MIKEY-DHHMAC-SAS – The New MIKEY Transportation Mode

Alexandre B. Barreto
Div. Ciência da Computação
Instituto Tecnológico Aeronáutica
S.J.Campos. S. Paulo, Brasil

Rafael di Lego Gonçalves
Div. Ciência da Computação
Instituto Tecnológico Aeronáutica
S.J.Campos. S. Paulo, Brasil

Antônio C. Faleiros
Departamento de Matemática
Instituto Tecnológico Aeronáutica
S.J.Campos. S. Paulo, Brasil

Abstract - *Voice over IP (VoIP) technology has become more common. However, its implementation associated to the necessity of guaranteeing the security of the transmission channel compatible to the commuted telephony, is still a challenge. Despite of existing some consensus on how to provide a safety transmission on a VoIP conference, some discussions still exist on how to provide a protected form of key exchange that allows the creation of the cryptography channel. When the scenario involves the creation of this channel independent of any existing security infrastructure, the alternatives are the S/MIME, MIKEY or ZRTP protocols. Although these protocols solve the problems cited, they have limitations as it does not allow them to attend to all existing implementation scenarios. This article introduces a new form of transmission to MIKEY, the MIKEY-DHHMAC-SAS, in which main characteristics from the MIKEY protocol are kept, but it adds some existing characteristics of the ZRTP.*

Keywords: Voice over IP, SIP, ZRTP, MIKEY.

1 Introduction

Differently from the commuted telephone network, the Voice over IP (VoIP) technology is based on an IP package network, which is highly decentralized. In this way, there are many vulnerabilities and less control over the channel where the data will pass through. This fact makes its predecessor a better technical choice in scenarios where information security of multimedia data is necessary.

The VoIP technology has two alternatives to overcome those limitations. The first one is the construction of cryptography tunnels establishing connection between terminals involved in the call. In this case, the whole channel where the data will be transmitted must have some kind of trust relationship structuralized.

The second alternative searches for guaranteeing the information security solely based on protections developed in the terminals, which does not make the control over the channel where the information will pass through necessary. This is known as end-to-end protection.

The importance of end-to-end architectures resides on the possibility of its use in existing scenarios, also on those where it is not possible to guarantee a trust relationship between all the intermediary elements necessary to create a channel. It can be cited as an example, the case of an user that has a personal trust relationship with another one but the companies providing the telephone service to the users do not have a similar trust. In this case, it only is possible to establish a security channel through an end-to-end security architecture.

This article has the objective to propose an architecture that aims at guaranteeing secrecy of an end-to-end multimedia conference, using the mainly characteristics of existing protocols and eliminating its limitations and vulnerabilities.

This article is organized in the following way. Section 2 will be carried through a summary about the basic functioning of SIP, SDP and RTP/RTCP protocols. In section 3, it will be introduced a summary about the mainly existing standards of end-to-end protection of media. It will be introduced its potentialities, limitations and vulnerabilities found. In section 4, the architecture proposed will be described and explained how it solves the problems found on previous standards and its functioning structure. In section 5, final considerations will be made about the research in progress.

2 Basic Notion of IP Telephony

Second [1], the IP telephony relates to systems that make the transportation of voice, video, text and any other type of real time media through an IP package network. Despite of the existence of other protocols that implement an environment based on this architecture, in the following article it will only be studied the ones based on the SIP, SDP and RTP/RTCP protocols, since these are the most used protocols nowadays.

In general, VoIP networks follow a functioning model very similar to the one existing on traditional telephony networks, where the basic pillar of those networks is the concept of signalization. Like the existing definition in [2], signalization is the communication process that coordinates the exchange of control messages, enabling that each equipment involved in the communication may exchange information about its capacity and localization, making the creation and maintenance of a data channel possible. In a VoIP environment, the phases related to establishment and releases of the channel are implemented by the SIP protocol, while the creation and maintenance is made by the RTP/RTCP stack.

Session Initiation Protocol (SIP) is a protocol localized in the application layer of the TCP/IP model, which its objective is to establish, modify and terminate multimedia conferences between two or more users [3]. This concept defines that the SIP protocol is responsible for the negotiation of parameters between the communication pair so that the establishment of a channel can be allowed.

To implement the channel where conference data will pass through, the Real Time Protocol and Real Time Control Protocol (RTP/RTCP) stack is used [4]. This set of protocols has the objective to transport the multimedia data, combining in a balanced way the delivery control offered by TCP and efficiency of UDP protocol.

To describe the channel that has to be built by the RTP/RTCP stack, the SIP protocol uses another protocol: Session Description Protocol (SDP) [5]. The SDP provides to the SIP the necessary means to terminals involved in the communication to establish the multimedia channel. SDP is transported in the SIP message body. In this way SIP only concentrates on the specific signalization for the establishment of communication.

To accomplish this task, the user that initiates the process send an INVITE request, which in its body contain a SDP message. In this message multimedia parameters (doors, terminal final address, CODEC, etc.) that the initiator wishes to use to establish the conference are described.

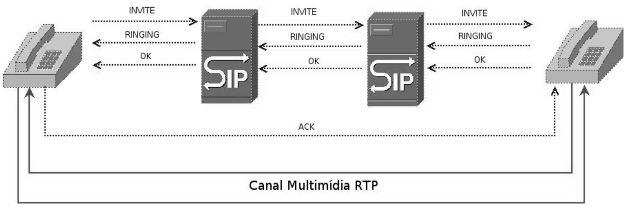

Figure 1: Establishment Process of a SIP Conference

In the case of destiny user be available, the terminal responds to the initial request with a RINGING message that it does not have any information in its SDP body.

In the moment that the remote pair answers the telephone, an OK message is sent by the pair. This message has some SDP headers in its body that describes to the initiator the way in which its pair accepts to establish the multimedia conference.

At last, the initiator terminal must send an ACK message directly to its pair, confirming the receiving of the OK answer. After its message, at the end of the transaction initiated by the INVITE request, an end-to-end and uni-directional multimedia channel is created, that it is implemented through RTP/RTCP protocol.

3 End-to-End Multimedia Security Architectures

When there is the wish to create safety multimedia architecture, the signalization is responsible for transporting the necessary parameters to the creation of the security channel. In this way, protocols like SIP and/or SDP must provide some alternatives for this task, as protocols like RTP/RTCP do the security transportation job.

There is a consensus that the more efficient way of providing a security multimedia transportation is using the stack of protocols SRTP/SRTCP [7].

Secure Real Time Protocol / Secure Real Time Control Protocol (SRTP/SRTCP) is a specific profile of the RTP/RTCP [4] that provides confidentiality and integrity to the whole RTP/RTCP package, besides it provides a protection against attacks of replay-packets kind.

To accomplish a protection against a not allowed exposure of multimedia content been transported by the RTP/RTPC protocols, only the binary data voice is transported by the RTP, while the RTCP sends statistical information of control reports. It is important to highlight that this protection does not increase the final size of the multimedia package.

In order to guarantee the correct authentication of the pairs involved in the channel, the SRTP/SRTCP stack inserts a tag in the end of the package containing a signature about the information of the SRTP/SRTCP package, so that, the overhead generated by the package be the minimum, even more when compared with tunnel based solutions.

However, the protocols stack requires that a series of information, related to the protected channel that is being created, must be agreed between the pair. From the information that need to be agreed upon, it can be cited the way that the authentication and cryptography algorithm will work; the key sizes used and the value of symmetric keys are necessary for the protection process offered by the protocol. This set of variables is necessary for the establishment and maintenance of the safety multimedia conference and it is known as cryptography context.

To increase the protocol robustness, the SRTP/SRTCP stack also uses a key derivation process which objective is to generate a individual set of keys to do the protection tasks (authentication and cipher) offered by the protocol, through only one section key negotiated between the pair and transported by a section protocol. Moreover, a salty process of the derivation keys is used to change the value package by package. For this task, it is included in the negotiation process of the cryptography context a value known as master salt.

Even thought the multimedia transportation offered by the SRTP/SRTCP is considered efficient enough, the form that the cryptography context is carried is not, which results in the existence of various standards to solve this problem.

Although the SIP specification has standard architectures for multimedia protection based in tunnels (SDES [8] and TLS [3]) and there is a considerable effort of the industry for its adoption, these technologies are not considered practicable in scenarios with many users. Because of this, the tunnels architectures will not be mentioned in this paper.

When it is necessary to guarantee security in a multimedia channel based in an end-to-end structure, only three alternatives fit this requirement, the protections based in the S/MIME, MIKE e ZRTP protocols.

3.1 Secure Multipurpose Internet Mail Extensions (S/MIME)

The SIP specification [3] only offers the use of S/MIME as a solution to provide a secure signalization to a based end-to-end architecture.

The S/MIME consists in asymmetric cipher message syntax to transference of a non-textual content (MIME). Although the standard has been developed initially for email exchange, its use can be done for any type of message that needs to be transported in a protected way through the Internet, i.e. the contents of the SIP and SDP header.

Although [3] dealing with the use of the S/MIME to protect both the content of message SIP and SDP attributes, in the present article, this protocol will be used to protect only SDP content, because the cryptographic and establishment parameters of the channel are inserted in it.

When protecting SDP header using the S/MIME special attention must be taken into the fact that some proxies need to modify specific information in order to construct a correct multimedia channel, for example, the information of the pair's address when it uses a service of translation of addresses - NAT.

To use the S/MIME, the initiator must possess the certificate of his destination pair, so it can be possible to cipher the data. This brings the necessity to carry through some additional rounds of negotiation for the acquirement of this certificate.

In addition to the fact described in the previous paragraph, this limitation causes another inconvenient, the possibility of a man-in-the-middle (MITM) attack. This attack consists of an action in which the enemy is situated between the victims, intercepts their messages, modifies its content and communicates to each one as if he is the other legal party in the conversation. The MITM may occur when the aggressor participates in the negotiation process of the cryptographic parameters.

3.2 Multimedia Internet KEYing

The second solution to implement this architecture is through the MIKEY protocol [10]. The purpose of this protocol is not only to describe the necessary parameters for the construction of the cryptographic context, but also to protect them during its transportation through an unprotected channel.

One of the great virtues of MIKEY is its capacity of negotiating the cryptographic parameters in only one round, which means that it only needs one message exchange of offer and acceptance (or refusal), thus making its insertion in the SDP [11] possible and causing a minimum modification in the preexisting signaling protocols.

MIKEY offers several types of safe transportation to the key, however all these types of transportation have some kind of limitation that makes them not very appropriate to be used on large scales.

The first type is MIKEY-PS. It uses a pre-shared key to provide authentication and secrecy of the cryptographic context. This solution is sufficiently efficient in scenarios with a small number of users, because its use in an environment with many users (n) would require a negotiation of a great number of session keys $(n^2-n)/2$, as the secrets are combined pair-by-pair.

The second type uses a public key infrastructure (PKI) to conduct this negotiation, there is two types: the MIKEY-PK and the MIKEY-RSA-R [12]. Both types use the public key of its pair to cipher the negotiated key and its private key to sign the messages negotiated by the protocol. In the first one, MIKEY-PK, a problem similar to the one found in S/MIME exists, so in order to complete conduct the cipher task, it is necessary that the user had previously acquired its pair's certificate. This problem is solved by MIKEY-RSA-R, which in the initiation message sends only its certificate instead of generating the symmetrical secret to be shared by the pairs, which makes this task to be conducted in the terminal that will answer the initial message.

Although MIKEY-RSA-R is sufficiently robust, the use of PKI can be very onerous in some scenarios, for example residential users. The use of PKI in these scenarios would make necessary that each existing user has to acquire a certificate from a certification authority.

As an alternative to the use of PKI, MIKEY offers types of transportation using the Diffie-Hellman algorithm (DH): MIKEY-DH and MIKEY-DHHMAC [13]. Even though MIKEY-DH guarantees the confidentiality of secret shared in an independent way of PKI, it still needs to use certificates to guarantee protection against man-in-the-middle (MITM) attacks. To solve this problem, MIKEY-DHHMAC protects the protocol using authenticated messages for a previously shared secret established between the pairs, this creates/generates a scalable limitation similar to the one found in MIKEY-PS.

3.3 ZRTP

The last existing alternative to the problem cited in this document is the ZRTP [14]. The ZRTP is a specification draft studying in the IETF and it uses the DH to provide the safe negotiation. However, it uses two new properties: the authentication based on SAS and the use of Key Continuity, instead of using certified and static keys shared between the pairs.

The first one, the authentication process, it is based on a signature generated by a hash function using the DH public values (DHi and DHr) generated by the pairs, this is called Shorts Authentication String (SAS). If there is a MITM attack, the DH public values received will not be those emitted by the real pairs. They must check to see if they have the same SAS, before initiating the use of the channel with sensible information. Previous to initiating the use of the channel with sensible information, the pairs should match as a proof of evidence that they have the same SAS.

The second property offered for the ZRTP is called Key Continuity. This concept consists of not using only the private value generated by the DH algorithm as the key used for the cipher process, but the combination of this value with others keys previously shared between the terminals, negotiated by DH or through other protocol, as MIKE.

This characteristic guarantees an additional resistance to protocol attacks. If anyone has suffered a MITM attack, the secret that the aggressor has is not enough to monitor the content of the channel.

Although the ZRTP is safe enough, its functioning is based on a series of posterior messages to the signaling protocols (SIP) before opening the multimedia channel. This makes the protocol not very attractive from the efficiency point of view. Such fact occurs because many users can give up the call because of the delay caused by the establishment of the secret. Moreover, when compared with MIKEY, ZRTP needs more rounds to start working.

Moreover, it is possible to insert MIKEY messages inside of the existing signaling protocols. These messages are inserted in the offer/acceptance message of the protocols using only one attribute. This facilitates the life of VoIP developers, which have to make just a few modifications in its products to provide the security offered by the protocol.

4 MIKEY-DHHMAC-SAS

During the presentation of this document some alternatives have been introduced, some are standards and others are experimental. The underlying intention of that is the construction of a safe channel between two users.

Although all the solutions presented support this intention, when referring to security in an end-to-end scenario, the architectures studied in this document have some limitations, as it has previously been presented.

However, when comparing the solutions presented, MIKEY protocol has showed to be the most efficient one because in every type of functioning, it accomplishes the negotiation of the cryptography context in a safety way in only one round.

As it has been seen, in an end-to-end scenario composed by many users (telephony in the Internet), the most attractive way of MIKEY protocol is based on Diffie-Hellman secret (DH). Among all types of MIKEY that use DH, only one (MIKEY-DHHMAC) makes the implementation of the desired scenario possible with the necessary independence. However this scenario is not very scalable because it uses a pre-shared key.

Considering this scenario, in the present chapter a new extension of MIKEY protocol will be introduced, the MIKEY-DHHMAC-SAS. This type of functioning extends MIKEY-DH standard, solving the problem of scalability found in [13]. Additionally, this new type adds the properties of Key Continuity and authentication through SAS [14], thus increasing the robustness of the original protocol [10].

4.1 General Characteristics

The MIKEY-DHHMAC-SAS is basically the MIKEY-DHHMAC with some additional protections of ZRTP protocol. Its objective is to complement the MIKEY-DHHMAC, proving the necessary scalability so that it can be used in heterogeneous environments and by a very great number of users.

Its functioning occurs in one round. It is inserted in the initiation message (DHHMAC_SAS_I) the DH public value (DHi) constructed by the initiator, while in the reply message (DHMAC_SAS_R) the value sent for the initiator (DHi) is sent, besides the public value calculated by the addressee (DHr), which makes that both messages possess a very similar format, as it can be seen in figure 02.

The great difference between MIKEY-DHHMAC-SAS and others types of MIKEY is that it offers two levels of authentication. The first level uses KEMAC header to provide the authentication with initiation and reply messages. As well as MIKEY-DHHMAC, the content of KEMAC is used only for the authentication of all MIKEY message body and its cryptography resources are not used.

Differently from MIKEY-DHHMAC, this new protocol does not use static pre-shared keys to accomplish the authentication of messages and to provide a protection against MITM attacks. Instead, MIKE-DHHMAC-SAS uses dynamic keys which might have been agreed among the keys using other protocols or old DH sessions.

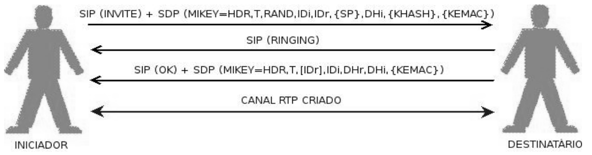
SIP (INVITE) + SDP (MIKEY=HDR,T,RAND,IDi,IDr,{SP},DHi,{KHASH},{KEMAC})
SIP (RINGING)
SIP (OK) + SDP (MIKEY=HDR,T,[IDr],IDi,DHr,DHi,{KEMAC})
CANAL RTP CRIADO
INICIADOR
DESTINATÁRIO

Figure 2: MIKEY-DHHMAC-SAS Transportation Mode

To inform which keys will be used to accomplish the authentication, MIKEY-DHHMAC-SAS uses a new MIKEY header, the KHASH (*KeyHASH*). This new header carries the last 32 bits from the signatures generated by three shared keys chosen randomly between the pairs, which will be used by the protocol to make the authentication messages. The KHASH header is represented by the expression *1*.

$$KHASH = hash\ (s_0)_{32} || hash\ (s_1)_{32} || hash\ (s_2)_{32} \quad (1)$$

Once the users get the value carried in KHASH, they must use it to generate the shared key $Kh=hash\ (s_0|| s_1||s_2)$, which will be used to authenticate the content of MIKEY messages. It must be highlighted that the hash algorithm used to generate the Kh value is the same one used to generate the *KHASH* value.

An additional comment on KHASH is that this header only exists in the initiation message. This is done to prevent that a MITM attacker changes the keys offered by the initiator, substituting them for keys that he has access.

Once all keys are agreed in the previous sessions, they are locally stored in the terminal, in an isolated form by a definite amount of time. In the case of the destination terminal does not have some of the keys offered by the initiator, the destination terminal must generate an error message and send it to the initiator.

Another characteristic of MIKEY-DHHMAC-SAS is that the use of authentication based on KEMAC header is optional. Therefore it cannot be calculated in the first safe conference negotiated between the pairs. It has been distinguished that the guarantee against MITM attacks and the correct identification of the pairs are accomplished through a second level of protection, which is based on Shorts Authentication String (SAS).

As it has been presented in [14], the SAS consists of a signature generated by a HMAC function using the DH public values agreed in the session as input. The HMAC function used is the same one specified in KEMAC header.

The SAS value has the form presented in expression 2, where DHi and DHr are respectively those DH public values calculated in the initiator and its pair.

$$SAS=hash\ (Dhi||DHr||\text{"Short String Authentication"}) \quad (2)$$

As in [14], the new MIKEY mode depends on an active participation of the user. Besides, it requires from the system some type of interface with the user, since they must check the SAS values that have been received before initiating the discussion of a secret content. In the case of suffering a MITM attack, these values will not be the same, which makes the attack to be detected and the conference interrupted, as it can been seen in figure *3*.

To provide more robustness against MITM attacks, instead of using the secret generated by DH algorithm (DHKey) to derive the used keys for the multimedia transportation protocol (TGK), the terminals will generate a new secret derived from its combination with the one used to authenticate MIKEY messages (Kh).

The TGK is defined by *TGK=hash (DHKey||Kh)*, where the hash function is the same one specified in KHASH.

The property above offers MIKEY the characteristic of Key Continuity, such as the ZRTP, which preserves the channel even if a MITM attack occurs.

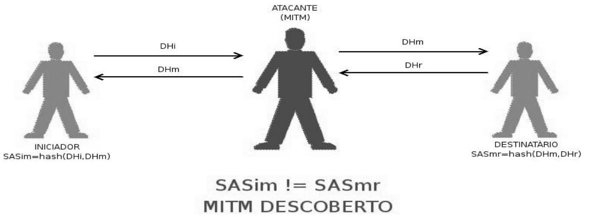

Figure 3: Protection against MITM using SAS

Since the shared keys used in the cipher process are chosen randomly, the system can store the DHKey secret automatically, as soon as the channel is initialized, therefore, it does not need any type of synchronization. This is different from the ZRTP, which offers some concern to identify and validate through the system when checking SAS.

4.2 Functioning of the Protocol

As it has been showed in figure 2, the communication process of MIKEY-DHHMAC-SAS is done in only one round, where the initiator generates a DH public value (DHi) and inserts it in an initial message (DHHMAC_SAS_I), which must be confirmed and returned with the second public part DHr in a reply message (DHHMAC_SAS_R).

The first step that the initiator must accomplish is the generation of its DH secret (x), as well as the public value (DHi) to be carried by the initiation message. The rule of generation of DH secret is the same one presented in the original specification of MIKEY protocol.

After generating the DHi, the user needs to define several parameters of the initiation message. The definition rule of these attributes is the same one in [10], with exception of the KEMAC attribute, which obeys the rules based on MIKEY-DHHMAC and KHASH. This is specific to the new type of transportation and it has been presented in the previous section.

If the initiator has all the attributes necessary to compose the DHHMAC message, then it constructs the MIKEY message, inserts it in the SDP message body[11] and it sends it to the person that he wishes to establish the communication.

When receiving the initialization message, the destination terminal shall open its pair identification header and the transportation type in which MIKEY is functioning, which in this case will be the MIKEY-DHHMAC-SAS. After that, it will locate the signatures of all the keys previously shared with that user.

If the authentication of the messages is accomplished by the MIKEY through KEMAC header and it is possible to recover the keys used through KHASH, the matching of the message authentication tag will be done. If the authentication tag does not match, an error message must be generated and sent in reply to the initiation message.

If the authentication tag matches, the terminal will proceed with the generation of the local secret (y) and DH value to be shared in the same ways of the ones done in the initiator terminal.

If the terminal has these values, it will be able to build the reply message to be sent to the initiator. Moreover, the local terminal will must generate the shared secret (TGK).

Once this task is done, the called terminal will start its multimedia channel, while it waits the initiator terminal to send a reply message.

Once the initiator received the reply message, it will have to validate it through the verification of its authentication (if such authentication exists) and after that generate the TGK shared with its pair. After this, it will have to proceed to the initialization of the local multimedia channel, which will make the establishment of a safe conference possible.

After the initialization of the channel, even before its use to transport sensible information is possible, the terminals must check the SAS values. Optionally, the application might provide an interface that allows the user to inform the validation of SAS.

The information of SAS validation can be used to define the period of time that the secret (DHKey) generated for the session will have to remain stored in a safety key repository in the system. If this information is not validated, the system will not store the key.

Once the SAS is validated, the terminals can initiate the exchange of confidential information between themself, therefore all the protections offered by the protocol are active.

4.3 Security Analysis

Since the protocol uses DH as algorithm for the negotiation of keys, a special attention must be given to the generator of random numbers. This fact, although it is not commented in details in the original MIKEY specification, is very important in the current model, since the use of random operations is very common in MIKEY-DHHMAC-SAS.

A possibility consists in generating the random numbers using algorithms based on physical phenomena as those described in [14].

It is also important to say that many operations of the MIKEY-DHHMAC-SAS are based on hash algorithms. Although the MIKEY standardizes the SHA-1 and the HMAC-SHA-1 as the hash algorithms, the use of more robust algorithms as the SHA-2 and the HMAC-SHA-2, using keys of at least 256 bits, is strongly advised. This is because there are many known attacks on family 1 of SHA.

Although attacks for the functionalities inherited from ZRTP [15] exist, they are very theoretical and only occurs in very specific scenarios of the VoIP technology. This is not the case of the scenario studied in the present document, which makes the technology safe until the present moment.

To solve the problem of the necessity of confirmation in the ZRTP, the key storage is done independent and locally in the terminal. Moreover, it is not obligatory. This is possible because the key used to compose the message is based only on some secrets randomly chosen among the secrets locally stored by the terminal. The choice of the values to be used is announced in a protected way through the HMAC message, which makes the use of robust hash algorithms necessary.

5 Conclusion

As it could be observed in this study, the MIKEY-DHHMAC-SAS is a natural evolution of MIKEY-DHHMAC, solving limitations and problems found in its previous versions and inserting other functionalities found in the new protocols in the study of the IETF.

These functionalities made the protocol more robust, however it requires that special attention and recommendations must be taken during its implementation.

This evolutionary proposal to MIKEY is being studied with the intention of optimizing the protocol, so it can be used in devices of low computational power.

6 References

[1]Ransome, James F. e Rittinghouse, John W. VoIP Security. Elvesier, 2005.

[2]Olsson, A. Entendendo Telecomunicações 2. Érica, 2000.

[3]Rosenberg, J.; Schulzrinne, H.; Camarillo, G.; Johnston, A.; Peterson, J.; Sparks, R.; Handley, M. e Schooler, E. SIP: Session Initiation Protocol. IETF RFC 3261, 2002.

[4]Schulzrinne, H.; Casner, S; Frederick, R e Jacobson, V. RTP – A Transport Protocol for Real-Time Applications. IETF RFC 3550, 2003.

[5]Handley, M., Jacobson, V e Perkins, C. Session Description Protocol (SDP). IETF RFC 4566, 2006.

[6]Stredicke, C. Securing VoIP Media Communication. Presentation carried through in June of 2006 in the Third Annual VoIP Security Workshop, Berlim.

[7]Baugher, M.; McGrew, D.; Naslund, M.; Carrara, E. e Norrman, K. The Secure Real-time Transport Protocol (SRTP). IETF RFC 3711, 2004.

[8]Andreasen F; Baugher, M. e Wing, D. Session Description Protocol (SDP) Security Description for Media Streams. IETF RFC 4568, 2006.

[9]Ramsdell, B. S/MIME Version 3 Message Specification. IETF RFC 2633, 1999.

[10]Arkko, J.; Carrara, E.; Lindholm, F.; Naslund, M. e Norrman, K. MIKEY: Multimedia Internet KEYing. IETF RFC 3830, 2004.

[11]Arkko, J.; Lindholm, F.; Naslund, M.; Norrman, K. e Carrara, E.; Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP). IETF RFC 4567, 2006.

[12]Ignjatic D.; Dondeti, L.; Audet, F. e Lin, P. MIKEY-RSA-R: An Additional Mode of Key Distribution in Multimedia Internet KEYing (MIKEY). IETF RFC 4738, 2006.

[13]Euchner, M. HMAC-Authenticated Diffie-Hellman for Multimedia Internet KEYing (MIKEY). IETF RFC 4650, 2006.

[14]Zimmermann, P., Johston, A. e Callas, J. ZRTP: Extensions to RTP for Diffie-Hellman Key Agreement for SRTP. RFC Draft, 2006.

[15]Robin, J. e Schwartz, A. Analysis of ZRTP. Final Report of Discipline CS259 - Security Protocols of the Stanford University, 2006.