A Peer-to-peer Secure VoIP Architecture

Simone Cirani, Riccardo Pecori, and Luca Veltri

Abstract Voice over IP (VoIP) and multimedia real-time communications between two ore more parties are widely used over the Internet. The Session Initiation Protocol (SIP) is the current signaling standard for such applications and allows users to establish and negotiate any end-to-end multimedia session. Unfortunately current SIP-based platforms use a centralized architecture where calls between User Agents (UAs) are routed based on static public-reachable proxy servers, suffering of well-known scalability and availability problems. Moreover security is currently poorly implemented and, when supported, it usually relies on a third-party trust relationship or on a Public Key Infrastructure (PKI). In this work we propose a completely distributed P2P VoIP architecture where calls are routed relying on a Location Service implemented through a Distributed Hash Table (DHT). End-to-end security is also provided without the use of any centralized server or PKI. Secure media sessions are established and authenticated on the basis of previously established sessions or by simple peer's voice recognition. The proposed architecture has been also implemented and publicly released.

1 Introduction

Peer-to-peer (P2P) architectures have been getting very popular in the last years thanks to the great variety of services they can provide. When they were born, they were mainly deployed as a simple, decentralized and scalable way to exchange files, but they have now become very popular also for a lot of different services, exploiting the possibility of sharing bandwidth, computing power, storage capacity and other resources between peers.

Simone Cirani, Riccardo Pecori, and Luca Veltri,
Dept. of Information Engineering - University of Parma, Viale G.P. Usberti 181/A, Parma (Italy)
e-mail: simone.cirani@tlc.unipr.it, riccardo.pecori@tlc.unipr.it, luca.veltri@unipr.it

Thanks to the large diffusion of the broad-band connections, Voice over IP (VoIP) technology has reached more and more success; with its versatility it can serve from simple audio/video communications between two entities within the same administrated IP network to conference scenarios amongst different domains. The Session Initiation Protocol (SIP) [1] is the current signaling standard for VoIP applications, implemented in almost all standard VoIP devices. SIP can be used to setup any multimedia real-time session between two or more endpoints.

Although two SIP User Agents (UAs) can communicate directly without any intervening SIP infrastructure, which is why the protocol is sometimes described as peer-to-peer, this approach is impractical for a public service. In fact, according to a pure peer-to-peer SIP scenario the caller should know how to contact the callee, that is, it has to know the callee's IP address and port number which the callee's UA is listening on. Since this information is usually unknown in advance it is necessary to rely on additional network elements (i.e., proxy servers, redirect servers, registrar servers) that, according to the SIP architecture, provide all the functionalities to register UAs and to properly route SIP calls. This is how all current SIP-based VoIP platforms have been implemented and work.

Unfortunately proxy servers represent a single-point of failure, and make the overall SIP architecture suffer of well-known scalability and availability problems.

Security is also an open issue for the current VoIP solutions since it is still poorly implemented and, when supported, it usually relies on a third-party trust relationship (e.g. users' keys are maintained and distributed by servers) or on a Public Key Infrastructure (PKI) that, in turn, is still not widely implemented and supported, and suffers of scalability problems too.

For such reasons, we studied and propose a new architecture that, widely adopting a P2P paradigm, provides secure VoIP service in a completely distributed, scalable, and reliable manner. According to such an architecture, SIP calls are routed via a Distributed Hash Table (DHT) based P2P infrastructure [2, 3] allowing the two UAs to establish any multimedia session regardless of the current points of attachment of the UAs and the available SIP nodes. Multimedia sessions are end-to-end secured on the basis of a Security Association (SA) that the two peers share without the use of any intermediary node. Such a SA is dynamically built between peers through a new key agreement protocol based on the MIKEY [4] Diffie-Hellman exchange authenticated, in a ZRTP-like fashion [5], exploiting previous established session keys or voice recognition based on the vocal reading of an authenticating short string.

The rest of the paper is organized as follows. In section 2 current VoIP architectures and signaling and security protocols are briefly summarized. In section 3 we present our P2P secure VoIP proposal, accurately describing both the distributed architecture for call routing and the key agreement protocol used to secure end-to-end VoIP communications. Section 4 presents a possible implementation and finally in section 5 we draw some conclusions and indicate further works.

2 Current VoIP architectures and protocols

The Session Initiation Protocol (SIP) [1] is the IETF standard signaling protocol defined for initiating, coordinating and tearing down any multimedia real-time communication session between two or more endpoints. Such endpoints are commonly referred to as SIP User Agents (UAs). According to SIP, in order to setup a multimedia session a caller UA sends an INVITE request to the callee UA, addressed by a SIP URI that may identify: i) the callee, or ii) the actual contact IP address and port where the callee UA can currently be found. Since the former mechanism does not require the caller to know the actual contact address of the callee UA, it is the only way currently implemented by VoIP systems. However, such a method requires a way to dynamically map a user URI to the actual contact address of one or more UAs where he can be reached. In the standard SIP architecture, this is achieved by SIP intermediate nodes (like Proxy or Redirect SIP servers) and by a proper registration mechanism through which the UAs update their contact addresses. This results into a call scheme referred to as SIP trapezoid and formed by the caller UA, an outbound proxy (optional), the destination proxy (which the callee is registered with), and the callee UA. Unfortunately such an architecture is server-centric and suffers of well-known scalability and availability problems. In order to setup a session in a real P2P fashion, a fully distributed SIP architecture is needed. Within the IETF a specific IETF WG, named P2PSIP, has been started, aiming to develop a new protocol for establishing and managing sessions completely handled by peers. The current IETF proposal is a binary protocol named RELOAD [6]. Differently from RELOAD, in our work we considered and implemented a protocol completely based on SIP. Other examples of non-IETF P2P VoIP protocols have been proposed in literature. The most relevant example is Skype [7].

As far as a secure media session has to be established, the two peers also require to agree on protocols, encryption and authentication algorithms, and keys, used to secure media contents, e.g. through the Secure Real-time Transport Protocol (SRTP) [8]. Such an agreement is often referred to as a Secure Association (SA). SAs between two peers are usually the result of a dynamic process involving a key agreement protocol (e.g. Internet Key Exchange (IKE) or TLS Handshake protocol for Transport Layer Security (TLS)) that in turn uses some pre-shared secret or public key to authenticate the SA negotiation. The core aspect of a key agreement protocol is the exchange of a master-key (or a pre-master key). For reasons of freshness and of Perfect Forward Secrecy (PFS) guarantee, Diffie-Hellman (DH) exchange is usually deployed for such an aim. Unfortunately DH is vulnerable to the well-known Man-in-the-middle (MITM) attack, through which a third party is able to trick both peers forcing them to agree to two different keys shared with itself. In order to prevent such a type of attack some sort of authentication of exchanged messages is needed. This can be achieved through the use of a pre-shared secret, by means of private and public keys and digital signature, or through the use of other authentication mechanisms such as Short Authentication String (SAS) [5]. The agreement on shared keys is a very strong assumption when applied to a P2P scenario in which peers want to communicate with each other without any pre-established relation-

ship. Moreover, Certification Authorities (CAs) and PKI, often used in conjunction with digital signatures, introduce a form of centralization that does not fit with the scalability claimed for a P2P architecture.
In the following some current key agreement protocol are briefly summarized. Multimedia Internet KEYing (MIKEY) [4] is a key exchange protocol ancillary to SRTP, or other session-level security protocols, as it provides the means for setting up session keys. It can work in three different modes in order to generate a common master key (called TGK - Traffic-encrypting Generation Key) between two parties:

- pre-shared key with key transport; it demands that an individual key is previously shared with every other peer;
- public key with key transport; it needs the knowledge of the responder's public key or of its certificate and the use of a centralized PKI;
- public key with authenticated (signed) Diffie-Hellman (DH) key exchange; it is more computationally expensive but grants perfect forward secrecy.

The main advantage of MIKEY, that is also the reason why we decided to use it, is that it is independent from any other media or signaling protocol, and can be incapsulated as part of the SDP payload during the session setup phase in SIP. Thus, it requires no extra communication overhead. In MIKEY the joint DH value is used directly as the derived key. Unfortunately, this leads to a key that does not appear as randomly generated, as it would be expected for a robust master key [9]. In our proposal we derive the master key from a hashed value of the DH value and previous master secrets, ensuring both random-likeness and dependance on previous secrets.

ZRTP [5] is an Internet-draft that describes a method to establish a session key for SRTP sessions using authenticated DH key exchange and encapsulating the relative messages in band in the media stream. The main feature of ZRTP is that the DH key exchange is authenticated through the possession of pre-shared secrets or the reading aloud of a SAS. As the authentication is not based on a centralized PKI infrastructure it is particularly suitable for a pure P2P scenario as considered in this work.

3 P2P secure VoIP architecture

In this section we present a P2P VoIP architecture that can be used to setup a secure media session between two VoIP UAs without requiring prior knowledge of the callee's IP address, and without relying on any centralized, server-based infrastructure. In order to achieve such an architecture, two main components are needed:

- a method for routing calls and for performing session setup in a completely distributed, server-free, and reliable manner;
- a method for establishing a SA and for agreeing on a session key, hopefully guaranteeing perfect forward secrecy (PFS).

The next two subsections detail how these two components are designed in our architecture.

3.1 Distributed Location Service

When a caller UA wants to initiate a session with a callee UA, it needs a way to obtain the actual network address (IP address and port) of the callee.

In the standard SIP architecture, this is usually achieved through the SIP trapezoid scheme. The goal of our architecture is basically to collapse the SIP trapezoid into a single line, connecting UAs directly. In order to create a fully distributed architecture for VoIP applications, we envisaged that SIP URI resolution can be provided by a peer-to-peer network, allowing for storing and lookup operations on SIP URIs. The most suitable peer-to-peer network type to do this is represented by Distributed Hash Tables (DHTs). DHTs are structured peer-to-peer networks which provide an information storage and retrieval service among a number of collaborating nodes. Information are stored as key/value pairs, as in regular hash tables. The structured nature of DHTs allows for an upper-bounded (logarithmic) lookup procedures, which let DHTs scale well for high number of participating nodes. Based on DHTs, we have created a framework for a Distributed Location Service (DLS) [2]. The DLS is a peer-to-peer service built upon a DHT which allows to store and retrieve information about the location of resources in order to allow direct connections among the endpoints of a communication. From an application perspective, the DLS offers two main methods: put(key,value) used to store a mapping into the DHT, and get(key) to retrieve the value associated with the given key.

According to our P2P VoIP architecture, the DLS stores mappings between a URI identifying the resource (the callee UA) and a set of contact for the resource (where and how the UA is currently reachable). Such information includes the routable URL of the UA (containing IP address and port number), an optional human-readable display name, an expiration time, and an access priority value. An example of session setup between two SIP UAs through DLS is depicted in figure 1.

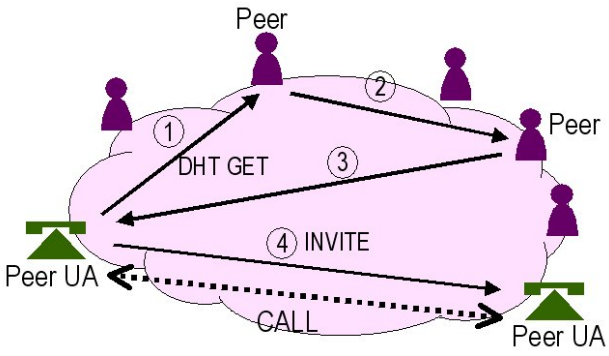

Fig. 1 P2P session setup through DHT-based DLS.

3.2 P2P VoIP security

In this section a new key agreement protocol for multimedia P2P communications is described. The objective of the proposed protocol is to securely establish a master key between two multimedia SIP UAs that may or may not have already communicated with each other. Our proposal has been designed in such a way that it does not rely on any centralized PKI, as the new master key, created through a DH exchange, is authenticated in one of the following methods:

1. by means of the previously established secrets between the two peers;
2. by means of a pre-shared key (PSK) or passphrase;
3. by performing a ZRTP-like SAS based authentication.

Note that the latter method which directly involves the two users by requiring them to read and compare the SAS (and verify the correctness of the voice of the remote peer), is used only in case the previous methods are not available or have failed.

The proposed mechanism is similar to the one used by the ZRTP protocol [5]; however the two mechanisms differ in some aspects and particularly:

- in the information exchanged during the key setup,
- in the way such information is effectively encapsulated and exchanged,
- in the possibility of authenticating the DH exchange through the use of a pre-shared secret (e.g. a passphrase).

Particularly, ZRTP establishes a new master key directly at media level, by using the RTP protocol as transport support for the key negotiation. Instead, the proposed solution uses MIKEY as negotiation protocol, opportunely encapsulated within SIP messages used for the session setup. In order to support a fully authenticated DH exchange, the MIKEY protocol has been extended to consider a MIKEY 3-way handshake (MIKEY originally supported DH in a 2-way request/response transaction). The new offer/answer/confirm handshake between the initiator (the caller) and the responder (the callee) is depicted in figure 2.

The Initiator is the entity sending a MIKEY OFFER to a Responder. The Offer encompasses: i) a MIKEY header (HDR), ii) identities of both the initiator (ID_i) and the responder (ID_r), iii) a random value $(RAND_i)$, iv) the list of the offered encryption and hash algorithms (SA_i), v) the DH part of the initiator (DH_i), vi) secrets used for the further exchange authentication $(AUTH_{offer})$. The $AUTH_{offer}$ is particularized depending on the selected authentication method. In case master keys from previously established sessions are used for authentication (1), the $AUTH_{offer}$ is formed by two values RS_1 and RS_2 (retained secrets) respectively obtained directly by the last two previously established master keys MK_1 and MK_2, as follows:

$$RS_j = HMAC(MK_j, \text{“Retained Secret”}) \ , \ j \in \{1,2\}.$$

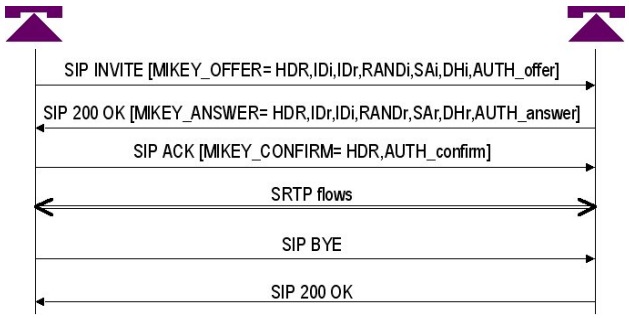

Fig. 2 Proposed three-way key agreement and session setup.

Such RS_j are used when computing the actual authentication field in the ANSWER and CONFIRM messages. The reason for sending two retained secrets is to face the case when in a previous setup, one of the two parties succeeded into computing the correct master key and the other one did not (e.g. caused by a fatal interruption during the session set up).

In case a pre-shared key (PSK) or passphrase is used (2), or in case of SAS based authentication (3), the $AUTH_{offer}$ field contains no data, and it is left empty. Note that authentication method 2 is used only if no previous secrets have been established and saved, while method 3 is used only if no previous secrets nor pre-shared secret is available between the two parties, or if the previous methods have failed.

Once the responder receives the MIKEY OFFER message, it controls the entire message, chooses $RAND_r$, SA_r, and its part DH_r of the DH exchange and calculates the hash (DH_{res}) of the new generated DH secret. He then generates the new master secret MK_0 as follows:

$$MK_0 = hash(DHres||ID_r||RAND_i||RAND_r||MK_j)$$

where MK_j is the previous master key corresponding to the more recent RS_j that matches one of the local stored retained secrets; if both given retained secrets don't match the locally stored ones, or no RSs have been given at all, MK_j is left empty. The responder composes a MIKEY ANSWER message including: MIKEY HDR, the identities ID_i and ID_r, the random value $RAND_r$, the selected SA_r, the responder's DH part DH_r. Then he calculates an authentication MAC of the entire MIKEY ANSWER message as follows:

$$HMAC_r = HMAC(MK_0, \text{MIKEY ANSWER})$$

and uses it to build an $AUTH_{answer}$ field that is appended to the ANSWER. The $AUTH_{answer}$ depends on the type of authentication that is performed:

- if $AUTH_{offer}$ contained retained secrets, the $AUTH_{answer}$ includes the matching RS_j and a $HMAC_r$;
- if $AUTH_{offer}$ was empty, if a pre-shared key PSK is available, the responder composes an $AUTH_{answer}$ formed by the $HMAC_r$ encrypted with the PSK;

- otherwise $AUTH_{answer}$ is simply formed by the $HMAC_r$, and SAS authentication is performed successively.

Once the initiator receives the MIKEY ANSWER message, it checks the correctness of the $HMAC_r$ and, if it succeeds, it sends a MIKEY CONFIRM message including a $AUTH_{confirm}$ built with the same rules used by the responder. The $AUTH_{confirm}$, in turn, includes a $HMAC_i$ that authenticate the original MIKEY OFFER with the new master key, calculated as follows:

$$HMAC_i = HMAC(MK_0, \text{MIKEY OFFER})$$

If SAS authentication is required, it takes place after the multimedia session has been setup. A short authentication string is generated as follows:

$$SAS = \text{hexadecimal values of first n bytes of } HMAC(MK_0, \text{“SAS”})$$

Both users are then invited to read the SAS aloud. If the SAS showed at both UA sides vocally matches, the new master key is considered secure and is saved in order to be used for authenticating successive master keys.

The described MIKEY offer/answer/confirm exchange is encapsulated within the standard SIP session setup procedure and can be used for establishing any P2P multimedia communication.

As an example, in figure 3 an INVITE message with MIKEY offer is shown.

```
INVITE sip:bob@192.168.1.8:5080 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.66:5070;port;branch=z9hG4bK1ef3d084
To: "Bob" <sip:bob@192.168.1.8:5080>
From: "Alice" <sip:alice@wonderland.net>;tag=857919546037
Call-ID: 747548207772@192.168.1.66
CSeq: 1 INVITE
Contact: <sip:alice@192.168.1.66:5070>; expires=3600
Security-Association: mikey
offer="AQQFgJraE7sDAAEAAAYT6bW4NALuaygB7msoAQPptbg0AAAGEwsAzdJQc
Gi0OVgGFFx3UOLse/3zehlCOFHvTuTZDQJhBgAAF0FsaWNIQHN0dWRlQHN0dWRlb
nRpLnVuaXByLmI0CgAAFUJvYkBzdHVkZW50aS51bmlwci5pdAMBAAAbAQMBAQECA
QIDBQMDAwMDBAIEBAUGBQUFBQUFDQAAzzhvYedNFRzEykg2QS56fxw6edNp5if+V
RthEancWoRwFk6++z7EAoMqrcU7Rn0t8ZiQX/aSBwQ+32rQhBhrm5W5modA0sQOh
iWikvrKxkMbv2m7rt61YTWqJpPzqhAnhHeoLCsR7ZDo1EoMdREiJolRf2lcMX5Y+
9o3vdSLXFkSLN+IqeKTpE6yYjhvoTuN/I+3QcN6j0OWXKaW3eYR5PhKb55V+Tkrc
CXYehDpETyG6O55xO7G83p0WJBPeXBTAADN0vNUDuDlq9XkNAlONjER1eQ0CU42M
RHV5DQJTjYxEdXkNAIONjER"
Content-Length: 143
Content-Type: application/sdp

v=0
o=alice 0 0 IN IP4 192.168.1.66
s=-
c=IN IP4 192.168.1.66
t=0 0
m=audio 3000 rtp/avp 0 8
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
```

Fig. 3 SIP INVITE message including MIKEY offer.

4 Implementation

The proposed P2P Secure VoIP architecture has been completely implemented in Java language, according to the specifications provided in the previous section, and integrated into an open source SIP UA. For this purpose we have implemented a DHT-based DLS, completely transparent to the particular DHT algorithm and RPC protocol used for DLS maintenance. Our current implementation uses Kademlia [10] as DHT algorithm and dSIP [11] as DLS signaling protocol [2]. Pure P2P SIP calls are performed by exploiting the DLS as a SIP LS. After the P2P UA has enrolled into the DHT, it stores within the DHT the binding of the current UA's address to the user SIP URI. When a UA wants to perform a SIP call, the peer performs a lookup to resolve the target user's address, retrieves its location, and sends the INVITE request to the UA. Legacy SIP UAs are also supported by using a special peer named SIP Adapter and acting as SIP Proxy server. Registration requests received by the SIP Adapter peer are translated to DHT PUT requests, having the effect of storing the UA's contact into the DHT. Outgoing INVITE requests are sent to the SIP Adapter peer that will perform the lookup on behalf of the UA and forward the request.

The implementation has been based on the open source MjSip stack [12] that is a complete Java-based implementation of the layered SIP stack architecture as defined by RFC 3261 [1], supporting Both JavaSE and JavaME (J2ME/CLDC1.1/MIDP2.0). The key agreement protocol described in the previous section has been also implemented ([13] reports a first implementation of such a protocol) and integrated. According to that, when a UA contacts a remote UA for the first time no pre-stored keys are available and the media flows are established through SRTP by using a new unauthenticated DH generated master key. In such a case, the two parties perform SAS authentication, that in turn leads the two users to read a displayed SAS string. If the authentication succeeds, the new key is stored and re-used for authenticating further key agreement procedures, without requiring SAS re-authentication.

5 Conclusions

In this paper we have presented a distributed architecture for P2P secure session initiation. In order to correctly route calls between any peers, a Distributed Location Service has be considered, based on DHT.

Security of end-to-end sessions is guaranteed by media encryption and authentication via SRTP protocol. The SRTP master key is negotiated through a proper new key agreement protocol that does not require any third-party relationship. The key agreement is performed via the DH algorithm and authenticated through a previously used session key (between the two parties), if available, or by means of vocal reading and recognition of a short string (SAS).

The proposed architecture has been also implemented in Java language, based on a SIP open source implementation. The current implementation includes a DLS

based on Kademlia as DHT algorithm and dSIP as communication protocol. We also realized the Peer Adapter (that is a peer that can act as a standard SIP proxy for legacy UAs) in order to route calls between DHT unaware UAs.

References

1. Rosenberg J et al (2002) RFC 3261: SIP: Session Initiation Protocol. IETF Standard Track. http://www.ietf.org/rfc/rfc3261.txt
2. Cirani S, Veltri L (2008) Implementation of a framework for a DHT-based Distributed Location Service. In: Proceedings of the 16th International Conference on Software, Telecommunications and Computer Networks, Split-Dubrovnik, Croatia
3. Cirani S, Veltri L (2007) A Kademlia-based DHT for Resource Lookup in P2PSIP. IETF Internet-Draft ciranip2psip-dsip-dhtkademlia-00. http://tools.ietf.org/html/draft-cirani-p2psip-dsip-dhtkademlia-00
4. Arkko J et al (2004) RFC 3830: MIKEY: Multimedia Internet KEYing. IETF Standard Track. http://tools.ietf.org/html/rfc3830
5. Zimmermann P, Johnston A, Callas J (2010) ZRTP: Media Path Key Agreement for Secure RTP. IETF Internet-Draft draft-zimmermann-avt-zrtp-21. http://tools.ietf.org/html/draft-zimmermann-avt-zrtp-21
6. Jennings C et al (2010) REsource LOcation And Discovery (RELOAD) Base Protocol. IETF Internet-Draft draft-ietf-p2psip-base-09. http://tools.ietf.org/html/draft-ietf-p2psip-base-09
7. Baset S A, Schulzrinne H G (2006) An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. In: Proceedings of the 25th IEEE International Conference on Computer Communications, Barcelona, Spain
8. Baugher M et al (2004) RFC 3711: The Secure Real-time Transport Protocol (SRTP). IETF Standard Track. http://www.ietf.org/rfc/rfc3711.txt
9. Gupta P, Shmatikov V (2007) Security Analysis of Voice-over-IP Protocols. In: Proceedings of the 20th IEEE Computer Security Foundations Symposium, Venice, Italy
10. Maymounkov P, Mazires D (2002) Kademlia: A Peer-to-Peer Information System Based on the XOR metric. In: 1st International Workshop on Peer-to-peer Systems, Cambridge, MA, USA
11. Bryan D (2007) dSIP: A P2P Approach to SIP Registration and Resource Location. IETF Internet-Draft draft-bryan-p2psip-dsip-00. http://www.p2psip.org/drafts/draft-bryan-p2psip-dsip-00.html
12. Veltri L (2010) MjSIP Project. http://www.mjsip.org/
13. Pecori R, Veltri L (2009) A Key Agreement Protocol for P2P VoIP Applications. In: Proceedings of the 17th International Conference on Software, Telecommunications and Computer Networks, Hvar-Korcula-Split, Croatia