Security Considerations on Pervasive Real-time Collaboration

Fuwen Liu
Brandenburg University of Technology Cottbus
Department of Computer Science
PF 10 13 44, 03013 Cottbus, Germany
e-mail: lfw@informatik.tu-cottbus.de

Hartmut Koenig
Brandenburg University of Technology Cottbus
Department of Computer Science
PF 10 13 44, 03013 Cottbus, Germany
e-mail: koenig@informatik.tu-cottbus.de

***Abstract*— A real-time pervasive collaboration system enables people to work together on a certain task at anytime and from anyplace. Security is one of the primary concerns for these systems, when running over public networks. In particular, end-to-end security represents a crucial issue in pervasive collaboration environments due to the heterogeneity of the networks and devices used. In this paper, we explore the feasibility of security measures at the diverse protocol layers to shielding pervasive real-time collaboration systems, and propose a security framework residing at the application layer to secure pervasive collaborations.**

Keywords: security requirements; security framework; pervasive; real-time collaboraion.

I. INTRODUCTION

Real-time collaboration systems, such as instant messaging and video conferencing, are getting more and more popular due to their tremendous value in easing communication among people. The legacy real-time collaboration systems rarely support mobile collaborations. So users have to stay at a fixed location to communicate with others. This limitation confines the wide use of collaboration applications to a great extent. A newly emerging trend in real-time collaboration is the pervasive collaboration which enables people to collaborate using any device at any time and from any place. This enhances the productivity of people, since they are not bound to certain locations and can response immediately. The widespread availability of the broadband wireless networks, e.g. the universal mobile telecommunication system (UMTS) [1] and wireless local area networks (WLANs) [2], the rapid proliferation of handheld devices, such as personal digital assistants (PDAs) and smart phones, pave the way for pervasive real-time collaboration applications.

Security is one of the major concerns for the use of pervasive collaboration applications, especially in business settings, since the penalties for compromised contents can be severe. Security is usually difficult to achieve in a pervasive real-time collaboration environment due to the heterogeneity of networks and devices used. A pervasive collaboration application might run over the wired and wireless links simultaneously. Wireless links are significantly different from wired ones in that they are more vulnerable to attacks due to the public accessibility of radio transmission, and they present a higher bit error rate than wired links. Hence special concerns have to be taken when protecting an application running on top of wireless links. People may utilize various kinds of devices for their collaboration ranging from desktops and laptops with a powerful computing capability to PDAs and smart phones with limited computing power. No research efforts have been devoted to the development of a security framework for the protection of pervasive real-time collaboration applications as yet. In this paper, we strive to address this urgent issue by outlining a security framework for pervasive real-time collaboration applications. The rest of the paper is organized as follows. The architecture of pervasive real-time collaboration systems is briefly introduced in Section 2. In Section 3 we discuss the security requirements on the design of such systems. Next we explore the applicability of security measures at the various protocol layers to shielding pervasive real-time collaboration systems in Section 4. Thereafter, Section 5 gives a detailed description of the proposed security framework for pervasive collaborations. Some final remarks conclude the paper.

II. ARCHITECTURE OF COLLABORATION SYSTEMS

Pervasive real-time collaboration systems should be able to run across a variety of wired and wireless networks, such as digital subscriber line (DSL), UMTS, and WLAN. Nowadays all networks tend to be converged at the network layer, i.e. they all support the same network protocol, namely the IP protocol. This convergence makes a pervasive collaboration possible, when the users are located at the different networks. The proposed pervasive real-time collaboration system architecture is therefore built upon the IP protocol stack. The protocol stack is depicted in Figure 1.

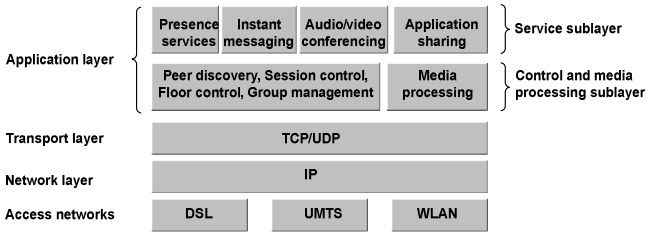

Figure 1. Protocol stack

The pervasive real-time collaboration system resides at the application layer. It is usually further divided into two

978-1-4244-5113-5/09/$25.00 ©2009 IEEE

sub-layers: the service sub-layer and the control and media processing sub-layer. The possible real-time collaboration services that the system can deliver, e.g. presence services, instant messaging, audio/video conferencing, application sharing, are grouped into the service sub-layer. The control and media processing sub-layer represents the basis of the real-time collaboration system. It consists of a media processing and a control part. Video/audio data are digitalized and compressed in the media processing part for an efficient transmission over the network. The control part is composed of several essential control functions needed by collaboration services. The *peer discovery* serves to find the contact address (usually IP address) of the intended partner no matter which network he/she is located at. The *session control* deals with the establishment and release of collaboration sessions. The *floor control* regulates the access to the shared resources to avoid race condition problems. The *group management* is responsible for the supervision of the group composition and allows every participant to learn the current state of the membership in a collaboration session.

A pervasive real-time collaboration system can be created in two ways: as a client/server system or using a peer-to-peer (P2P) approach. The proposed pervasive real-time collaboration system will be implemented using P2P technologies rather than as a client/server system, considering the drawbacks of client/server systems, such as single point of the failure, high maintenance cost, performance bottleneck.

III. SECURITY REQUIREMENTS

As other secured applications a secure P2P pervasive collaboration system has to provide the known set of **basic security services**: *user authentication*, i.e. new participants are only allowed to join a collaboration session after their claimed identities are verified; *authorization*, i.e. users are able to make authorization decisions to the incoming invitations in order to block the unwanted communication; *confidentiality*, i.e., the content of a collaboration session is only accessible to the current participants; *data integrity*, i.e. all messages exchanged in the collaboration session are not altered during transmission. Considering the decentralized structure of the pervasive collaboration system, real-time services that it provides, and the roaming possibilities of users, a security solution should meet the following requirements additionally.

End-to-end security: This is the deciding factor to determine whether a security solution can be used for pervasive collaborations or not. The security services can be provided in two fashions: end-to-end or end-to-gateway and gateway-to-gateway approach. *End-to-end security* means that messages are delivered securely from the sender host to the receiver host and that they are not accessible to any intermediate node or server along the transmission path. *End-to-gateway* and *gateway-to-gateway* security mean that messages are merely protected in the Internet, while they are transmitted in plaintext in the intranet. The gateway is the boundary between the Internet and the intranet. It is obvious that a pervasive collaboration desires an end-to-end security because security threats may occur not only in Internet but also in the intranet. It is a known fact that a significant number of threats originate from insiders as indicated in [3].

Group communication: The security solution should be able to support secure group communication. A group key management protocol rather than a two-party key exchange protocol has to be applied for a secure group communication. Two-party key exchange protocols are less appropriate for group communication because each message sent to the group has to be separately encrypted with the respective keys of the group members, i.e. *n-1* encryptions are required for an *n* participants group. On the contrary, only one encryption is needed for forwarding a message to the group, when a group key management protocol is deployed.

No administrative overhead: The membership composition of a collaboration session differs from session to session. It would raise an unbearable burden to the system administrator if he/she had to set up a corresponding security configuration for each collaboration session. Thus the security configuration has to be set up by the participants themselves rather than by a dedicated network administrator.

Internetworking roam: In a mobile environment users may migrate from one network to another one, e.g. from a UMTS network to a WLAN. User roaming should not raise additional costs for maintaining the security configuration, i.e. the users can keep the security configuration unchanged for secure collaborations even if they are roaming.

Real-time multimedia: Multimedia communication represents an essential part in pervasive collaborations. Therefore the security solution should protect not only the control signal data but also the real-time multimedia data, such as audio and video data.

Quality of service (QoS): QoS is a critical requirement for the real-time multimedia communication. The deployed security solution should not pose a negative impact on the collaboration QoS.

IV. POSSIBLE SECURITY SOLUTIONS

The security measures could be applied to the different layers of the TCP/IP protocol stack for securing pervasive collaborations. They may reside at data link layer, network layer, transport layer, or application layer. In the following we evaluate them according to the aforementioned security requirements.

A. Data Link Layer Security

The various access networks employ different security measures to protect the upper layer data. The security measures used in DSL, WLAN and UMTS are briefly introduced, respectively.

1) DSL

DSL makes full use of the legacy telephone line to provide high-speed Internet access services. The *Point-to-Point Protocol* (PPP) [4] is used as the data link layer protocol to transport the multi-protocol traffic (including IP traffic) over the DSL links. The PPP protocol itself does not

provide any security functions. It relies on the other two protocols for its security.

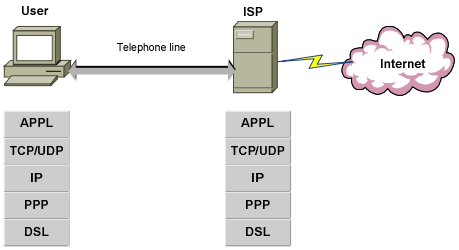

Figure 2. DSL connection

The PPP *challenge handshake authentication protocol* [5] authenticates the user at the ISP (Internet service provider) using a challenge/response scheme. The *encryption control protocol* (ECP) [6] is responsible for configuring and enabling data encryption algorithms on the point-to-point link. The security measures applied to DSL only guarantee the secure data delivery between the user and the ISP, while the traffic between the ISP and the Internet is left unencrypted. Moreover, the security solutions do not ensure data integrity.

2) WLAN

WLANs are mostly used in the infra-structure mode to provide a ubiquitous internet access for mobile users. In this mode, a WLAN is composed of nodes and the access point as shown Figure 3. The access point acts as a gateway to access the Internet.

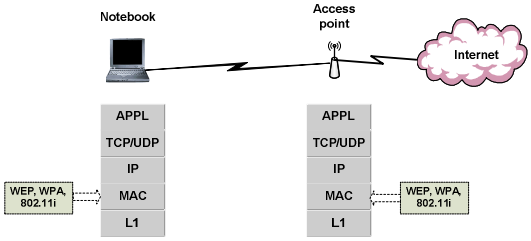

Figure 3. WLAN structure

There are several security solutions to secure message transmission between nodes and access points. The original IEEE 802.11 WLAN security mechanism is known as *Wired Equivalent Privacy* (WEP) [2]. WEP makes use of the RC4 stream cipher to encrypt the link traffic. It lacks integrity and replay protection. Moreover, WEP is considered as an unsafe scheme today, since it was completely broken in 2001 by a key recovery attack [7].

WPA (*Wi-Fi Protected Access*) overcomes the weaknesses of WEP by using the following measures. The 802.1X authentication and key management [8] is used to enable authentication of individual devices and distribute keys. WPA replaces the WEP encryption with TKIP (*Temporal Key Integrity Protocol*) which changes the way RC4 keys are used and adds message integrity and replay protection. However, WPA has still security flaws. Recently it was reported that TKIP is vulnerable to a key stream recovery attack [9].

IEEE 802.11i [10] eliminates the weakness of WPA. It adds a CTR (Counter) with the CBC-MAC protocol (CCMP) that uses the AES block cipher standard for encryption and integrity protection of the link traffic. TKIP and CCMP are both specified in IEEE 802.11i, but CCMP is mandatory for use. All security protocols (WEP, WPA, and IEEE 802.11i) operate at the MAC layer to ensure confidentiality and data integrity of the link traffic. The protection is confined to the radio links within the WLAN, i.e. links between nodes and access point.

3) UMTS

As shown in Figure 4, a UMTS network is usually partitioned into two parts: the radio access network and the core network.

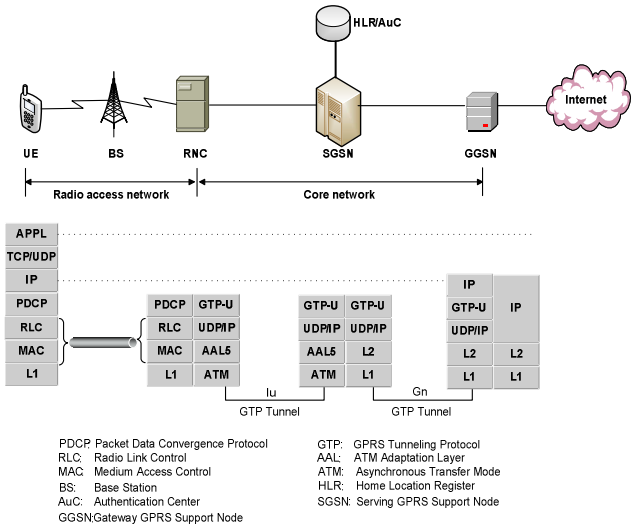

Figure 4. UMTS protocol stack

The UMTS security specifications mainly concern the security of the radio access network [11]. They enable users to securely access a UMTS network by providing the basic security services, such as user authentication, confidentiality, and data integrity. The confidentiality of the data traffic between UEs (*User Equipment*) and RNC (*Radio Network Controller*) is protected at either MAC sub-layer or RLC (*Radio Link Control*) sub-layer. The integrity of the network signaling data exchanged between UE and RNC is ensured. But no integrity protection mechanism is suggested for the user data for performance reasons. This is not a serious problem, when user data are real-time multimedia data. But this may cause application failures, when user data are the control data that are altered in transition without detection. In the core network, UMTS has defined the network domain security (NDS) which ensures that network signaling exchanges are protected within the core network. However, user data in the UMTS core network are left unencrypted. Considering the limited security coverage (only within the radio access network) as well as the limited security functionalities (only authentication and confidentiality), we can conclude that the security provided by UMTS for user data is quite inadequate

and higher layer security measures should be taken to protect user data.

4) Summary

It has shown that security solutions at data link layer aim at providing the secure network access for users. The security coverage of these solutions is confined within the access network, i.e. from the user to the gateway that connects to the Internet. This corresponds to the end-to-gateway security. We do not further examine whether the data link security can fulfil the other security requirements of pervasive collaborations or not, since it cannot provide end-to-end security.

B. Network Layer Security

IPsec was designed to protect IP traffic between two nodes at the network level assuring data integrity, authenticity, confidentiality, and anti-replay. It supports two security protocols: IP *Authentication Header* (AH) [12] and IP *Encapsulating Security Payload* (ESP) [13]. The fundamental components of the IPsec security architecture are the security policy database (SPD), the security association database (SAD), and the Internet key exchange protocol (IKE) [14], as shown in Figure 5.

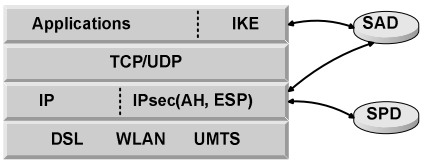

Figure 5. IPsec architecture

IPsec can protect any kind of applications including real-time pervasive collaboration, since it operates independently of the upper transport layer protocols. However, IPsec inherently possesses several disadvantages when used for pervasive collaborations:

Difficulty to afford end-to-end security: IPsec operates in the network layer which is the lowest layer to provide end-to-end security in theory, but it is rarely used for end-to-end protection of applications in practice. This mainly results from the limitations of IPsec policy mechanisms, such as the lack of expressive power in policy specifications and missing application control over policies [15]. Therefore IPsec is mostly used to afford end-to-gateway and gateway-to-gateway security as shown in Figure 6.

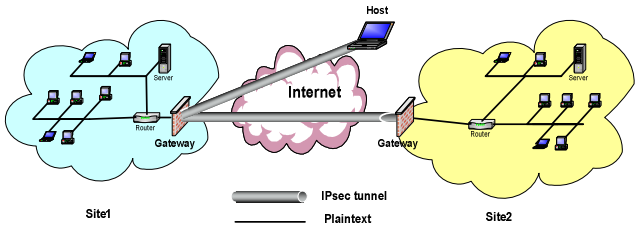

Figure 6. End-to-gateway and gateway-to-gateway security

Inefficient group support: IPsec is inefficient in supporting group communication because the key management protocol IKE is a two-party key exchange protocol rather than a group key management protocol.

Heavy administrative overhead: The associated security policies in SPD have to be manually configured in the related IP nodes prior to the application of IPsec. This specific task requires some professional knowledge about security.

Difficulty to support internetworking roam: The use of SPD and SAD is tightly bound with IP addresses. As a user moves from one network to another, he/she may be allocated a new IP address in the new network. The configuration of the data bases SPD and SAD has to be correspondingly changed due to the update of the IP address. This poses a huge management burden for mobile users.

Degrading Quality of Service (QoS): Once IPsec is applied to the IP packets, the integrity protection is always enabled because the ESP and the AH protocols both provide the integrity service. A single bit error in the packet can render the integrity check to fail, and the packet to be discarded in the IP layer. As a result, the QoS of real-time video/audio data may be degraded because an audio/video decoder is usually designed to be able to process any incoming packets for a better quality, even if they contain bit errors. This problem may be more serious in a wireless network environment due to its high bit error rate.

C. Transport Layer Security

The *transport layer security* TLS protocol [16] can be used to protect any kind of applications that utilizes TCP at the transport layer. TLS is mainly composed of two sub-protocols: the TLS handshake protocol and the TLS record protocol. TLS has several advantages over IPsec, when it is used for pervasive collaboration. It always provides the end-to-end protection for applications. No administrative overhead is raised by the use of TLS, since it is invoked by the upper layer applications on demand without the need of a network administrator for security configuration. Moreover, users do not need care about the security configuration for a roam because the operation of TLS is not bound to a fixed IP address. However, there still exist problems for TLS, when it is used for pervasive collaboration. These are:

Inefficient group communication: TLS is intended to secure a two-party communication rather than group communication, i.e. no group key management protocol is deployed.

No real-time multimedia support: Multimedia data are mostly transmitted over the UDP protocol for its higher throughput and lower latency compared to the TCP. TLS merely supports TCP based applications because its design assumes that the underlying layer offers a reliable transport service. Thus TLS cannot be applied to the protection of real-time multimedia.

D. Application Layer Security

A secure application is an application that protects itself without relying on lower layer security. This is achieved by embedding security functions into the application. PGP [17]

for secure E-mail and Groove [18] for secure workspace are typical examples. Due to its embedded implementation security solutions can provide a more tailored protection compared to those in the underlying layer. Moreover, no special administration support for configuring the system is required and no change is needed for the security configuration, when internetworking roam takes place because its operation is independent of the underlying network infrastructure. To sum up, a security scheme residing at the application layer can easier meet the specific requirements of the application than schemes located in the lower layers.

E. Comparison

The features of the security solutions introduced in previous sections are summarized in Table I.

TABLE I. COMPARISON OF SECURITY SOLUTIONS

Security requirements	Data link layer security	Network layer security	Transport layer security	Application layer security
End-to-end Security	×	?	√	√
Group communication	–	×	×	√
No administrative overhead	–	×	√	√
Internetworking Roam	–	?	√	√
Real-time multimedia	–	√	×	√
QoS	–	×	–	√

×: Not support. –: No need to discuss. ?: Difficult to support. √: Support.

It shows that the higher layer the security solution resides at, the more requirements raised by pervasive collaboration it can meet. Only application layer security offers a unique solution that entirely meets the security requirements of pervasive collaborations.

V. A SECURITY FRAMEWORK

The security architecture of a pervasive real-time collaboration system is depicted in Figure 7. It is achieved by integrating security functions into the collaboration system introduced in Section II.

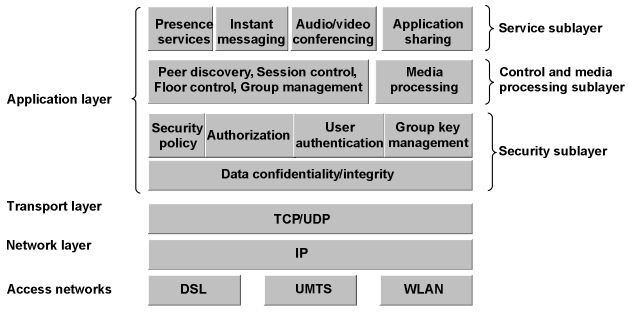

Figure 7. Security architecture

The security functions are grouped together to form a security sub-layer placed at the bottom of the application layer to provide the security services for the upper sub-layers. The security functions needed for pervasive collaboration include security policy management, authorization, user authentication, group key management, and data confidentiality/integrity, which are introduced next.

A. Security Policy Management

The security policy determines the protection level of a collaboration session and specifies the security algorithms to be applied. The main reason to introduce diverse security levels into a collaboration system is that collaboration sessions may claim varying security demands to the system. For example, a business talk needs a high security level, while a teleseminar may not require any protection at all. Moreover, devices used by participants may have different processing capabilities due to the heterogeneity of devices so that the participants have to negotiate an appropriate security agreement to admit participants with a lower power device to the session.

B. User Authentication

User authentication ensures that the claimed identities presented in the collaboration session are authentic. Password based authentication and signature based authentication are two most common used techniques: *Password based authentication* works fine in client/server environments where the server authenticates the clients by using the clients' passwords that are installed in the server in advance, but it is inappropriate for P2P systems where there exists no server centrally managing the passwords for the peers. The *signature based authentication protocol* is designed on the basis of a digital signature. The claimant demonstrates the possession of its private key by signing a message with it. The verifier can ascertain the validity of the signature by using the respective public key of the claimed identity without relying on a central server. Thus signature based authentication is well suited for P2P systems. It is worth noting that the signature based authentication protocol relies on the certificates to obtain the authentic public keys of partners. The certificates can be managed in two ways: centralized or decentralized scheme. The latter well matches the P2P communication model. But it is vulnerable to the Sybil attacks [26], i.e. attackers can use different identities to join a collaboration session so that they may always be present in the session although they are never invited. Therefore a centralized certificate management scheme such as PKI is preferred to be deployed for security reason.

C. Authorization

Like VoIP, real-time collaboration applications are susceptible to spam attacks that malicious parties can initiate unsolicited and unwanted communications to victims. There are some mechanisms developed to counter against spam attacks in VoIP systems [19]. These can be applied to pervasive real-time collaboration systems as well in principle. The black and the white list that function as ACLs (*access control list*) to authorize the access to traditional computer

systems are the most used and effective countermeasure to block the unwanted communications. The *black list* contains a list of identities regarded as spammers. The invitations originated from the users listing in the black list are automatically rejected. In contrast, the *white list* is a list of trusted users. The invitations originated from the users listing in the white list are always accepted. It is obvious that user authentication (i.e. verifying user identity) is the prerequisite to perform authorization.

D. Group Key Management

The group key management is the central component in the whole security architecture, since the security of all applied cryptographic protocols or algorithms mainly rely on the privacy of the used key. To be in line with the decentralized nature of pervasive collaboration systems, a decentralized group key protocol should be deployed in the security architecture. It has to meet a couple of rigid security requirements like key authentication, forward and backward confidentiality, collusion freedom, and others. In addition, it should be efficient enough to meet the performance requirements of pervasive real-time collaborations. Several such kinds of protocols are available, such as TGDH [20] and VTKD [21].

E. Data Confidentiality/Integrity

1) Data confidentiality

Efficient encryption algorithms should be employed to ensure data confidentiality, since mobile users may be joined in a collaboration session by using devices with poor computing capabilities. Recently some fast stream ciphers have emerged in the context of the eSTREAM project [22], such as Sosemanuk and Salsa. They can achieve a several times faster encryption speed than AES. Thus they may be good alternative encryption algorithms beyond AES for securing pervasive real-time collaboration.

2) Data integrity

In general, cryptographic data authentication schemes, such as HMAC [23], can be directly applied to messages exchanged in a collaboration session to verify their integrity in transit. But for a better QoS, content authentication schemes rather than cryptographic authentication schemes should be employed for verifying the integrity of video/audio data. The content authentication schemes intend to identify whether the content of video/audio is altered or not, not to determine whether every bit in the video/audio data is modified or not. So the receiver can still play these video/audio containing errors as long as their content is not tampered by an attacker. Feature extraction approach [24] and fragile watermarking approach [25] are typical examples of content authentication.

VI. Final Remarks

Security is of paramount importance to the widespread use of pervasive real-time collaboration applications. Through a comprehensive investigation on security schemes at the different layers, it revealed that only the application layer can fully meet the specific security requirements raised by pervasive real-time collaborations. Thereby we have proposed a security framework running in the application layer. The outstanding benefit of this scheme is that users can spontaneously set up a secure real-time collaboration without caring about the security of the underlying networks. This paper is a blueprint outlining how secure pervasive collaboration systems can be constructed. The performance of the proposed scheme needs further studies.

References

[1] 3rd Generation Partnership Project (3GPP): UMTS Phase 1 Release 99. TS 22.001.

[2] IEEE 802.11 (1999 edition) Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[3] ITU-T manual: Security in Telecommunications and Information Technology. Dec. 2003.

[4] W. Simpson: The Point-to-Point Protocol (PPP). STD 51, RFC 1661, July 1994.

[5] W. Simpson: PPP Challenge Handshake Authentica-tion Protocol (CHAP). RFC 1994, August 1996.

[6] G. Meyer: The PPP Encryption Control Protocol (ECP). RFC 1968, June 1996.

[7] S. R. Fluhrer, I. Mantin, and A. Shamir: Weaknesses in the Key Scheduling Algorithm of RC4. Selected Areas in Cryptography 2001, LNCS 2259, pp 1-24, 2001.

[8] IEEE 802.1X Standards for Local and Metropolitan Area Networks: Port Based Access Control, June 2001.

[9] M. Beck and E. Tews: Practical Attacks against WEP and WPA. http://dl.aircrack-ng.org/breakingwepandwpa. pdf

[10] NIST 800-97, Establishing Wireless Robust Security Networks: A Guide to IEEE 802.11i.

[11] V. Niemi and K. Nyberg: UMTS Security. John Wiley &Sons, Ltd, 2003.

[12] S. Kent: IP Authentication Header. RFC 4302. Dec. 2005.

[13] S. Kent: IP Encapsulating Security Payload. RFC 4303. Dec. 2005.

[14] C. Kaufman: Internet Key Exchange (IKEv2) Protocol. RFC 4306. Dec. 2005.

[15] J. Arkko and P. Nikander: Limitations of IPsec Policy Mechanisms. Security Protocols, LNCS 3364, 2005.

[16] T.Dierks and E. Rescorla: The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, August 2008.

[17] Network Associates, Inc: How PGP works. http://www.pgpi.org/doc/pgpintro/

[18] Groove:http://office.microsoft.com/

[19] J. Rosenberg and C. Jennings: The Session Initiation Protocol (SIP) and Spam. RFC 5039, January 2008.

[20] Y. Kim, A. Perrig, and G. Tsudik: Tree-based Group Key Agreement. ACM TISSEC 7 (2004) 1: 60-96.

[21] F. Liu and H. Koenig: Secure and Efficient Key Distribution for Collaborate Applications. In: Proc. IEEE CollaborateCom 2005, Dec. 2005.

[22] The eSTREAM project. http://www.ecrypt.eu.org/stream/index.html.

[23] H. Krawczyk, M. Bellare, and R. Canetti: HMAC: Keyed-Hashing for Message Authentication, RFC 2104, Feb. 1997.

[24] Chung-Ping Wu and C.-C. Jay Kuo: Speech Content Authentication Integrated with CELP Speech Coders. IEEE ICME 2001

[25] M. Steinebach and J. Dittmann: Watermarking-Based Digital Audio Data Authentication. EURASIP 2003:10, pp. 1001-1015.

[26] J. R. Douceur: The Sybil Attack. IPTPS'02, March 2002.