" Password Proof Systems** (See Section 4.16):
  - **EKE**
  - **A-EKE**
  - **SPEKE**
  - **SRP**

- **Server Certificate Systems**
  - Protocols over **SSL/TLS**; this is one variant of the protocols of Section 4.25.
  - **IPsec (under some conditions)**, this is one variant of the protocols of Section 4.17.

- **Mutual Public Key Systems**
  - **SSL/TLS (client auth mode)**, this is one variant of the protocols of Section 4.25.
  - **IPsec IKE**, this is one variant of the protocols of Section 4.17.
  - **S/MIME**, see Section 4.3

- **Generic Authentication Systems**. In general these are "containers" that pass further unspecified authentication information for negotiating security mechanisms. Their security properties are those of the negotiated mechanisms.
  - **GSS-API**
  - **SASL**
  - **EAP**

- **Authentication Server Systems**
  - **Kerberos**, see Section 4.19
  - **RADIUS**, see Protocol IPv6-RADIUS in Section 4.18
  - **DIAMETER**, see Protocol AAA-MIP in Section 4.1

Our list of protocols does not include two groups of insecure authentication mechanisms for the Internet discussed but not recommended in [159]:

- **Passwords in the Clear**
  - TELNET (basic authentication)
  - HTTP (basic authentication)
  - SASL (password mode)
  - RLOGIN
  - POP
  - IMAP
- **Anonymous Key Exchange Mechanisms**
  - SSH (password mode)
  - SSL/TLS (anonymous keying)

The other two RFCs solely dedicated to security ([160, 174]) do not contain lists of protocols or mechanisms, but discuss the types of security properties that an Internet may want to have and the attacks that it may be subject to. We have gone through the two documents with great care and incorporated all relevant security properties into our list.

3 Properties (Goals)

The usual properties referred to as security properties (or security goals) in IETF documents are the following (see [160, 174, 201, 3], and also [87, 118]).

1. **Authentication (unicast)**: Verifying an identity (distinguishing identifier) claimed by or for a system entity, which may be a peer in a communication or the source of some data. This assured Identity may be well known (a real name, telephone number, mailing address, phone number, social security number, IP- or email address) or it can be an unlinkable identifier (like a pseudonym). The verification is achieved presenting authentication information (credentials) that corroborates the binding between the entity and the identifier. Authentication is usually divided into entity and message (or data) authentication. The main difference between the two is that message authentication provides no timeliness guarantee (the authenticated message may be old),

while entity authentication implies actual communication with an associated verifier during execution of the current run of the protocol. Authentication is usually unilateral ("Alice authenticates Bob"). *Mutual Authentication* refers to Authentication in both directions.

(a) **(G1) Entity authentication (Peer Entity Authentication)**: Assuring one party, through presentation of evidence and/or credentials of the identity of a second party involved in a protocol, and that the second has actually participated during execution of the current run of the protocol. Usually this is done by presenting a piece of data that could only have been generated by the second party in question (as a response to a challenge, for instance). Thus, usually entity authentication implies that some data can be unequivocally traced back to a certain entity, which implies Data Origin Authentication.

(b) **(G2) Message authentication (Data Origin Authentication)**: The protocol must provide means to ensure confidence that a received message or piece of data has been created by a certain party at some (typically unspecified) time in the past, and that this data has not been corrupted or tampered with, but without giving uniqueness or timeliness guarantees. The confidence that data has been created by a certain party, but without the assurance that it has not been modified, is of no interest for us. Thus Message authentication implies integrity. See also [174] "Relationship between data integrity service and authentication services". Only very few Internet protocols offer Data Origin Authentication without providing Entity Authentication (IPsec AH or PKI Signatures would be examples, both relatively trivial to verify). In our list of candidate protocols for AVISPA we have no protocol in this category.

(c) **(G3) Replay Protection**: Some IETF documents define Replay Protection as "The protocol must provide means to ensure confidence that a received message has not been recorded and played back by an adversary". As such, this property is not verifiable. We define it rather as: Assuring one party that an authenticated message is not old. Depending on the context, this could have different meanings:

- that the message was generated during this session, or
- that the message was generated during a known recent time window, or

- that the message has not been accepted before.

2. **Authentication in Multicast or via a Subscribe / Notify Service**: These are the authentication requirements for groups with a single source and a very large number of potential recipients (multicast), or a source and a service which posts the information to subscribed (and authorized) users. The basic requirements for the solution are:

   (a) **(G4) Implicit Destination Authentication** The protocol must provide means to ensure that a sent message is only readable by entities that the sender allows. That is, only legitimate authorized members will have access to the current information, multicast message or group communication. This includes groups with highly dynamic membership.

   (b) **(G5) Source Authentication** Legitimate group members will be able to authenticate the source and contents of the information or group communication. This includes cases where group members do not trust each other.

3. **Authorization (by a Trusted Third Party)**: In some protocols, a Trusted Third Party **T3P** introduces one principal **B** to another principal **A** and **A** is assured that **B** is "trusted" by the **T3P** and is "authorized", in the required sense of the protocol. When the protocol is run between **A**, **B** and **T3P**, then **A** is perhaps not able to use local access control lists or other mechanisms to authorize **B**, (because the name of **B** is unknown to **A**, or may even be a pseudonym), but **A** is assured that **B** is authorized by **T3P**.

4. **Key Agreement Properties**:

   (a) **(G6) Key authentication** is the property whereby one party is assured that no other party aside from a specifically identified second party (and possibly additional identified trusted parties) may gain access to a particular secret key.

   (b) **(G7) Key confirmation (Key Proof of Possession)**: one party is assured that a second (possibly unidentified) party actually has possession of a particular secret key (or of all keying material needed to calculate it).

   (c) **(G8) Perfect Forward Secrecy (PFS)**: A protocol has this property if compromise of long-term keys does not compromise past session keys.

(d) **(G9) Fresh Key Derivation** The protocol uses dynamic key management in order to derive fresh session keys.

(e) **(G10) Secure capabilities negotiation (Resistance against Downgrading and Negotiation Attacks)**. When a key agreement protocol also discovers the cryptographic capabilities and preferences of the peers and negotiates the security parameters (such as security association identifiers, key strength, and ciphersuites), it is important to ensure that the announced capabilities and negotiated parameters have not been forged by an attacker.

5. **(G11) Confidentiality (Secrecy)**: The property that a particular data item or information (usually sent or received as part of the content of a "secured" message, or else constructed on the basis of exchanged data) is not made available or disclosed to unauthorized individuals, entities, or processes, and remains unknown to the intruder. We choose the convention that the secrecy of a session key generated during a key agreement is not considered here but in Goal "Key authentication" above. Also the secrecy of a long-term key used within a protocol is not part considered as a secrecy goal of the protocol.

6. **Anonymity** Many protocols do not provide anonymity, since the peer has to know with whom he is speaking in order to determine which cryptographic key to use. But some protocols do hide the Identities:

   (a) **(G12) Identity Protection against Eavesdroppers**: An attacker (eavesdropper) should not be able to link the communication exchanged by one party to the real identity of the party.

   (b) **(G13) Identity Protection against Peer**: The peer in a communication should not be able to link the communication exchanged by one party to the real identity of the party, but rather to an unlinked pseudonym or private identifier.

7. **(G14) (Limited) Denial-of-Service (DoS) Resistance**: It is difficult to verify DoS Resistance. One reason is that a protocol may be subject to DoS attacks for many different reasons, the most common being that it consumes too many resources (memory, computational power), before the peer authenticates itself. But there are many other reasons: among others, protocols may be vulnerable to

   (a) DoS on memory allocation,

   (b) DoS on computational power, and

(c) Bombing Attacks on third parties. (This is inducing one or several hosts to send large amounts of packets to a victim).

[117] shows how some principles that have already been used to make protocols more resistant to denial of service can be formalized.

8. **(G15) Sender Invariance**: A party is assured that the source of the communication has remained the same as the one that started the communication, although the actual identity of the source is not important to the recipient.

9. **Non-repudiation** Prevention of a user wrongly denying having performed an action, in particular:

   (a) **(G16) "Accountability"**: The property of a system (including all of its system resources) that ensures that the actions of a system entity may be traced uniquely to that entity, which can be held responsible for its actions.

   (b) **(G17) Proof of Origin** Undeniable evidence of having sent a message.

   (c) **(G18) Proof of Delivery** Undeniable evidence of having received a message.

10. **(G19) Safety Temporal Property**: Using two temporal-logic operators (for linear temporal logic), the "Always" and the "Sometime in the Past" operators, it is possible to formalize properties of the form: in any reachable state that has a property p, there was in the past a state with property q. Formulaically: $\Box(p \Rightarrow \overline{\Diamond} q)$. Properties of this kind, which are not directly seen as standard security properties, are: if a user has to pay for a service, then he must have obtained the service, or if a host receives a large amount of packets in streaming mode, then he must have asked for them. Notice also that the security properties discussed above are all expressible in the form of a Safety Temporal Property.

In the last few months, a new set of properties have been discussed in some IETF drafts (in particular, at the Extensible Authentication Protocol (EAP) Working Group, see [37]) and Section 4.13). The properties are:

- **Session Formation** A protocol accomplishes session formation by binding each particular protocol instance with a unique value, the generalized session ID, which is usually a distinct tuple of nonces and

identifiers. This generalized session ID distinguishes this session from any other sessions. After an initial part of the protocol, all mesages contain the generalized session Identifier (not necessarily in cleartext).

- **Consistent View** When an instance of protocol succeeds, all parties in the run share the same view of the participants in the protocol instance and their respective roles, and of the protocol state.
- **Key naming** In order to ensure against confusion between the appropriate keying material to be used in a given secure association protocol exchange, the protocol must include explicit key names and context appropriate for informing the authenticator how the keying material is to be used. If the protocol provides key proof of possession, the protocol must explicitly name the keys used during the the proof of possession exchange, so as to prevent confusion when more than one set of keying material could potentially be used as the basis for the exchange.

We are optimistic that we will be able to formalize and verify these properties in AVISPA, but it is still not fully clear how to do it.

Other properties that are sometimes discussed as relevant security properties at the IETF (see in particular [37]) include:

- **Cryptographic Separation of Keys**
- **Cipher suite negotiation**
- **Dictionary Attack Resistance**
- **Cryptographic Binding**
- **Support for fast reconnect**
- **Acknowledged success and failure indications**
- **Session independence**
- **Man-in-the-Middle Attack Resistance**
- **Peer Liveness**

The last two are not really new security properties, they relate directly to the Authentication properties discussed above.

The first seven properties are not model-checking properties, but are verified by other methods, outside of the scope of AVISPA.

4 The IETF Protocols

Please note that since the desired security properties are often not explicitly stated in the IETF documents, the proposed properties (goals) of the protocols may be revised after further feedback from IETF representatives and authors.

4.1 MobileIP

IP Routing for Wireless/Mobile Hosts

The Mobile IP Working Group has developed routing support to permit IP nodes to seamlessly "roam" among IP subnetworks and media types. The Mobile IP method supports transparency above the IP layer, including the maintenance of active TCP connections and UDP port bindings.

Protocol **AAA-MIP (1)** is used by a mobile node to authenticate in a visited network in order to receive service from foreign service providers, using the Internet Authentication, Authorization and Accounting (AAA) infrastructure.. It is defined in the Diameter Base Protocol, [44], together with the Diameter Mobile IPv4 Application, [43], and related drafts ([147, 146]). During the protocol, the Home AAA server authenticates the mobile node (MN), (Problem: Entity authentication), and the MN is guaranteed that he is attached to a Foreign Agent in a "trusted" Visited Domain.

As part of the exchange, three short-lived keys are distributed: the Mobile-Home Session Key, the Mobile-Foreign Session Key, and the Foreign-Home Session Key. These three keys should be secret (Problems: Key authentication for Mobile-Home, for Mobile-Foreign, and for Foreign-Home).

Protocol AAA-MIP should provide Fresh Key Agreement and 3P-Authorization (G1-3,6,7,10,12).

Protocol **MIP-BU (2)** is a protocol used to secure a mobile-ip message known as a binding update (BU). When a mobile node moves (and in other circumstances), it sends to the correspondent a message called a binding update (BU), describing the new care-of-address needed to reach the mobile node directly. This BU must be secured, as otherwise anyone could pretend that a given mobile node had moved. Since we cannot assume the existence of a global PKI or other global security infrastructure, it is difficult to secure this packet, sent by any mobile node to any other Internet node (and those two nodes may have no previous relationship or common security infrastructure). What should be possible to ensure, at least in some versions of the protocol, is that the sender of the BUs does not change (Sender Invariance). The protocol design of [93] is unusual and would not be considered secure by the measures of traditional security protocol analysis. The security of

the protocol depends on the partial reliability of the Internet routing infrastructure. Some "regions" of the Internet may be considered rather secure, and any insecurity in those regions that could be exploited in this setting, could also be exploited in the non-mobile case. But the only security requirement of the BU protocol is to counter the new threats created by mobility. The major problem when designing an unauthenticated BU is, surprisingly enough, the introduction of new Denial of Service threats, see [25].

In summary, Protocol MIP-BU should provide Sender Invariance and DoS Resilience (G15,16).

4.2 seamoby

Context Transfer, Handoff Candidate Discovery, and Dormant Mode Host Alerting

The Context Transfer Protocol Protocol **seamoby-ctp (3)** is a protocol used whan a MN is moving (or may be moving) from one router to another, allowing state information to be transferred between edge mobility devices. Examples of state information that could be useful to transfer include AAA information, security context, QoS properties assigned to the user, Robust Header Compression information, etc. [96] explains the main reasons why Context Transfer procedures may be useful in IP networks.

[109] describes the protocol, but security details are missing. We assume that the protocol will be fully specified within the next 6 months, or we will make assumptions about the missing details. The current proposal seems to be enough to construct a model for verification purposes. It proceeds roughly as follows: assume that a mobile node MN moves from one previous access router pAR to a new one, nAR, and that the two ARs have established a secure channel in advance (for instance over IPSec). To be more specific, we restrict ourselves to the case in which the context transfer is request by MN, the mobile node. Then either one or both of the pAR and nAR authenticates MN and authorizes the MN's credentials before authorizing the context transfer and release of context to the mobile. This should prevent the possibility of rogue MNs launching DoS attacks by sending large number of CT requests as well as causing a large number of context transfers between ARs. Another consideration is that the mobile provides an authentication "cookie" to be included with the context transfer message sent from the pAR to the nAR and confirmed by the MN at the nAR.

The most important properties here are temporal properties, for instance, if the context of a MN is moved to a certain AR, then the MN has indeed requested the context transfer. The protocol also has authentication and secrecy requirements, including that the ARs authenticate the MN and

vice-versa, and that the MN authenticates the authentication cookie (sent originally by itself) presented by the new AR.

For related drafts see [108] and [184].

Protocol seamoby-ctp should provide Key Agreement, 3P-Authorization, and a property that may be expressed as a Temporal Formula (G1-3,6,7,12, 20).

4.3 SIP

Session Initiation Protocol

SIP [171, 169, 170] is a text-based protocol, similar to HTTP and SMTP, for initiating interactive communication sessions between users. Such sessions include voice, video, chat, interactive games, and virtual reality. Due to the complexity of the protocol, many distinct mechanisms have been proposed to secure different aspects and usages of SIP. We will concentrate on the most common ones, Protocol **SIP-Digest (4)**, (see [171], Section 22) re-uses the seurity mechanisms developed for HTTP.

Protocol SIP-Digest should provide Authentication (G1,2).

For Protocol **SIP-SMIME (5)**, defined in Section 23 of the same RFC [171], we assume the existence and provision of user certificates.

Protocol SIP-SMIME should provide Authentication and Secrecy (G1,2, 12) (more explicitely: User-to-User Authentication, User-to-User Content Secrecy, and Proxy-to-User Authentication).

Related RFCs and drafts are: [134], [149], [202], [92], [16], [27], [62], [112], [154], [155], [153], [110], [197], and [200].

4.4 H323 Suite: H530

The ITU-T Study Group 16 is working on multimedia security for H.323 mobility. One of the general problems addressed aims at enabling mobility in small and medium sized H.323-based corporate and enterprise networks.

The ITU Protocol **H530 (6)** is defined in [89] and [90]. (See `http://www.itu.int/rec/recommendation.asp`). In this protocol, a mobile H.323 user attaches at a foreign visited H.323 network domain in order to obtain H.323 services from the corresponding subscribed home domain; e.g. forwarded H.323 Voice-over-IP calls. The security issues of concern in such a scenario include mutual authentication of the mobile user and the visited domain, mutual authentication of the visited domain to the home domain and secure key agreement between the mobile user terminal and the visited gatekeeper.

Protocol H530 should provide Fresh Key Agreement and 3P-Authorization (G1-3,6,7,10,12).

4.5 NSIS

Next Steps in Signaling

End-to-end Quality of Service (QoS) is needed for voice over IP and many other applications using the global Internet. The current signalling solution is RSVP, defined in [40]. The security extensions of [209, 26, 36] define the Protocol **RSVP-sec (7)**. For a discussion of the security properties of the protocol, see also [188].

Protocol RSVP-sec should provide Authentication (G1,2).

For end-to-end QoS, needed in many applications, it is likely that several administrative domains are traversed, but this may be problematic if the domains deploy different QoS solutions. Thus a protocol must signal the QoS parameters from one domain to the other. The NSIS WG is currently discussing the framework for a new simple protocol for signaling QoS, which would allow users to obtain QoS-aware services irrespective of the underlying mechanisms used. Compatibility with authentication and authorization mechanisms is also considered.

NSIS has not yet defined a protocol that will be interesting for our purposes. Perhaps the WG will not define a specific protocol. Nevertheless, Siemens is strongly involved in the activities of the group and may provide to AVISPA a version of Protocol **NSIS-acc (8)**, a next generation QoS signalling protocol that includes accounting and that could be proposed eventually at the IETF or elsewhere. In AVISPA, we intend to model and verify such a protocol.

The property that this protocol has to possess is easily posed as a temporal logic property: the user should not pay if he obtains no service, or in other words: if he pays, then he has obtained the service.

Protocol NSIS-acc should provide Authentication, Secrecy, Identity Protection (Eavesdropper and Peer) and a property that may be expressed as a Temporal Formula (G1,2,12-14,20).

For the framework and related drafts, see [42], [79], [192], [188], [187], and [189].

4.6 Geopriv

Some applications need to acquire geographic location information about certain resources or entities. These applications include navigation, emergency

services, management of equipment in the field, and other location-based services. The main issues arising from generating, using, and passing location information about users are privacy related, see [51] and [65].

Geopriv has not yet defined (and perhaps will not define) a protocol that will be interesting for our purposes. Nevertheless, Siemens, being strongly involved in the activities of the group, may provide to AVISPA a version of Protocol **Geopriv-nym (9)**, a pseudonym agreement protocol that could be used in a geopriv architecture.

The properties of the pseudonym agreement, together with the underlying geopriv protocol, give rise to the following Problems: first, a temporal logic property, namely, the location of the user should not be sent to anyone without prior user consent; and second a privacy property, namely the Location Server (an involved third party) does not learn the real identities of the user or the location recipient.

Protocol Geopriv-nym should provide Authn + ID Protection (Eavesdropper and Peer) (G1,2,13,14).

4.7 impp and simple

Instant Messaging and Presence Protocol and SIP for Instant Messaging and Presence Leveraging Extensions

A presence and instant messaging system allows users to subscribe to each other and be notified of changes in state, and for users to send each other short instant messages. Protocol **Impp (10)** ([53, 52, 21, 177, 150, 152, 151]) develops an architecture for simple instant messaging and presence awareness and notification. It specifies in particular how authentication, message integrity, encryption and access control are integrated. Protocol **Simple (11)** (see in particular [164, 173, 167, 168, 165, 166]) focuses on the application of SIP to instant messaging and presence (IMP), compliant with the requirements listed in [52] and including the security and privacy requirements.

Each of the 2 protocols should satisfy the following security properties: Message Authentication, Replay Protection, Implicit Destination Authentication

In summary, both protocols should provide Authentication, Secrecy, Implicit Destination Authentication, and Replay Protection (G1-4,12).

4.8 AAA

Authentication, Authorization and Accounting

Protocol **aaa-nasreq (12)** is used for Authentication, Authorization and Accounting (AAA) services in the Network Access Server (NAS) environment. It is defined in [45], in combination with [44, 6, 64]. Together, they satisfy the NAS-related requirements defined in [4] and [28].

Protocol aaa-nasreq should provide Fresh Key Agreement and 3P-Authorization (G1-3,6,7,10,12).

4.9 cat

Common Authentication Technology

The goal of the Common Authentication Technology (CAT) Working Group is to provide distributed security services to a variety of protocol callers in a manner which insulates those callers from the specifics of underlying security mechanisms.

Protocol **SPKM-LIPKEY (13)**, the Simple Public-Key GSS-API Mechanism (SPKM) and LIPKEY, A Low Infrastructure Public Key Mechanism Using SPKM, [8, 61], provide a method to supply a secure channel between a client and server, authenticating the client with a password, and a server with a public key certificate. This is analogous, but not identical, to the common usage of the Transport Layer Security (TLS) protocol [54].

Protocol SPKM-LIPKEY should provide Fresh Key Agreement (G1-3,7, 10,12).

4.10 Challenge-Response Systems

Protocol **CRAM-MD5 (14)** ([100]) is a classical challenge/response authentication extension for IMAP [49, 126]: the server provides a random challenge and the client transmits an HMAC of the challenge using the shared key as the HMAC key.

Protocol CRAM-MD5 should provide Authentication, Secrecy and Replay Protection (G1,2,3,12).

Protocol **APOP (15)**, defined in [127] as part of POP3, is a simple method of authentication using timestamps as nonces, which provides for both origin authentication and replay protection, but which does not involve sending a password in the clear over the network.

Protocol APOP should provide Authentication, Secrecy and Replay Protection (G1,2,3,12).

Protocol ACAP [132], is very close to [100] and thus not included in our list.

Protocol HTTP Digest [69], is actually part of the SIP security protocol considered in Section 4.3

Protocol AKA is discussed in Section 6.1.

Protocol Kerberos is discussed in Section 4.19.

Protocol SIM the predecessor of AKA, has evident security flaws (it does not provide mutual authentication). A more interesting (and more complex) version is Protocol EAP-SIM, described in Section 4.13.

4.11 DHC

Dynamic Host Configuration

The draft Authentication for DHCP Messages, [56], defines two simple mechanisms for authentication in DHCP. One of them, Protocol **DHCP-delayed (16)**, the Delayed Authentication method, combined with the Key Management Technique defined in the Appendix of [56] should be enough to secure against the common DHCP threat model, in particular, the establishment of "rogue" DHCP servers with the intent of providing incorrect configuration information to the client.

Protocol DHCP-delayed should provide Authentication, Secrecy and Replay Protection (G1,2,3,12).

4.12 DNSext

DNS Extensions

The Domain Name System (DNS) [121, 122, 63, 99] is a replicated hierarchical distributed database system that provides information fundamental to Internet operations, such as name to address translation (and vice-versa) and mail handling information. The basic documents are extended in [14, 12, 13] to provide for data origin authentication and public key distribution, all based on public key cryptography and public key based digital signatures. The Protocol **DNSSEC (17)** also includes the documents [58, 203, 204, 106, 48, 73, 205, 113, 20].

Protocol DNSSEC should provide Authentication and Replay Protection (G1,2,3).

The Protocol **TSIG (18)** includes operations such as dynamic update with transaction signatures and secret key establishment. TSIG uses symmetric cryptography and is described in [198, 59, 57, 203].

Protocol TSIG should provide Key Agreement (G1-3,7,12).

Protocol **SIG(0) (19)**, a variant of the Transaction Signatures that uses asymmetric cryptography, where the public keys are stored in DNS, is presented in [60, 163]

Protocol SIG(0) also should provide Key Agreement (G1-3,7,12).

4.13 EAP

The Extensible Authentication Protocol (EAP) framework, described in [37, 3, 199] and other drafts, provides a standard mechanism for support of additional authentication methods within PPP, IEEE 802 wired networks, IEEE 802.11 (with IEEE 802.1X) or other access technology.

The Protected Extensible Authentication Protocol (Version 2), Protocol **PEAP (20)**, provides an encrypted and authenticated tunnel based on transport layer security (TLS) that encapsulates EAP (Extensible Authentication Protocol) authentication mechanisms. [94]

Protocol PEAP should provide Fresh Key Agreement, 3P-Authorization, and ID Protection (Eavesdropper) (G1-3,6,7,10,12,13).

The PEAP conversation occurs between the EAP peer and EAP server, passing through a Network Access Server, NAS. The NAS is not involved in the PEAP conversation, which is confidentially protected, and therefore the NAS does not have knowledge of the TLS master secret derived between the peer and the EAP server. In order to provide keying material for link-layer purposes, the NAS obtains the master session key, which is derived from a one-way function of the TLS master secret as well as keying material provided by EAP methods. This enables the NAS and EAP peer to subsequently derive transient session keys suitable for encrypting, authenticating and integrity protecting session data. However, the NAS cannot decrypt the PEAPv2 conversation or spoof session resumption, since this requires knowledge of the TLS master secret.

The protocol is quite complex and includes 2 different parts, each one with several phases and variants: The first part is as follows: The initial identity exchange is used primarily to route the EAP conversation to the EAP server. Then a TLS session is established. TLS itself has several variants, depending in particular on the type of credentials used. If required for privacy reasons, the TLS session may be re-negotiated after the first server authentication. Then a version negotiation procedure enables PEAP implementations to be backward compatible with previous versions of the protocol. It should guarantee that the EAP peer and server will agree to the latest version supported by both parties. There is also a variant for resuming a previously established session. The second part of the PEAPv2 conversation typically consists of a complete EAP conversation occurring within the TLS session negotiated in Part 1. Besides being a standard Key Exchange Problem, TLS also offers identity protection against eavesdroppers.

Protocol **EAP-SIM (21)**, the EAP SIM Authentication of [82], is an authentication and session key distribution method based on the GSM Subscriber Identity Module SIM mechanism, a challenge/response authentication

and key agreement procedure based on a symmetric 128-bit pre-shared secret. EAP SIM also makes use of a peer challenge, not part of GSM, to provide mutual authentication. EAP-SIM specifies optional support for version negotiation and for protecting the privacy of the subscriber identity using the same concept as GSM, which uses pseudonyms/temporary identifiers. This protocol is also quite complex and includes Version Negotiation, Identity Management, Re-Authentication, EAP Notifications, Error Case Handling, and Key Generation, all including several phases and variants.

Protocol EAP-SIM should provide Fresh Key Agreement and 3P-Authorization (G1-3,6,7,10,12).

Protocol **EAP-AKA (22)**, the EAP AKA Authentication, proposed in [15], is a mechanism for authentication and session key distribution using the Universal Mobile Telecommunications System (UMTS) Authentication and Key Agreement (AKA) mechanism. UMTS AKA is based on symmetric keys and typically runs in a UMTS Subscriber Identity Module, a smart card like device. EAP AKA includes optional identity privacy support, an optional re-authentication procedure, EAP Notifications, Error Case Handling, Key Generation, and the protocol exists in different variants.

Each of the 2 protocols offers us a wide range of security problems: Identity Protection, Mutual Authentication, Key Derivation, Brute-Force and Dictionary Attacks, Integrity Protection, Replay Protection and Confidentiality, Negotiation Attacks.

Protocol EAP-AKA should provide Fresh Key Agreement, 3P-Authorization, and ID Protection (Eavesdropper and Peer) (G1-3,6,7,10,12-14).

Protocol **EAP-Archie (23)**, the EAP Archie Protocol, defined in [201], performs mutual authentication, and fresh session key derivation. EAP-Archie is based on a static long-lived 512-bit secret, the Archie Key, shared between the two main entities in the protocol, the EAP Peer and the EAP Server. The Archie Key consists of two 128-bit sub-keys and a 256-bit sub-key, respectively called the key-confirmation key (KCK), the key-encryption key (KEK), and the key-derivation key (KDK). The protocol uses the KCK to mutually authenticate the EAP Peer and the EAP Server, the KEK to distribute secret nonces used for session key derivation, and the KDK to derive a session key between the EAP Peer and the EAP Server.

Protocol EAP-Archie should provide Fresh Key Agreement, 3P-Authorization, and DoS Resilience (G1-3,6,7,10,12,15).

Newly discovered Man-in-the-middle attacks in the context of tunneled authentication protocols (see [157] and [18]) are applicable to IKEv2 if legacy authentication with EAP is used ([95, 38]). To counter this threat Protocol **EAP-IKEv2 (24)** was designed ([191]), providing a session key inside the AUTH payload.

EAP-IKEv2 consists of three parts plus an optional DoS protection exchange.

Protocol EAP-IKEv2 should provide Fresh Key Agreement, 3P-Authorization, and DoS Resilience (G1-3,6,7,10,12,15).

Protocol **EAP-TTLS (25)**, the EAP Tunneled TLS Authentication Protocol, is defined in [71] (see also [5]).

Protocol EAP-TTLS should provide Fresh Key Agreement and 3P-Authorization (G1-3,6,7,10,12).

4.14 One Time Password Systems

Protocol **OTP (26)**, One Time Password Protocol, is defined in [119, 78, 131].

Protocol OTP should provide (G1,2,12. (Confidentiality of the password)). Protocol **S/Key (27)** is defined in [77]

Protocol S/Key should provide Authentication and Secrecy (G1,2,12).

Protocol **SecureID (28)** is defined in [172]. Software residing on a PC or on a HW Token (smart-card like device) generates a random, one-time-use access code that changes every (say) 60 seconds. The user enters his PIN (Personal Identification Number) into the client software interface, and receives a SecurID passcode. The passcode is routed to the Server for verification, and if valid, the user gains access.

Protocol SecureID should provide Authentication and Secrecy (G1,2,12).

4.15 IDR

Inter-Domain Routing

Attacks on Routing Protocols are a major problem for the Internet. If an attacker is able to corrupt or modify routing tables as he chooses, he has a great amount of power to mount further attacks or simply to create massive DoS attacks. The IETF has been seriously concerned about these threats and has started a set of working items to deal with them.

The current proposal for the Border Gateway Protocol (Version 4, BGP-4) is described in [158]. The BGP Security Vulnerabilities are analyzed in [125].

Two main solutions for securing BGP have emerged: Protocol **S-BGP (29)**, Secure BGP, ([111]) and Protocol **soBGP (30)**, Secure Origin BGP, ([133]).

S-BGP addresses seven security goals:

1. Each update received by a BGP speaker from a peer was sent by the indicated peer and was not modified en-route from the peer.

2. Each update contains routing information no less recent than the routing information previously received for the indicated prefixes from that peer.

3. The update was intended for receipt by the peer that received it.

4. The peer that sent the update was authorized to advertise the routing information contained within the update.

5. The entity with the right to use an address space corresponding to a reachable prefix advertised in an update was given custodianship of that address space by a higher-level/parent entity.

6. The originating Domain was authorized, by the entity(s) with the right to use address space corresponding to the set of reachable prefixes, to advertise those prefixes.

7. If the update indicates a withdrawn route, then the peer withdrawing the route was a legitimate advertiser for that route, prior to its withdrawal.

soBGP addresses two security goals:

1. Is the Domain originating the destination authorized to advertise it? In other words, if a router receives an advertisement for the *dest* network originating in *advertiser*, is there any way to verify that *advertiser* is supposed to be advertising *dest*?

2. Does the Domain advertising the destination actually have a path to the destination? In other words, if a router is receiving an advertisement from *advertiser* that it can reach *dest*, is there any way to verify that *advertiser* actually has a path to the Domain *dest*?

To achieve those goals, Authentication is also necessary.

In summary, both Protocol S-BGP and Protocol soBGP should provide Authentication, 3P-Authorization, and a property that may be expressed as a Temporal Formula (G1,2,6,12,20).

4.16 Password-Authenticated Key Exchange

Password-Authenticated Key Exchange

A whole range of protocols rather robust against dictionary attacks has appeared. The list includes Protocol **EKE (31)**, Encrypted Key Exchange, defined in [34], Protocol **EKE2 (32)**, a second Encrypted Key Exchange, defined in [30], Protocol **SPEKE (33)**, defined in [91], and Augmented-EKE, Protocol **A-EKE (34)**, which has been described in [35] and further discussed in [176].

Protocol **SRP (35)**, the SRP Authentication and Key Exchange System, is defined in [208].

Each of these protocols, Protocol EKE, Protocol EKE2, Protocol SPEKE, Protocol A-EKE, and Protocol SRP should provide Authentication and Secrecy (G1,2,12). (Depending on the version they also provide Key Agreement).

Background and motivation for these protocols is as follows. To send a password in the clear, or a static function of it, is obviously insecure: an attacker can replay this information. The idea behind so-called challenge-response techniques is that challenges are never reused and that the response depends both on the password and on the challenge. Thus responses can not be replayed. Nevertheless many password based challenge-response protocols are vulnerable to dictionary attacks. This occurs when an attacker captures the messages exchanged during a legitimate run of the protocol and uses that information to verify a series of guessed passwords taken from a precompiled dictionary of common passwords. This works because users often choose simple, easy-to-remember passwords, which invariably are also easy to guess.

4.17 IPSec/IKE

IP Security Protocol

Protocol **IKE (36)**, [81], based on ISAKMP, Oakley, and SKEME ([115, 140, 102]) is a quite complex protocol with two different phases. It is a combination of 13 different subprotocols: in Phase I there are 8 different subprotocols, with 2 choices for mode, and 4 choices for authentication mechanism; in Phase II there are 4 different subprotocols with the choice of perfect forward secrecy or not and the inclusion of explicit identity information or not; lastly, there is a new group subprotocol. There are also variants depending on the types of credentials to be used.

In Phase I, the two peers establish a secure channel for further communication by negotiating ISAKMP SAs. In Phase II, protected by the SA negotiated in Phase I, the peers negotiate SAs that can be used to protect real

communication; that is, the IPsec SA. IKE defines two Phase I modes: main mode gives authenticated key exchange with identity protection. Aggressive mode gives quicker authenticated key exchange without identity protection. For Phase I, IKE defines (for main and aggressive modes) four different authentication methods: 1. authentication with digital signatures; 2. authentication with public key encryption; 3. authentication with a revised mode of public key encryption; and 4. authentication with a pre-shared key.

IKE is perceived as being too complex and it is believed that it is only a matter of time before more analyses show more serious security issues and problems become apparent. Contradictions between the different documents defining IKE have become evident (In particular the rekeying issue: multiple SAs may be pre-negotiated and used at will?) Also possible interaction of the different subprotocols, which use similar formats, could lead to new insecurities. (This has been shown to be the case for an early version of SSL, draft-benaloh-pct-00.txt, 1995)

Protocol IKE should provide Fresh Key Agreement, PFS, DoS Resilience, secure Negotiation, and ID Protection (Eavesdropper and Peer) (G1-3,7,9-15).

Protocol **IKEv2 (37)**, the Internet Key Exchange Version 2 (IKEv2) Protocol, [95] consists of two exchanges:

The first is an authentication and key exchange protocol which establishes an IKE-SA.

The second one consists of messages and payloads which focus on the negotiation of parameters in order to establish IPsec security associations (i.e., Child-SAs). These payloads contain algorithm parameters and traffic selector fields.

In addition to the above-mentioned parts IKEv2 also includes some payloads and messages which allow configuration parameters to be exchanged primarily for remote access scenarios.

Protocol IKEv2 should provide Fresh Key Agreement, PFS, DoS Resilience, secure Negotiation (G1-3,7,9-11,15).

Protocol **KINK (38)**, Kerberized Internet Negotiation of Keys, (defined in [182] and [181]) is a command/response protocol which can create, delete and maintain IPsec security associations ([98]), as an alternative to IKE ([81]). Participating nodes use Kerberos ([101]) for key management, mutual authentication, and replay protection. KINK's design mitigates denial of service attacks by requiring authenticated exchanges before the use of any public key operations and the installation of any state.

Protocol KINK should provide Fresh Key Agreement, 3P-Authorization, and ID Protection (Eavesdropper and Peer) (G1-3,6,7,10,12-14).

In the Ipsra (IP Security Remote Access) scenario (remote users that use

personal portable computing devices, or who use Internet kiosks to access private networks on the other side of an IPSEC gateway), [145] (together with the authentication for DHCP messages, [56]) defines Protocol **DHCP-IPSec-tunnel (39)**, an exchange that is secured using IPsec, and as a result the DHCP packets flowing between the remote host and the security gateway are authenticated and integrity protected.

Protocol DHCP-IPSec-tunnel should provide Authentication and Secrecy (G1,2,12).

4.18 IPv6 (including cga and HIP)

IP Version 6

Protocol **IPv6-RADIUS (40)**, defined in [7], uses RADIUS for the purposes of authentication, authorization and accounting in IPv6-enabled networks. Known security vulnerabilities of the RADIUS protocol are described in [76, 162, 161].

Protocol IPv6-RADIUS should provide Fresh Key Agreement and 3P-Authorization (G1-3,6,7,10,12).

Protocol **IPv6-cga (41)** is intended to solve one of the most difficult security related problems that arose during the design of IPv6, know as the *address ownership problem.* Since a node is able to autoconfigure its own IPv6 address (see [183]), the question is: "how does a node prove that it is allowed to use this IP address, and that it does not belong for instance to another node?" Cryptographically generated addresses (CGA) are IPv6 addresses where the interface identifier is generated by hashing the address owner's public key. The address owner can then use the corresponding private key to assert address ownership and to sign messages sent from the address without any additional security infrastructure, [22] (for more details and variants, see also [23] and [103]).

Protocol IPv6-cga should provide Sender Invariance (G16).

A similar problem is the Secure Neighbor Discovery Problem. [136] discusses the IPv6 Neighbor Discovery trust models and threats. The original solution to the problem was given in [128]. Protocol **send-cga (42)** is a protocol that uses Cryptographically Generated Addresses to secure the Neighbor Discovery for IP Version 6 (IPv6). Three variants of the send-cga have been proposed: [137] (cga header), [24], and [97] (Securing IPv6 Neighbor Discovery Using Address Based Keys (ABKs)).

Protocol send-cga should provide Sender Invariance (G16).

Protocol **HIP (43)** Host Identity Protocol (see [124, 123, 135]) proposes a solution for separating the end-point identifier and locator roles of IP addresses. It introduces a new Host Identity (HI) name space, based on public

keys. The public keys are typically, but not necessarily, self generated, as in the cga approach.

The HIP protocol permits IPv6 and IPv4 hosts to identify each other based on the public keys, to establish a pair of host-to-host ESP security associations using these public keys, and to run both IPv4 and IPv6 applications side-by-side independent of the underlying type of connectivity. It also allows many IPv4 applications to communicate directly with IPv6 applications, and vice versa.

Protocol HIP should provide Fresh Key Agreement, PFS, DoS Resilience (G1-3,7,9,10,12,15).

Protocol **pbk (44)**, Purpose-Built Keys, proposed in [41], is a two party protocol, played by a sender, Alice, and a receiver, Bob. Alice and Bob have no direct or indirect security relationship that they can use to perform an authenticated key agreement. This basically means that Alice and Bob do not both have secure channels to a common trusted party, that they do not both have access to a common security infrastructure, like a global PKI, and moreover, that they do not share a common shared secret and they have no knowledge of the correct binding of public keys to their respective identities. In this situation, any communication between Alice and Bob may be manipulated by an active attacker without Alice or Bob noticing it. Now let us suppose that at the beginning of an association between two parties an initial transaction has not been tampered with. In this case, future transactions can be secured, providing Bob with assurance that the source is the same one that started the communication, although the actual identity of Alice is not important to Bob.

The PBK framework may be explained as follows: Before Alice initiates sending a set of packets to Bob, Alice constructs a public/private key pair $PBK = (p, s)$ for use later while sending packets. This is known as a Purpose Built Key Pair. Alice then creates a Purpose Built ID $PBID$ by performing a cryptographic hash of p, the public part of PBK. This $PBID$ will be used as a pseudonymous identity for Alice. The value $PBID$ is sent along with the initial packets and each packet is signed using s, the private part of the PBK. At some point or another (we may assume, without loss of generality, that this happens also during the initial set up of the conversation) Alice also discloses the public key p to Bob. Bob is able to verify that the hash of p is $PBID$ and that the messages are signed with the secret key corresponding to p. Thus Bob is assured that the messages were sent by the same node that started the conversation. If replay protection is necessary, a nonce value (a monotonically increasing value) or time-stamp may be included with the message itself.

With luck, that is, if no active attacker tampered with the first exchanges

of the communication, $PBID$ is indeed the Purpose Built Identity created by Alice. Otherwise, $PBID$ was constructed by an attacker, in which case he may play a man in the middle attack.

Protocol pbk provide Sender Invariance.

4.19 Kerberos

Kerberos WG

[101, 130] define the well-known Kerberos Protocol (v5). The protocol has many subprotocols, including: Protocol **krb-core (45)** the core Kerberos protocol, Protocol **krb-renew (46)**, which uses renewable tokens, Protocol **krb-forward (47)**, using forwardable tokens, and Protocol **krb-cross-realm (48)**, allowing cross-realm authentication.

Protocol **bootstrap-krb (49)**, defined in [190], is a mechanism to obtain a Kerberos Ticket Granting Ticket based on a successful AAA authentication and key agreement message exchange. Such a AAA exchange is likely to be executed as part of a network access procedure. This proposal therefore allows Kerberos to be used within a local network without relying on a global Kerberos infrastructure and should allow an incremental deployment of Kerberos and in general a wider distribution of Kerberos into mobile environments without requiring a global Kerberos infrastructure.

Kerberos Set/Change Password (Version 2), Protocol **krb-password (50)**, is defined in [178, 185].

Protocol **krb-securecard (51)**, defined in [129], integrates Single-use Authentication Mechanisms based on the SecureCard within Kerberos.

Each one of those seven protocols, (Protocol krb-core, Protocol krb-renew, Protocol krb-forward, Protocol krb-cross-realm, Protocol bootstrap-krb, Protocol krb-password, and Protocol krb-securecard) should provide Fresh Key Agreement and 3P-Authorization (G1-3,6,7,10,12).

4.20 MSEC

Multicast Security

This is the only multicast protocol in our list; the security goals to be verified are Implicit Destination Authentication and Source Authentication.

The purpose of the MSEC WG is to standardize protocols for securing group communication over the Internet, initially focusing on scalable solutions for groups with a single source and a very large number of recipients.

Protocol **TESLA (52)**, the Multicast Source Authentication Transform, is defined in [148, 46]. It is a secure source authentication mechanism for multicast or broadcast data streams, for example for audio and video Internet

broadcasts, or data distribution by satellite. TESLA provides authentication of individual data packets, regardless of the packet loss rate. The symmetric MAC authentication used in unicast communication is not secure in a broadcast setting: every receiver knows the MAC key, and hence could impersonate the sender and forge messages to other receivers. TESLA uses mainly symmetric cryptography, and uses time delayed key disclosure requiring loosely synchronized clocks between the sender and the receivers.

Protocol TESLA should provide Implicit Destination Authentication and Source Authentication (G4,5).

4.21 NAS

Network Access

Protocol for carrying Authentication for Network Access

Protocol **pana (53)** is defined in [67, 210, 144, 139]

Protocol **pana-bootstrap (54)** is defined in [186]

Protocol pana and Protocol pana-bootstrap should provide Fresh Key Agreement and 3P-Authorization (G1-3,6,7,10,12).

Protocol **ChapV2 (55)**, the Microsoft PPP CHAP Extensions, Version 2, is defined in [216].

Protocol ChapV2 should provide Authentication and Secrecy (G1,2,12).

4.22 SACRED

Securely Available Credentials

Protocol **sacred (56)** is defined in [74, 66, 17]. Using this protocol, the user is able to download or obtain some secrets (keys and other credentials) from a key server.

Protocol sacred should provide Authentication and Secrecy (G1,2,12).

4.23 SECSH and Telnet

Secure Shell and Telnet

Protocol **SSH (57)** (Secure Shell) is defined in [213, 212, 68, 70, 211]

Protocol SSH should provide Fresh Key Agreement and secure Negotiation (G1-3,7,10-12).

The Telnet Authentication Option ([195]) is a general framework for adding authentication and encryption to the telnet protocol, including a generic method for negotiating an authentication type and mode, whether encryption should be used and if credentials should be forwarded. The single

authentication mechanisms are: Protocol **telnet-kermit (58)**, Telnet Kermit, defined in [11], Protocol **telnet-krb (59)**, Telnet Kerberos Version 5, defined in [193], Protocol **telnet-srp (60)**, Telnet SRP, defined in [207], and Protocol **telnet-data-encr (61)**, Telnet Data Encryption Option, defined in [194].

Protocol telnet-kermit, Protocol telnet-krb, Protocol telnet-srp, and Protocol telnet-data-encr should provide Authentication and Secrecy (G1,2,12).

4.24 STIME

Secure Network Time Protocol

Protocol **stime-ntpauth (62)**, defined in [120], uses public-key cryptography for the Network Time Protocol Version 2, which is used to securely obtain time from authenticated sources. Secure Network Time is becoming a key factor in security and non-repudiation. Existing approaches to distributing time are vulnerable to external attack and tampering, as these do not take advantage of advances in public key infrastructure and cryptographic methods, and require distribution of cryptographic keys via non-scalable out-of-band means. Securing time distribution using PKI mechanisms allows the process to scale and minimizes risk.

Protocol **TSP (63)**, the Time-Stamp Protocol, defined in [9] (Internet X.509 Public Key Infrastructure Time-Stamp Protocol, TSP) is in the expanded scope of the IETF PKIX Working Group.

Protocol stime-ntpauth and Protocol TSP should provide Authentication and Replay Protection (G1,2,3).

4.25 TLS

Transport Layer Security

Protocol **TLS (64)**, Transport Layer Security (TLS) Protocol Version 1.0, was specified in 1999 in [54], based on SSL version 3.0.

Protocol **TLS-v1.1 (65)**, TLS Protocol Version 1.1, is discussed in [55].

Protocol **TLS-SRP (66)**, TLS Protocol using SRP, is discussed in [180].

Protocol **tls-sharedkeys (67)**, TLS Protocol using shared keys,, is discussed in [75].

Protocol TLS, Protocol TLS-v1.1, Protocol TLS-SRP, and Protocol tls-sharedkeys should provide Fresh Key Agreement, secure Negotiation, and ID Protection (Eavesdropper) (G1-3,7,10-13).

5 e-Business

5.1 Payment

Protocol **ASW (68)** is defined in [19].

Protocol **PaymentUMTS (69)** is defined in [84].

Protocol **FairZG (70)** is defined in [215].

Protocol ASW, Protocol PaymentUMTS, and Protocol FairZG should provide Non-Repudiation (Accountability, Proof of Origin, and Proof of Delivery) (G1-3,13,17-19).

5.2 Electronic Commerce

SET (Secure Electronic Transaction), Protocol **SET (71)**, defined in [114], is an immense e-commerce technical standard for the commerce industry developed by Visa and MasterCard as a way to facilitate secure payment card transactions over the Internet using Digital Certificates.

Protocol **EDI (72)**, the Electronic Data Interchange, is a set of protocols for conducting structured inter-organization exchanges, such as for making purchases or initiating loan requests. [50] (MIME Encapsulation of EDI Objects) defined the method for packaging the EDI transactions sets in a MIME envelope. The RFC MIME-based Secure Peer-to-Peer Business Data Interchange over the Internet ([80]) introduces the notion of non-repudiation of receipt (NRR), used to notify a sending trading partner that requested the signed receipt that: The receiving trading partner acknowledges receipt of the sent EDI Interchange, (Problem: Non-Repudiation), the receiving trading partner has authenticated the sender of the EDI Interchange (Problem: Entity Authentication), and the receiving trading partner has verified the integrity of the sent EDI Interchange (Problem: Message Authentication).

Protocol **TRADE (73)**, the Internet Open Trading Protocol is an interoperable framework for Internet commerce. It is optimized for the case where the buyer and the merchant do not have a prior acquaintance and is payment system independent. It can encapsulate and support payment systems such as SET, Mondex, secure channel card payment, GeldKarte, etc. IOTP is able to handle cases where such merchant roles as the shopping site, the payment handler, the deliverer of goods or services, and the provider of customer support are performed by different Internet sites.

Protocol SET, Protocol EDI, and Protocol TRADE should provide Non-Repudiation (Accountability, Proof of Origin, and Proof of Delivery) (G1-3, 13,17-19). (Plus other standard Authentication goals).

6 Non IETF Protocols

6.1 3GPP

Protocol **AKA (74)** is defined in [1].

Protocol AKA should provide Fresh Key Agreement and 3P-Authorization (G1-3,6,7,10,12).

6.2 IEEE 802.11

The IEEE has developed a new procedure to provide confidentiality of user information being transferred over a wireless LAN (WLAN) and authentication of IEEE 802.11 conformant devices. The current version of Protocol **IEEE 802.1X (75)** is [206]. It is not clear if we can use this protocol in AVISPA, as it is not openly available to the public.

Protocol IEEE 802.1X should provide Fresh Key Agreement and 3P-Authorization (G1-3,6,7,10,12).

Related IETF drafts are [138, 47, 214, 179, 83, 141, 142, 143, 29, 2].

The security services provided by the protolcol are: Authentication (including Data origin authenticity and Replay detection), Confidentiality and Key Management.

6.3 LAP

Liberty Alliance Project

Usually each Web Service requires its own sign-on procedure where each user must present his own user name and password. This results in a cumbersome user experience. What users would like to be able to do is to sign-on only once to a set of related services and then freely move between them, although they may be located on multiple servers or even in different domains. This is known as Single-Sign-On (SSO). LAP tries to provide a SSO solution by specifying a simple but secure mechanism for "federating" identities (a system for binding multiple accounts for a given user).

One of the core pieces in this architecture is the Protocol **lap-lecp (76)** presented in [107] (for further discussion, see also [156]).

Protocol lap-lecp should provide Fresh Key Agreement, 3P-Authorization, and ID Protection (Eavesdropper and Peer) (G1-3,6,7,10,12-14).

6.4 ISO/IEC

ISO has proposed Protocol **ISO/IEC 9798-3 (77)**, a Public Key Protocol without Trusted Third Party, in four different versions. The protocol is defined in [88].

The IETF SASL variant is defined in [217].

Protocol ISO/IEC 9798-3 should provide Authentication (G1,2).

6.5 2pS

In a two-party signature (2pS) scheme, a client and server, each holding a share of decryption key material, collaborate to compute a signature under some parameters known to both. The RSA version of this scheme, Protocol **2pRSA (78)** is proposed in [31].

Protocol 2pRSA should provide Authentication and Replay Protection (G1, 2,3).

6.6 LPD

The survey article [39] discusses several well-known key establishment protocols for mobile communications. We will choose one or several of them as our Protocol **MutAuthLPD (79)** a protocol for mutual authentication for low-powered devices.

Protocol MutAuthLPD should provide Authentication, Secrecy and Replay Protection (G1,2,3,12).

7 Summary of Protocols and Goals

The following table summarizes the proposed goals (properties) for all protocols in our list. As mentioned before, the required security properties are often not explicitly stated in the IETF documents. Thus, this table may be revised after further feedback from IETF representatives and authors.

Protocol	(G1) Entity authentication	(G2) Message authentication	(G3) Replay Protection	(G4) Implicit Dest. Authn	(G5) Source Authentication	(G6) Authorization (by T3P)	(G7) Key authentication	(G8) Key confirmation	(G9) Perfect Forward Secrecy	(G10) Fresh Key Derivation	(G11) Secure capabilities negot.	(G12) Confidentiality	(G13) ID Protection (Eavesdr.)	(G14) ID Protection (Peer)	(G15) Limited DoS Resistance	(G16) Sender Invariance	(G17) Accountability	(G18) Proof of Origin	(G19) Proof of Delivery	(G20) Temporal Property
AAA-MIP (1)	×	×	×				×	×			×									
MIP-BU (2)																×	×			
seamoby-ctp (3)	×	×	×				×	×												×
SIP-Digest (4)	×	×																		
SIP-SMIME (5)	×	×											×							
H530 (6)	×	×	×				×	×			×									
RSVP-sec (7)	×	×																		
NSIS-acc (8)	×	×											×	×	×					×
Geopriv-nym (9)	×	×												×	×					
Impp (10)	×	×	×	×									×							
Simple (11)	×	×	×	×									×							
aaa-nasreq (12)	×	×	×				×	×			×									
SPKM-LIPKEY (13)	×	×	×					×			×									
CRAM-MD5 (14)	×	×	×										×							
APOP (15)	×	×	×										×							
DHCP-delayed (16)	×	×	×										×							
DNSSEC (17)	×	×	×																	
TSIG (18)	×	×	×					×												
SIG(0) (19)	×	×	×					×												
PEAP (20)	×	×	×				×	×			×			×						
EAP-SIM (21)	×	×	×				×	×			×									
EAP-AKA (22)	×	×	×				×	×			×			×	×					
EAP-Archie (23)	×	×	×				×	×			×					×				
EAP-IKEv2 (24)	×	×	×				×	×			×					×				
EAP-TTLS (25)	×	×	×				×	×			×									

Protocol	(G1) Entity authentication	(G2) Message authentication	(G3) Replay Protection	(G4) Implicit Dest. Authn	(G5) Source Authentication	(G6) Authorization (by T3P)	(G7) Key authentication	(G8) Key confirmation	(G9) Perfect Forward Secrecy	(G10) Fresh Key Derivation	(G11) Secure capabilities negot.	(G12) Confidentiality	(G13) ID protection (Eavesdr.)	(G14) ID protection (Peer)	(G15) Limited DoS Resistance	(G16) Sender Invariance	(G17) Accountability	(G18) Proof of Origin	(G19) Proof of Delivery	(G20) Temporal Property
OTP (26)	×	×										×								
S/Key (27)	×	×										×								
SecureID (28)	×	×										×								
S-BGP (29)	×	×				×														×
soBGP (30)	×	×				×														×
EKE (31)	×	×										×								
EKE2 (32)	×	×										×								
SPEKE (33)	×	×										×								
A-EKE (34)	×	×										×								
SRP (35)	×	×										×								
IKE (36)	×	×	×				×		×	×	×		×	×	×					
IKEv2 (37)	×	×	×				×		×	×	×					×				
KINK (38)	×	×	×			×	×			×			×	×						
DHCP-IPSec-tunnel (39)	×	×										×								
IPv6-RADIUS (40)	×	×	×			×	×			×										
IPv6-cga (41)																	×			
send-cga (42)																	×			
HIP (43)	×	×	×				×		×	×						×				
pbk (44)																	×			
krb-core (45)	×	×	×			×	×			×										
krb-renew (46)	×	×	×			×	×			×										
krb-forward (47)	×	×	×			×	×			×										
krb-cross-realm (48)	×	×	×			×	×			×										
bootstrap-krb (49)	×	×	×			×	×			×										
krb-password (50)	×	×	×			×	×			×										

Protocol	(G1) Entity authentication	(G2) Message authentication	(G3) Replay Protection	(G4) Implicit Dest. Authn	(G5) Source Authentication	(G6) Authorization (by T3P)	(G7) Key authentication	(G8) Key confirmation	(G9) Perfect Forward Secrecy	(G10) Fresh Key Derivation	(G11) Secure capabilities negot.	(G12) Confidentiality	(G13) ID Protection (Eavesdr.)	(G14) ID Protection (Peer)	(G15) Limited DoS Resistance	(G16) Sender Invariance	(G17) Accountability	(G18) Proof of Origin	(G19) Proof of Delivery	(G20) Temporal Property
krb-securecard (51)	×	×	×				×	×			×									
TESLA (52)				×	×															
pana (53)	×	×	×				×	×			×									
pana-bootstrap (54)	×	×	×				×	×			×									
ChapV2 (55)	×	×											×							
sacred (56)	×	×											×							
SSH (57)	×	×	×					×			×	×								
telnet-kermit (58)	×	×											×							
telnet-krb (59)	×	×											×							
telnet-srp (60)	×	×											×							
telnet-data-encr (61)	×	×											×							
stime-ntpauth (62)	×	×	×																	
TSP (63)	×	×	×																	
TLS (64)	×	×	×					×			×	×		×						
TLS-v1.1 (65)	×	×	×					×			×	×		×						
TLS-SRP (66)	×	×	×					×			×	×		×						
tls-sharedkeys (67)	×	×	×					×			×	×		×						
ASW (68)	×	×	×											×				×	×	×
PaymentUMTS (69)	×	×	×											×				×	×	×
FairZG (70)	×	×	×											×				×	×	×
SET (71)	×	×	×											×				×	×	×
EDI (72)	×	×	×											×				×	×	×
TRADE (73)	×	×	×											×				×	×	×
AKA (74)	×	×	×				×	×			×									
IEEE 802.1X (75)	×	×	×				×	×			×									

Protocol	(G1) Entity authentication	(G2) Message authentication	(G3) Replay Protection	(G4) Implicit Dest. Authn	(G5) Source Authentication	(G6) Authorization (by T3P)	(G7) Key authentication	(G8) Key confirmation	(G9) Perfect Forward Secrecy	(G10) Fresh Key Derivation	(G11) Secure capabilities negot.	(G12) Confidentiality	(G13) ID protection (Eavesdr.)	(G14) ID protection (Peer)	(G15) Limited DoS Resistance	(G16) Sender Invariance	(G17) Accountability	(G18) Proof of Origin	(G19) Proof of Delivery	(G20) Temporal Property
lap-lecp (76)	×	×	×			×	×			×				×						
ISO/IEC 9798-3 (77)	×	×																		
2pRSA (78)	×	×	×																	
MutAuthLPD (79)	×	×	×										×							

References

[1] 3GPP (3rd Generation Partnership Project). Ts 33.102 v5.1.0: Technical specification group services and system aspects; 3g security; security architecture (release 5), Dec. 2002.

[2] B. Aboba. A Model for Context Transfer in IEEE 802, Oct. 2003. Work in Progress.

[3] B. Aboba. EAP Key Management Framework, Oct. 2003. Work in Progress.

[4] B. Aboba, P. Calhoun, S. Glass, T. Hiller, P. McCann, H. Shiino, G. Zorn, G. Dommety, C. Perkins, B. Patil, D. Mitton, S. Manning, M. Beadles, P. Walsh, X. Chen, S. Sivalingham, A. Hameed, M. Munson, S. Jacobs, B. Lim, B. Hirschman, R. Hsu, Y. Xu, E. Campbell, S. Baba, and E. Jaques. RFC 2989: Criteria for Evaluating AAA Protocols for Network Access, Nov. 2000. Status: Informational.

[5] B. Aboba and D. Simon. RFC 2716: PPP EAP TLS Authentication Protocol, Oct. 1999. Status: Experimental.

[6] B. Aboba and J. Wood. RFC 3539: Authentication, Authorization and Accounting (AAA) Transport Profile, June 2003. Status: Proposed Standard.

[7] B. Aboba, G. Zorn, and D. Mitton. RFC 3162: RADIUS and IPv6, Aug. 2001. Status: Proposed Standard.

[8] C. Adams. RFC 2025: The Simple Public-Key GSS-API Mechanism (SPKM), Oct. 1996. Status: Proposed Standard.

[9] C. Adams, P. Cain, D. Pinkas, and R. Zuccherato. RFC 3161: Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP), Aug. 2001. Status: Proposed Standard.

[10] C. Adams and S. Farrell. RFC 2510: Internet X.509 Public Key Infrastructure Certificate Management Protocols, Mar. 1999. Status: Proposed Standard.

[11] J. Altman and F. da Cruz. RFC 2840: TELNET KERMIT OPTION, May 2000. Status: Informational.

[12] R. Arends. Protocol Modifications for the DNS Security Extensions, Oct. 2003. Work in Progress.

[13] R. Arends. Resource Records for DNS Security Extensions, Oct. 2003. Work in Progress.

[14] R. Arends, R. Austein, D. Massey, M. Larson, and S. Rose. DNS Security Introduction and Requirements, Oct. 2003. Work in Progress.

[15] J. Arkko and H. Haverinen. EAP AKA Authentication, Oct. 2003. Work in Progress.

[16] J. Arkko, V. Torvinen, G. Camarillo, A. Niemi, and T. Haukka. RFC 3329: Security Mechanism Agreement for the Session Initiation Protocol (SIP), Jan. 2003. Status: Proposed Standard.

[17] A. Arsenault and S. Farrell. RFC 3157: Securely Available Credentials - Requirements, Aug. 2001. Status: Informational.

[18] N. Asokan, V. Niemi, and K. Nyberg. Man-in-the-middle in tunnelled authentication. In *Proceedings of the 11th International Workshop on Security Protocols*, LNCS, Cambridge, UK, Apr. 2003. To be published by Springer-Verlag.

[19] N. Asokan, V. Shoup, and M. Waidner. Asynchronous protocols for optimistic fair exchange. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 86–99, 1998.

[20] D. Atkins and R. Austein. Threat Analysis Of The Domain Name System, Oct. 2003. Work in Progress.

[21] D. Atkins and G. Klyne. Common Presence and Instant Messaging: Message Format, Jan. 2003. Work in Progress.

[22] T. Aura. Cryptographically Generated Addresses (CGA), Feb. 2003. Work in Progress.

[23] T. Aura. Cryptographically generated addresses (CGA). In *Proc. 6th Information Security Conference (ISC'03)*, LNCS, Bristol, UK, Oct. 2003. Springer.

[24] T. Aura. Cryptographically Generated Addresses (CGA), Oct. 2003. Work in Progress.

[25] T. Aura, M. Roe, and J. Arkko. Security of internet location management. In *Proc. 18th Annual Computer Security Applications Conference*, pages 78–87, Las Vegas, NV USA, Dec. 2002. IEEE Press.

[26] F. Baker, B. Lindell, and M. Talwar. RFC 2747: RSVP Cryptographic Authentication, Jan. 2000. Status: Proposed Standard.

[27] M. Barnes. A Mechanism to Secure SIP Identity headers inserted by Intermediaries, Oct. 2003. Work in Progress.

[28] M. Beadles and D. Mitton. RFC 3169: Criteria for Evaluating Network Access Server Protocols, Sept. 2001. Status: Informational.

[29] L. Bell, D. Romascanu, and B. Aboba. History of the IEEE 802/IETF Relationship, June 2003. Work in Progress.

[30] M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated key exchange secure against dictionary attacks. *Lecture Notes in Computer Science*, 1807:139ff, 2000.

[31] M. Bellare and R. Sandhu. The security of a family of two-party RSA signature schemes. Technical Report 2001/060, 2001.

[32] S. Bellovin. RFC 2316: Report of the IAB Security Architecture Workshop, Apr. 1998. Status: Informational.

[33] S. Bellovin, C. Kaufman, and J. Schiller. Security Mechanisms for the Internet, July 2003. Work in Progress.

[34] S. Bellovin and M. Merritt. Encrypted Key Exchange: Password-based protocols secure against dictionary attacks. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, May 1992.

[35] S. Bellovin and M. Merritt. Augmented encrypted key exchange: a password-based protocol secure against dictionary attacks and password file compromise, 1993.

[36] L. Berger and T. O'Malley. RFC 2207: RSVP Extensions for IPSEC Data Flows, Sept. 1997. Status: Proposed Standard.

[37] L. Blunk. Extensible Authentication Protocol (EAP), Sept. 2003. Work in Progress.

[38] L. Blunk and J. Vollbrecht. RFC 2284: PPP Extensible Authentication Protocol (EAP), Mar. 1998. Status: Proposed Standard.

[39] C. Boyd and A. Mathuria. Key establishment protocols for secure mobile communications: A selective survey. *Lecture Notes in Computer Science*, 1438:344ff, 1998.

[40] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. RFC 2205: Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification, Sept. 1997. Status: Proposed Standard.

[41] S. Bradner, A. Mankin, and J. Schiller. A Framework for Purpose-Built Keys (PBK), June 2003. Work in Progress.

[42] M. Brunner. Requirements for Signaling Protocols, Aug. 2003. Work in Progress.

[43] P. Calhoun, T. Johansson, and C. Perkins. Diameter Mobile IPv4 Application, Oct. 2003. Work in Progress.

[44] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko. RFC 3588: Diameter Base Protocol, Sept. 2003. Status: Proposed Standard.

[45] P. Calhoun, G. Zorn, D. Spence, and D. Mitton. Diameter Network Access Server Application, Oct. 2003. Work in Progress.

[46] R. Canetti, A. Perrig, and B. Whillock. TESLA: Multicast Source Authentication Transform Specification, Oct. 2002. Work in Progress.

[47] P. Congdon, B. Aboba, A. Smith, G. Zorn, and J. Roese. RFC 3580: IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines, Sept. 2003. Status: Informational.

[48] D. Conrad. RFC 3225: Indicating Resolver Support of DNSSEC, Dec. 2001. Status: Proposed Standard.

[49] M. Crispin. RFC 2060: Internet Message Access Protocol - Version 4rev1, Dec. 1996. Status: Proposed Standard.

[50] D. Crocker. RFC 1767: MIME Encapsulation of EDI Objects, Mar. 1995. Status: Proposed Standard.

[51] J. Cuellar, J. Morris, and D. Mulligan. Geopriv requirements, Oct. 2003. Work in Progress.

[52] M. Day, S. Aggarwal, G. Mohr, and J. Vincent. RFC 2779: Instant Messaging / Presence Protocol Requirements, Feb. 2000. Status: Informational.

[53] M. Day, J. Rosenberg, and H. Sugano. RFC 2778: A Model for Presence and Instant Messaging, Feb. 2000. Status: Informational.

[54] T. Dierks and C. Allen. RFC 2246: The TLS Protocol Version 1.0, Jan. 1999. Status: Proposed Standard.

[55] T. Dierks and E. Rescorla. The TLS Protocol Version 1.1, July 2003. Work in Progress.

[56] R. Droms and W. Arbaugh. RFC 3118: Authentication for DHCP Messages, June 2001. Status: Proposed Standard.

[57] D. Eastlake 3rd. RFC 2137: Secure Domain Name System Dynamic Update, Apr. 1997. Status: Proposed Standard.

[58] D. Eastlake 3rd. RFC 2535: Domain Name System Security Extensions, Mar. 1999. Status: Proposed Standard.

[59] D. Eastlake 3rd. RFC 2930: Secret Key Establishment for DNS (TKEY RR), Sept. 2000. Status: Proposed Standard.

[60] D. Eastlake 3rd. RFC 2931: DNS Request and Transaction Signatures (SIG(0)s), Sept. 2000. Status: Proposed Standard.

[61] M. Eisler. RFC 2847: LIPKEY - A Low Infrastructure Public Key Mechanism Using SPKM, June 2000. Status: Proposed Standard.

[62] J. Elwell. User identification in a SIP/QSIG environment, May 2003. Work in Progress.

[63] R. Elz and R. Bush. RFC 2181: Clarifications to the DNS Specification, July 1997. Status: Proposed Standard.

[64] P. Eronen, T. Hiller, and G. Zorn. Diameter Extensible Authentication Protocol (EAP) Application, Oct. 2003. Work in Progress.

[65] M. D. et al. Threat Analysis of the geopriv Protocol, Sept. 2003. Accepted as an Informational RFC.

[66] S. Farrell. Securely Available Credentials Protocol, June 2003. Work in Progress.

[67] D. Forsberg. Protocol for Carrying Authentication for Network Access (PANA), Oct. 2003. Work in Progress.

[68] M. Forssen and F. Cusack. Generic Message Exchange Authentication For SSH, Apr. 2003. Work in Progress.

[69] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart. RFC 2617: HTTP Authentication: Basic and Digest Access Authentication, June 1999. Status: Draft Standard.

[70] M. Friedl, N. Provos, and W. Simpson. Diffie-Hellman Group Exchange for the SSH Transport Layer Protocol, July 2003. Work in Progress.

[71] P. Funk and S. Blake-Wilson. EAP Tunneled TLS Authentication Protocol (EAP-TTLS), Aug. 2003. Work in Progress.

[72] J. Galvin, S. Murphy, S. Crocker, and N. Freed. RFC 1847: Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted, Oct. 1995. Status: Proposed Standard.

[73] O. Gudmundsson. RFC 3226: DNSSEC and IPv6 A6 aware server/resolver message size requirements, Dec. 2001. Status: Proposed Standard.

[74] D. Gustafson, M. Just, and M. Nystrom. Securely Available Credentials - Credential Server Framework, June 2003. Work in Progress.

[75] P. Gutmann. Use of Shared Keys in the TLS Protocol, Oct. 2003. Work in Progress.

[76] E. Guttman, C. Perkins, J. Veizades, and M. Day. RFC 2608: Service Location Protocol, Version 2, June 1999. Status: Proposed Standard.

[77] N. Haller. RFC 1760: The S/KEY One-Time Password System, Feb. 1995. Status: Informational.

[78] N. Haller, C. Metz, P. Nesser, and M. Straw. RFC 2289: A One-Time Password System, Feb. 1998. Status: Standard.

[79] R. Hancock. Next Steps in Signaling: Framework, Oct. 2003. Work in Progress.

[80] T. Harding, R. Drummond, and C. Shih. RFC 3335: MIME-based Secure Peer-to-Peer Business Data Interchange over the Internet, Sept. 2002. Status: Proposed Standard.

[81] D. Harkins and D. Carrel. RFC 2409: The Internet Key Exchange (IKE), Nov. 1998. Status: Proposed Standard.

[82] H. Haverinen and J. Salowey. EAP SIM Authentication, Oct. 2003. Work in Progress.

[83] Y.-G. Hong. Considerations of FMIPv6 in 802.11 networks, June 2003. Work in Progress.

[84] G. Horn and B. Preneel. Authentication and payment in future mobile systems. In *ESORICS: European Symposium on Research in Computer Security*. LNCS, Springer-Verlag, 1998.

[85] R. Housley. RFC 3369: Cryptographic Message Syntax (CMS), Aug. 2002. Status: Proposed Standard.

[86] R. Housley. RFC 3370: Cryptographic Message Syntax (CMS) Algorithms, Aug. 2002. Status: Proposed Standard.

[87] ISO/IEC. ISO/IEC 7812-2: Information Processing Systems, Identification cards – Identification of Issuers, Part 2: Application and Registration Procedures , 1993.

[88] ISO/IEC. ISO/IEC 9798-3: Information technology - Security techniques - Entity authentication - Part 3: Mechanisms using digital signature techniques, 1997.

[89] ITU. ITU H.530: Symmetric security procedures for H.323 mobility in H.510, Oct. 2002. Available through http://www.itu.int/ITU-T/studygroups/com16/index.html.

[90] ITU. ITU H.530 Corrigendum 1: Symmetric security procedures for H.323 mobility in H.510, July 2003. Available through http://www.itu.int/ITU-T/studygroups/com16/index.html.

[91] D. P. Jablon. Strong password-only authenticated key exchange. *Computer Communication Review*, 26(5):5–26, 1996.

[92] C. Jennings, J. Peterson, and M. Watson. RFC 3325: Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks, Nov. 2002. Status: Informational.

[93] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6, July 2003. Work in Progress.

[94] S. Josefsson, A. Palekar, D. Simon, and G. Zorn. Protected EAP Protocol (PEAP), Oct. 2003. Work in Progress.

[95] C. Kaufman. Internet Key Exchange (IKEv2) Protocol, Oct. 2003. Work in Progress.

[96] J. Kempf. RFC 3374: Problem Description: Reasons For Performing Context Transfers Between Nodes in an IP Access Network, Sept. 2002. Status: Informational.

[97] J. Kempf. Securing IPv6 Neighbor Discovery Using Address Based Keys (ABKs), May 2003. Work in Progress.

[98] S. Kent and R. Atkinson. RFC 2401: Security Architecture for the Internet Protocol, Nov. 1998. Status: Proposed Standard.

[99] J. Klensin. RFC 3467: Role of the Domain Name System (DNS), Feb. 2003. Status: Informational.

[100] J. Klensin, R. Catoe, and P. Krumviede. RFC 2195: IMAP/POP AUTHorize Extension for Simple Challenge/Response, Sept. 1997. Status: Proposed Standard.

[101] J. Kohl and C. Neuman. RFC 1510: The Kerberos Network Authentication Service (V5), Sept. 1993. Status: Proposed Standard.

[102] H. Krawczyk. SKEME: A versatile secure key exchange mechanism for internet. In *IEEE Proceedings of the 1996 Symposium on Network and Distributed Systems Security*. IEEE, 1996.

[103] J. Laganier and G. Montenegro. Using IKE with IPv6 Cryptographically Generated Address, July 2003. Work in Progress.

[104] M. Leech. RFC 1929: Username/Password Authentication for SOCKS V5, Mar. 1996. Status: Proposed Standard.

[105] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones. RFC 1928: SOCKS Protocol Version 5, Mar. 1996. Status: Proposed Standard.

[106] E. Lewis. RFC 3090: DNS Security Extension Clarification on Zone Status, Mar. 2001. Status: Proposed Standard.

[107] L. Liberty Alliance Project. Liberty ID-FF Bindings and Profiles Specification Version: 1.2, 2003.

[108] M. Liebsch. Candidate Access Router Discovery, Sept. 2003. Work in Progress.

[109] J. Loughney. Context Transfer Protocol, Oct. 2003. Work in Progress.

[110] J. Loughney and G. Camarillo. Authentication, Authorization and Accounting Requirements for the Session Initiation Protocol, June 2003. Work in Progress.

[111] C. Lynn. Secure BGP (S-BGP), July 2003. Work in Progress.

[112] R. Mahy. Discussion of suitability: S/MIME instead of Digest Authentication in the Session Initiation Protocol (SIP), July 2003. Work in Progress.

[113] D. Massey and S. Rose. RFC 3445: Limiting the Scope of the KEY Resource Record (RR), Dec. 2002. Status: Proposed Standard.

[114] Mastercard and VISA. SET Secure Electronic Transaction Specification, May 1977.

[115] D. Maughan, M. Schertler, M. Schneider, and J. Turner. RFC 2408: Internet Security Association and Key Management Protocol (ISAKMP), Nov. 1998. Status: Proposed Standard.

[116] D. McDonald, C. Metz, and B. Phan. RFC 2367: PF_KEY Key Management API, Version 2, July 1998. Status: Informational.

[117] C. Meadows. A cost-based framework for analysis of denial of service in networks. *Journal of Computer Security*, to appear.

[118] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Oct. 1996. Fifth Printing (August 2001).

[119] C. Metz. RFC 2243: OTP Extended Responses, Nov. 1997. Status: Proposed Standard.

[120] D. Mills. Public-Key Cryptography for the Network Time Protocol Version 2, Nov. 2002. Work in Progress.

[121] P. Mockapetris. RFC 1034: Domain names - concepts and facilities, Nov. 1987. Status: Standard.

[122] P. Mockapetris. RFC 1035: Domain names - implementation and specification, Nov. 1987. Status: Standard.

[123] R. Moskowitz. Host Identity Protocol Architecture, Oct. 2003. Work in Progress.

[124] R. Moskowitz, P. Nikander, and P. Jokela. Host Identity Protocol, Oct. 2003. Work in Progress.

[125] S. Murphy. BGP Security Vulnerabilities Analysis, June 2003. Work in Progress.

[126] J. Myers. RFC 1731: IMAP4 Authentication Mechanisms, Dec. 1994. Status: Proposed Standard.

[127] J. Myers and M. Rose. RFC 1939: Post Office Protocol - Version 3, May 1996. Status: Standard.

[128] T. Narten, E. Nordmark, and W. Simpson. RFC 2461: Neighbor Discovery for IP Version 6 (IPv6), Dec. 1998. Status: Draft Standard.

[129] C. Neuman. Integrating Single-use Authentication Mechanisms with Kerberos, Oct. 2003. Work in Progress.

[130] C. Neuman. The Kerberos Network Authentication Service (V5), June 2003. Work in Progress.

[131] C. Newman. RFC 2444: The One-Time-Password SASL Mechanism, Oct. 1998. Status: Proposed Standard.

[132] C. Newman and J. G. Myers. RFC 2244: ACAP – Application Configuration Access Protocol, Nov. 1997. Status: Proposed Standard.

[133] J. Ng. Extensions to BGP to Support Secure Origin BGP (soBGP), June 2003. Work in Progress.

[134] A. Niemi, J. Arkko, and V. Torvinen. RFC 3310: Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA), Sept. 2002. Status: Informational.

[135] P. Nikander. End-Host Mobility and Multi-Homing with Host Identity Protocol, June 2003. Work in Progress.

[136] P. Nikander. IPv6 Neighbor Discovery trust models and threats, Oct. 2003. Work in Progress.

[137] P. Nikander. Secure Neighbor Discovery using separate CGA extension header, June 2003. Work in Progress.

[138] K. Norseth. Definitions for Port Access Control (IEEE 802.1X) MIB, July 2003. Work in Progress.

[139] Y. Ohba. Problem Statement and Usage Scenarios for PANA, Apr. 2003. Work in Progress.

[140] H. Orman. RFC 2412: The OAKLEY Key Determination Protocol, Nov. 1998. Status: Informational.

[141] S. Park. 802.11 Mobility Framework Supporting GPRS Handover, June 2003. Work in Progress.

[142] S. Park. IPv6 DAD Consideration for 802.11 Environment, July 2003. Work in Progress.

[143] Y.-J. Park and Y. Mun. Layer 2 Handoff for Mobile-IPv4 with 802.11, Oct. 2003. Work in Progress.

[144] M. Parthasarathy. PANA Threat Analysis and security requirements, May 2003. Work in Progress.

[145] B. Patel, B. Aboba, S. Kelly, and V. Gupta. RFC 3456: Dynamic Host Configuration Protocol (DHCPv4) Configuration of IPsec Tunnel Mode, Jan. 2003. Status: Proposed Standard.

[146] C. Perkins. Mobile IPv4 Challenge/Response Extensions (revised), Oct. 2003. Work in Progress.

[147] C. Perkins and P. Calhoun. AAA Registration Keys for Mobile IPv4, Oct. 2003. Work in Progress.

[148] A. Perrig. TESLA: Multicast Source Authentication Transform Introduction, Oct. 2002. Work in Progress.

[149] J. Peterson. RFC 3323: A Privacy Mechanism for the Session Initiation Protocol (SIP), Nov. 2002. Status: Proposed Standard.

[150] J. Peterson. Address Resolution for Instant Messaging and Presence, Oct. 2003. Work in Progress.

[151] J. Peterson. Common Profile for Instant Messaging (CPIM), Oct. 2003. Work in Progress.

[152] J. Peterson. Common Profile for Presence (CPP), Oct. 2003. Work in Progress.

[153] J. Peterson. Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP), Mar. 2003. Work in Progress.

[154] J. Peterson. Role-based Authorization Requirements for the Session Initiation Protocol, Oct. 2003. Work in Progress.

[155] J. Peterson. SIP Authenticated Identity Body (AIB) Format, July 2003. Work in Progress.

[156] B. Pfitzmann and M. Waidner. Token-based web single signon with enabled clients.

[157] J. Puthenkulam. The Compound Authentication Binding Problem, Oct. 2003. Work in Progress.

[158] Y. Rekhter. A Border Gateway Protocol 4 (BGP-4), Oct. 2003. Work in Progress.

[159] E. Rescorla. A Survey of Authentication Mechanisms, Oct. 2003. Work in Progress.

[160] E. Rescorla and B. Korver. RFC 3552: Guidelines for Writing RFC Text on Security Considerations, July 2003. Status: Best Current Practice.

[161] C. Rigney, W. Willats, and P. Calhoun. RFC 2869: RADIUS Extensions, June 2000. Status: Informational.

[162] C. Rigney, S. Willens, A. Rubens, and W. Simpson. RFC 2865: Remote Authentication Dial In User Service (RADIUS), June 2000. Status: Draft Standard.

[163] S. Rose. DNS Request and Transaction Signatures (SIG(0)s), Aug. 2003. Work in Progress.

[164] J. Rosenberg. A Presence Event Package for the Session Initiation Protocol (SIP), Jan. 2003. Work in Progress.

[165] J. Rosenberg. A Session Initiation Protocol (SIP) Event Package for Modification Events for the Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Managed Documents, June 2003. Work in Progress.

[166] J. Rosenberg. An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usage for Presence Lists, Oct. 2003. Work in Progress.

[167] J. Rosenberg. Extensible Markup Language (XML) Configuration Access Protocol (XCAP)Usages for Setting Presence Authorization, Oct. 2003. Work in Progress.

[168] J. Rosenberg. The Extensible Markup Language (XML) Configuration Access Protocol (XCAP), Oct. 2003. Work in Progress.

[169] J. Rosenberg and H. Schulzrinne. RFC 3262: Reliability of Provisional Responses in Session Initiation Protocol (SIP), June 2002. Status: Proposed Standard.

[170] J. Rosenberg and H. Schulzrinne. RFC 3263: Session Initiation Protocol (SIP): Locating SIP Servers, June 2002. Status: Proposed Standard.

[171] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. RFC 3261: SIP: Session Initiation Protocol, June 2002. Status: Proposed Standard.

[172] RSA. Secureid hardware token and software tokens, 2003. http://www.rsasecurity.com/products/securid/hardware_token.html.

[173] H. Schulzrinne. RPID – Rich Presence Information Data Format, July 2003. Work in Progress.

[174] R. Shirey. RFC 2828: Internet Security Glossary, May 2000. Status: Informational.

[175] B. Sommerfeld. Requirements for an IPsec API, June 2003. Work in Progress.

[176] M. Steiner, G. Tsudik, and M. Waidner. Refinement and extension of encrypted key exchange. *Operating Systems Review*, 29:22–30, July 1995.

[177] H. Sugano and S. Fujimoto. Presence Information Data Format (PIDF), May 2003. Work in Progress.

[178] M. Swift, J. Trostle, and J. Brezak. RFC 3244: Microsoft Windows 2000 Kerberos Change Password and Set Password Protocols, Feb. 2002. Status: Informational.

[179] P. Tan. Recommendations for Achieving Seamless IPv6 Handover in IEEE 802.11 Networks, Mar. 2003. Work in Progress.

[180] D. Taylor. Using SRP for TLS Authentication, June 2003. Work in Progress.

[181] M. Thomas. RFC 3129: Requirements for Kerberized Internet Negotiation of Keys, June 2001. Status: Informational.

[182] M. Thomas and J. Vilhuber. Kerberized Internet Negotiation of Keys (KINK), Jan. 2003. Work in Progress.

[183] S. Thomson and T. Narten. RFC 2462: IPv6 Stateless Address Autoconfiguration, Dec. 1998. Status: Draft Standard.

[184] D. Trossen, G. Krishnamurthi, H. Chaskar, and J. Kempf. Issues in candidate access router discovery for seamless IP-level handoffs, Oct. 2002. Work in Progress.

[185] J. Trostle, M. Swift, J. Brezak, and B. Gossman. Kerberos Set/Change Password: Version 2, May 2001. Work in Progress.

[186] H. Tschofenig. Bootstrapping RFC3118 Delayed authentication using PANA, Oct. 2003. Work in Progress.

[187] H. Tschofenig. QoS NSLP Authorization Issues, June 2003. Work in Progress.

[188] H. Tschofenig. RSVP Security Properties, Oct. 2003. Work in Progress.

[189] H. Tschofenig. Security Implications of the Session Identifier, June 2003. Work in Progress.

[190] H. Tschofenig and D. Atkins. Bootstrapping Kerberos, Jan. 2003. Work in Progress.

[191] H. Tschofenig and D. Kroeselberg. EAP IKEv2 Method (EAP-IKEv2), Oct. 2003. Work in Progress.

[192] H. Tschofenig and D. Kroeselberg. Security Threats for NSIS, Oct. 2003. Work in Progress.

[193] T. Ts'o. RFC 2942: Telnet Authentication: Kerberos Version 5, Sept. 2000. Status: Proposed Standard.

[194] T. Ts'o. RFC 2946: Telnet Data Encryption Option, Sept. 2000. Status: Proposed Standard.

[195] T. Ts'o and J. Altman. RFC 2941: Telnet Authentication Option, Sept. 2000. Status: Proposed Standard.

[196] S. Turner. CMS Symmetric Key Management and Distribution, Jan. 2003. Work in Progress.

[197] V. Venkataramanan. Enhancements to Asserted Identity to Enable Called Party Name Delivery using SIP, June 2003. Work in Progress.

[198] P. Vixie, O. Gudmundsson, D. Eastlake 3rd, and B. Wellington. RFC 2845: Secret Key Transaction Authentication for DNS (TSIG), May 2000. Status: Proposed Standard.

[199] J. Vollbrecht. State Machines for EAP Peer and Authenticator, Oct. 2003. Work in Progress.

[200] M. Wagle and R. Aradhaya. An Out-of-Band authentication procedure for SIP, Feb. 2003. Work in Progress.

[201] J. Walker and R. Housley. The EAP Archie Protocol, June 2003. Work in Progress.

[202] M. Watson. RFC 3324: Short Term Requirements for Network Asserted Identity, Nov. 2002. Status: Informational.

[203] B. Wellington. RFC 3007: Secure Domain Name System (DNS) Dynamic Update, Nov. 2000. Status: Proposed Standard.

[204] B. Wellington. RFC 3008: Domain Name System Security (DNSSEC) Signing Authority, Nov. 2000. Status: Proposed Standard.

[205] B. Wellington and O. Gudmundsson. Redefinition of DNS AD bit, June 2002. Work in Progress.

[206] I. . working group. Medium access control (mac) security enhancements, Sept. 2003.

[207] T. Wu. RFC 2944: Telnet Authentication: SRP, Sept. 2000. Status: Proposed Standard.

[208] T. Wu. RFC 2945: The SRP Authentication and Key Exchange System, Sept. 2000. Status: Proposed Standard.

[209] S. Yadav, R. Yavatkar, R. Pabbati, P. Ford, T. Moore, S. Herzog, and R. Hess. RFC 3182: Identity Representation for RSVP, Oct. 2001. Status: Proposed Standard.

[210] A. Yegin and Y. Ohba. Protocol for Carrying Authentication for Network Access (PANA)Requirements, June 2003. Work in Progress.

[211] T. Ylonen, T. Kivinen, M. J. Saarinen, T. Rinne, and S. Lehtinen. SSH Authentication Protocol, Oct. 2003. Work in Progress.

[212] T. Ylonen, T. Kivinen, M. J. Saarinen, T. Rinne, and S. Lehtinen. SSH Protocol Architecture, Oct. 2003. Work in Progress.

[213] T. Ylonen, T. Kivinen, M. J. Saarinen, T. Rinne, and S. Lehtinen. SSH Transport Layer Protocol, Oct. 2003. Work in Progress.

[214] R. Zhang. Extended IEEE802.1x Support Authentication of users sharing a Single Ethernet Port, Oct. 2002. Work in Progress.

[215] J. Zhou and D. Gollmann. A fair non-repudiation protocol. In *Proc. of the 15th IEEE Symposium on Security and Privacy*, pages 55–61. IEEE Computer Society Press, 1996.

[216] G. Zorn. RFC 2759: Microsoft PPP CHAP Extensions, Version 2, Jan. 2000. Status: Informational.

[217] R. Zuccherato and M. Nystrom. RFC 3163: ISO/IEC 9798-3 Authentication SASL Mechanism, Aug. 2001. Status: Experimental.