

Using Cryptographically Generated SIP-URIs to Protect the Integrity of Content in P2P-SIP

Jan Seedorf

Security in Distributed Systems (SVS)
University of Hamburg, Dept. of Informatics
Hamburg, Germany
seedorf@informatik.uni-hamburg.de

ABSTRACT

Recently, it has been proposed to use a peer-to-peer network instead of servers to facilitate SIP user registration and location. One of the main security problems of using SIP in a peer-to-peer setting (P2P-SIP) is the authentication of content¹ (the binding of SIP-URI and user location) in the absence of a central authority. Using self-certifying data is one known approach to authenticate content in distributed databases. In this paper, we discuss the possibility to use cryptographically generated SIP-URIs as self-certifying data for user registration and location lookup in a P2P network. With such a solution, any user can generate a SIP-URI and sign binding updates. The authenticity of these binding updates can be verified by any entity in the network without relying on any kind of security infrastructure. We show how our solution works in principle and how it renders man-in-the-middle attacks on content stored in the overlay infeasible. Further, we discuss some practical problems that arise when using self-certifying SIP-URIs.

Categories and Subject Descriptors

C.2.4 [Computer-Communications Networks]: Distributed Systems – *Distributed applications*

General Terms

Design, Security

Keywords

Peer-to-peer, session initiation protocol, VoIP security, distributed hash tables, self-certifying data, P2P-SIP

© ACM, (2006). This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. The definitive version was/will be published in *ISW06, June, 2006, Berlin, Germany*.
Copyright 2006 ACM 1-59593-387-5

¹ In this paper, *content* describes the content to be stored distributedly in the P2P network; this is not to be confused with the content of the voice-over-IP connection (the actual voice traffic) which is usually transferred via RTP (real-time transport protocol)

1. INTRODUCTION

Voice-over-IP (VoIP) has matured over the past years to a widely used application. During this process, the session initiation protocol (SIP) [1] has evolved as a standard for signaling in multimedia connections. SIP is a text-based client-server protocol. However, recently it has been proposed to use a peer-to-peer network instead of servers to support mobility (the registration and lookup of a current user location) in SIP communications (P2P-SIP) [2], [3], [4].

The advocates of P2P-SIP state its fast setup, easy deployment, and robustness against failure as benefits compared to using servers [5]. To analyse the benefit or a potential business model for P2P-SIP is outside the scope of this paper. Instead, we focus on security and specifically authentication in P2P-SIP networks.

Due to the lack of a central authority, authentication of nodes participating in the P2P network and authentication of the content stored in the P2P network are non trivial tasks. However, without authentication of the content stored in the network the service offered by such a network is of little use: Nodes cannot verify that messages they receive from the overlay have not been altered on some (overlay-) hop by an adversary node.

In this paper, we present an approach to authenticate data stored in a P2P-SIP network. Our solution makes man-in-the-middle attacks on content stored in the network infeasible. Further, the authenticity of location-bindings can be verified by any entity in the network without relying on any kind of authentication infrastructure. To accomplish this, we use self-certifying SIP-URIs. We show how message-integrity can be achieved for P2P-SIP with our solution. Finally, we discuss benefits and deficiencies of this approach.

The rest of this paper is structured as follows: Section II gives a short introduction to Structured Overlay Networks and Chord. Further, it is described how an overlay network can be used for SIP registration and location. Section III summarizes previous work on the security of Distributed Hash Tables (which focuses to large extent on availability) and explains the different focus of this paper: message integrity. The rationale behind the solution presented in this paper is elucidated in section IV before the proposed scheme is presented in detail in section V. Section VI discusses the deficiencies and benefits of the proposed scheme as well as related work. The paper concludes in section VII with a summary and an outlook on future work.

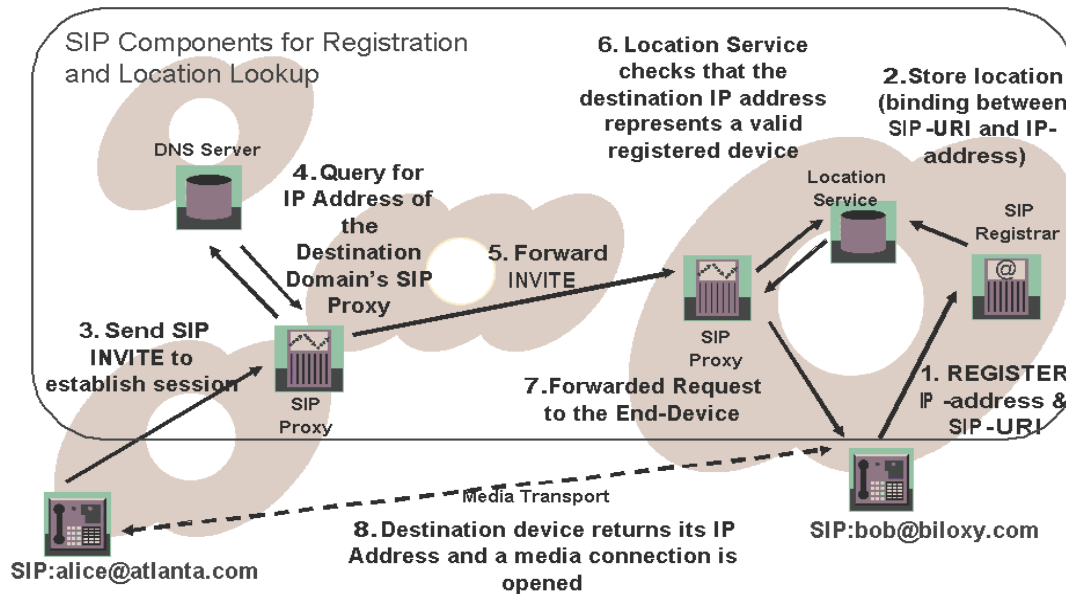


Figure I. Setup of a Call with Client-Server SIP

2. USING PEER-TO-PEER NETWORKS TO SUPPORT MOBILITY FOR SIP

2.1 SIP User Registration and Location Lookup with Client-Server SIP

SIP is originally specified as a protocol that uses servers (namely: proxy, redirect, registrar, and location server) for user registration and location [1]. Unified resource identifiers (URIs), such as `SIP:bob@example.de`, are used for addressing in SIP. In this way, SIP enables mobility: users can register their location with a registrar by sending a message that binds their SIP-URI to their current location (e.g. IP-address and port). This binding gets stored by the registrar at a location server which can be used for location lookup by other SIP entities. For locating the location server of a domain SIP relies on DNS. The static SIP-URI of a user is called *address-of-record* (for example `bob@example.de`). The current location of the user is also stored as a URI, the so-called *contact address* (for example `bob@192.168.2.1:5060`). We will generalise the concept and simply denote the SIP address-of-record as the SIP-URI and the contact address as the user location. Throughout this paper we will refer to the use of servers to support mobility as *Client-Server SIP* (the common usage of SIP today, as described above).

Fig. I shows a simple call setup with (Client-Server) SIP [6]. In this example, Alice's and Bob's user agent are in different domains and have different proxies. First, the callee (Bob's user agent) needs to register with its local registrar (1) to be able to receive calls. The registrar stores the location information at a location server (2). When Alice's user agent wants to call Bob's user agent, it sends an INVITE-request to its local SIP-proxy (3) which passes on the request - possibly after a DNS lookup (4) - to the proxy of Bob's domain (5). The proxy in Bob's domain needs to look up the IP-address of Bob's user agent at the location

server (6) before it can send the request to the user agent (7). After Bob's user agent receives the Invite-message the actual media content between Bob and Alice can be sent directly from user agent to user agent. Note that all the servers are necessary to facilitate mobility of the participating users.

2.2 SIP User Registration and Location Lookup Using Structured Overlay Networks

Singh/Schulzrinne [2] and Bryan et al. [3], [4] have published proposals to use a peer-to-peer network (P2P) to support mobility in SIP communications. We will call this approach P2P-SIP, summarizing the basic scheme embodied in their publications. P2P networks are also called overlay networks; we will use both terms interchangeably throughout this paper. With P2P-SIP, user registration and location lookup are provided by a peer-to-peer network instead of the SIP registrar/location servers. A *Structured Overlay Network* is used for reliable storage and lookup of user locations. Structured Overlay Networks are peer-to-peer networks that have been developed with focus on reliable, distributed content storage. Some popular examples for Structured Overlay Networks are CAN [7], Chord [8], Pastry [9], or Tapestry [10]. In these networks a search request will succeed with high probability if the content requested is stored in the network. Further, the network can give formal guarantees on the number of routing hops a search request needs to get forwarded until it reaches the node in the network that stores the desired content. For example, any key lookup in a Chord network of size m will need at most $\log(m)$ routing hops in order to reach the node responsible for the content belonging to the key [8]. The proposals for P2P-SIP use Chord as the overlay protocol for their prototype specification of P2P-SIP.

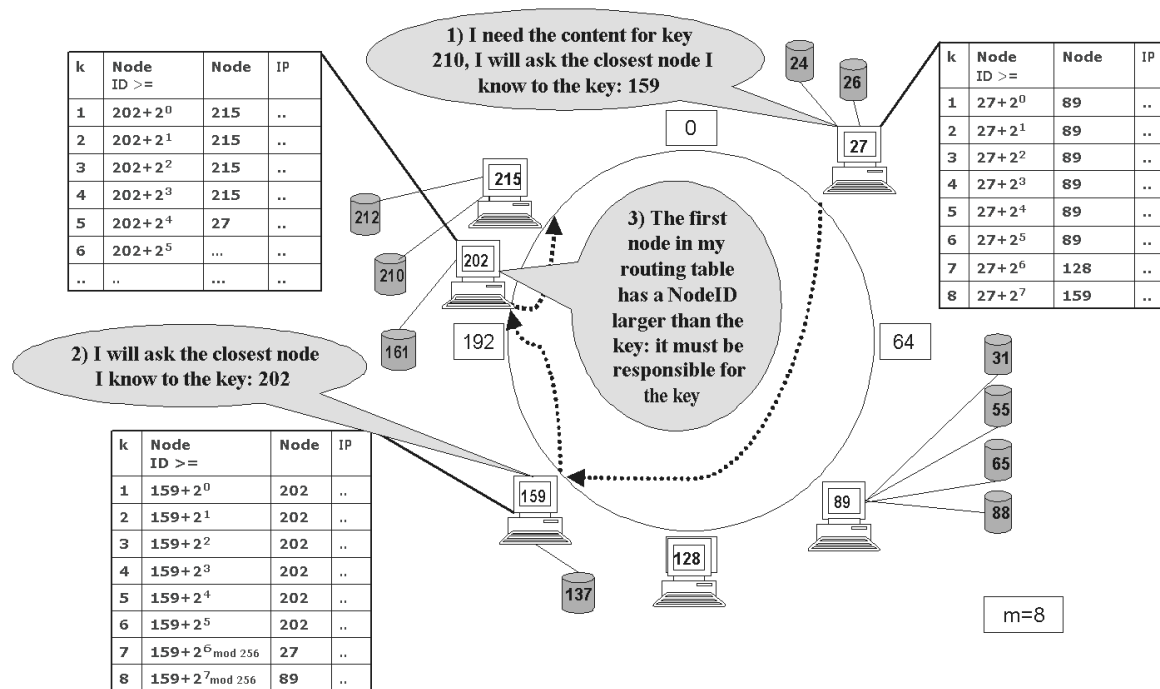


Figure II. Routing in Chord

2.2.1 Chord

Chord uses a *Distributed Hash Table* (DHT) for content storage and lookup. A Distributed Hash Table stores content in a distributed database. So-called *keys* are used as content-IDs: Each node participating in the DHT can compute the key for some content in order to determine the node responsible for the content. The DHT algorithm guarantees efficient routing of lookup requests to the node in the DHT that is responsible for storing the desired content.

In a Chord network, every node gets assigned an m -bit *node-ID* by hashing the node's IP-address with a predefined hash function h . The same hash function h is used to map content data (that is stored in the DHT) onto a key: $\text{key}=h(\text{content})$. Chord arranges nodes in a virtual ring of size 2^m . Every node n has a predecessor (the node with the highest node-ID smaller than n 's node-ID) and a successor (the node with the smallest node-ID larger than n 's node-ID).

Each node n is responsible for storing the content for keys that have a higher value than its predecessor's node-ID, up to n 's node-ID. n also stores its predecessor's IP-address. Further, a routing table is maintained by n in which it stores the IP-addresses of m nodes with a node-ID higher than n . Each routing table entry points to a node exponentially further away from n : The k^{th} entry in the routing table of node n is the first node in the ring with a node-ID higher than $\{n\text{'s node-ID} + 2^{k-1}\} \bmod m$. Since modular arithmetic is used, routing past 0 is possible in the ring.

Fig. II shows a Chord ring with $m=8$. Six nodes are in the ring. Each of these nodes stores the content for keys it is responsible for. For example, the node with ID 215 stores the content for key 210 and key 212. In its routing table, each node keeps m (8) entries. If node 27 wants to request the content with key 210, it sends a message to the node with ID closest to but below the key (node 159). This node again forwards the message in the ring. Finally, node 202 realizes that the first node in its routing table (its direct successor in the ring, node 215) has a node-ID higher than the requested key: Hence, it must be responsible for storing the content for key 210.

When a new node joins the DHT, it first computes its node-ID by hashing its IP-address. It then uses a join message to determine the nodes preceding and succeeding it in the Chord ring. Finally, the new node exchanges messages with its successor and predecessor to update routing tables. More details on Chord operations can be found in [8].

Routing in Chord can be done in two different ways: *iterative* or *recursive* routing. With *iterative routing*, a node c_1 contacted by a requesting node r , returns the node closest to the key, c_2 , from its routing table. r can then contact c_2 to get iteratively closer to the key. Finally, node c_1 is responsible for the key and delivers the content requested by r . With *recursive routing* (as in fig. II), a node d_1 contacted by a node r forwards the message to the closest node to the key from its routing table (d_2). The message is forwarded further (e.g. proxied as in Client-Server SIP) through the network until it finally reaches the node responsible for the key (d_i). For a more detailed introduction to Structured Overlay

the-middle attack. Nodes can also arbitrarily insert false messages in the overlay. To prevent some of the attacks on routing in Distributed Hash Tables, Castro et al. suggest adding a central authority to the overlay network which certifies node-IDs. Based on these certified node-IDs, availability of the service provided by an overlay network can be achieved [14].

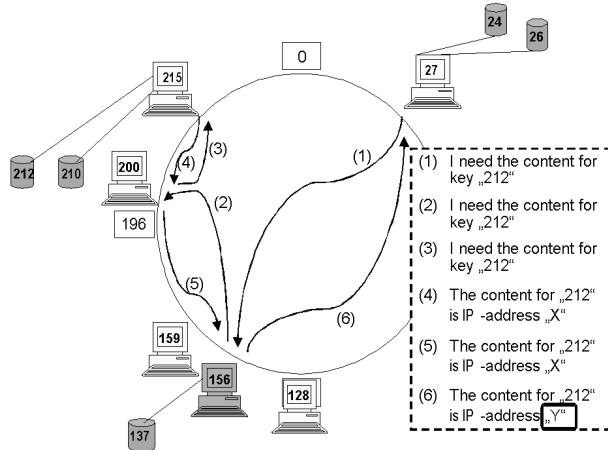


Figure IV. Man-in-the-Middle Attack on Content Stored in a P2P-SIP Network

3.2 Attacks on Message-Integrity in P2P-SIP

Previous work on DHT security considers mainly techniques to provide availability of content stored in the network. However, even if the availability of content can be achieved (by added security measures) other attacks on Structured Overlay Networks are possible. For instance, a message that gets delivered successfully to its recipient in the overlay could have been altered by any intermediate entity.

As an example, fig. IV shows a man-in-the-middle attack on content that is stored in a P2P-SIP network (the binding of SIP-URI to a location). In the example recursive routing is used. When iterative routing is used similar attacks are possible. Without authentication of messages, the requesting node (node 27) cannot verify the authenticity of messages it receives from the overlay. Any intermediate node on the overlay path can change the content of the message, thereby directing the phone call that follows the location lookup to a false location. In the example, node 156 acts as an adversary and exchanges the IP-address before handing over the message to the requesting node 27.

The attack model for a DHT depends on the application the overlay is used for. In the case of P2P-SIP the integrity of messages has to be protected (in addition to availability): Otherwise, attacks on the content stored in the network are possible, as shown in the previous example. The content stored in the network is the binding of SIP-URI (the identity of users in the system) and a location. Thus, it is of high importance to protect the integrity of the content in order to prevent impersonation of SIP-URIs by an attacker.

4. WHY USE SELF-CERTIFYING SIP-URIS?

In this paper, we focus on attacks on the content stored in Distributed Hash Tables. More specifically, we look at the

location bindings stored in P2P-SIP networks and how to protect the integrity of these bindings. Our solution uses self-certifying data as content. In this section we motivate why we take this approach and why it is a good choice to protect the integrity of location bindings for P2P-SIP.

4.1 Options for authenticating content in P2P-SIP

To protect the integrity of content stored in a peer-to-peer network, one obvious solution is to add a central authority which certifies nodes (more precisely: node-IDs) in the network. Nodes could then sign content before storing it in the network. Verification of these signatures would be possible for any node requesting content stored in the network through the added public key infrastructure.

A different approach – which has been suggested to secure Structured Overlay Networks – is to use a (distributed) reputation management system. In a reputation management system trust values are assigned to nodes in the network based on prior behaviour. A node can decide whether to trust information it receives from the network based on these values.

In this paper, we propose a different approach to authenticate content in Structured Overlay Networks: self-certifying data. The idea of self-certifying data is not new and has been introduced in several applications (e.g. [15], [18]). We suggest this approach for P2P-SIP because of the deficiencies we see in the other options:

- *Trusted Authority for Certification:* A central authority which certifies identities limits scalability of the network. Also, a central authority would lower the ease of deployment of the network and add a single point of failure to the network. Thus, it diminishes much of the advocated benefits of P2P-SIP. Furthermore, known problems with certification infrastructures such as revocation of certificates, validation of identities to certify, and trust in the central authority would be inherited.
- *Distributed Reputation Management System:* At the time of this writing, there does not exist a fully distributed reputation management system that can be integrated into P2P-SIP without further ado. Most reputation management systems that have been introduced for P2P networks focus on traditional file-sharing applications and the problem of free-riding (nodes using the services of a P2P network without delivering services to the network). In addition, most existing reputation management systems are not fully distributed: They rely on a central authority to store trust values.

4.2 Self-Certifying Identities

A self-certifying identity is an identity where ownership of the identity can be verified without relying on a trusted third party (e.g. a certificate authority in a public key infrastructure). This can be achieved as follows: The identity to be verified is represented as the hash of a public key. Only the owner of the identity possesses the corresponding private key and can prove its ownership by signing the identity. In the case of P2P-SIP, the identity stored as content in the network is the SIP-URI. Hence,

for a self-certifying SIP-URI the URI must be generated as the hash of a public key. Our scheme is described in detail in the following section.

5. A SCHEME FOR SELF-CERTIFYING SIP-URIS

5.1 Using Self-Certifying SIP-URIs in P2P-SIP Networks

In order to protect the integrity of location binding updates, users can use self-certifying SIP-URIs. First, the user needs to create a public-private key pair². Then she/he hashes the public-key and converts the hash into a valid SIP-URI. She/he can then sign any binding update to be stored in the network with the corresponding private key.

To check that a user location (IP-address and port) stored in the overlay indeed belongs to the specified SIP-URI, any node requesting a key lookup can verify the authenticity of the location-binding it receives from the network by doing two things: a) hashing the public key (which is sent along) to check that the public key indeed belongs to the SIP-URI and b) verifying the digital signature with the public key.

In detail, the scheme works as follows:

5.1.1 Cryptographically Generating a SIP-URI

1. User generates an arbitrary RSA public/private key-pair, computing $k_{pub,u1}$ and $k_{priv,u1}$ for his SIP-URI $u1$ to be generated:

$$RSA.Generate(k_{pub,u1}; k_{priv,u1}) \quad (1)$$

2. User hashes his public key $k_{pub,u1}$ with a predefined hash function h which is collision resistant and second pre-image resistant to obtain $h1$:

$$h1 = h(k_{pub,u1}) \quad (2)$$

3. User converts $h1$ to a string using a predefined function f to obtain $f1$:

$$f1 = f(h(k_{pub,u1})) \quad (3)$$

4. User prepends the generated string $f1$ followed by $@$ to his domain to generate his SIP-URI:

$$u1 = f1@domain \quad (4)$$

5.1.2 Registering a Location for a SIP-URI

1. User signs his current location (IP-address+port) for $u1$ with the private key for $u1$ to obtain $s1$:

$$s1 = \text{sign}_{k_{priv,u1}}(ip_{u1}) \quad (5)$$

2. User stores the URI ($u1$), the IP-address+port (ip_{u1}), the signature ($s1$) and the public key for the URI ($k_{pub,u1}$) in the overlay:

$$Chord.Store(u1, ip_{u1}, s1, k_{pub,u1}) \quad (6)$$

5.1.3 Verifying the Authenticity of Location Data

1. Node requests a location for SIP-URI $u2$ from the overlay network:

$$Chord.Request_location(u2) \quad (7)$$

2. Node receives the location binding, signature, and the corresponding public key from the network:

$$Chord.Request_location(u2) \rightarrow (u2, ip_{u2}, s2, k_{pub,u2}) \quad (8)$$

3. Node verifies that the public key sent along belongs to $u2$:

$$f(h(k_{pub,u2}))@domain \stackrel{?}{=} u2 \quad (9)$$

4. If (9) is true, the node verifies the location signature with the public key:

$$\text{verify}_{k_{pub,u2}}(s2) \stackrel{?}{=} ip_{u2} \quad (10)$$

5.2 Notable Properties of the Proposed Scheme

We briefly mention some notable properties of the proposed solution. Generally speaking, our scheme is independent of any network property and can thus be used as a security add-on in almost any scenario:

5.2.1 No central authority

Most notably, our solution enables authentication of messages without the use of a central authority. Thus, it retains the spirit of peer-to-peer computing (no server), and scalability is not a problem.

5.2.2 Verification of message-integrity possible at all routing hops

The verification-procedure can be done at any routing hop. Thus, not only the requesting node can verify message-integrity: Any node in the network can verify the integrity of messages it receives and detect (and possibly drop) falsified messages.

5.2.3 Independency of overlay and routing strategy

The scheme works independent of the DHT used. At this point, only prototypes for P2P-SIP have been developed. Chord has been chosen as the DHT but may not be the final choice for P2P-SIP. Therefore, it is important that any security solution can also be used with a different DHT/overlay protocol. Also, the scheme can be used with iterative, recursive, or more sophisticated routing strategies (as suggested in [16]).

5.2.4 Backwards compatibility

Because the added cryptography is encoded within the SIP-URI, our scheme works with any existing or future specification for P2P-SIP. No new headers or components are added. Hence, the scheme can also be used on top of any existing (Client-Server) SIP deployment.

5.3 Converting the hash of the public key into a SIP-URI

In our scheme, a private key is used to sign the binding of an IP-address to a SIP-URI (5). To prove that this private key indeed belongs to the (owner of the) SIP-URI in the binding, the hash of

² We use RSA, but in principle any asymmetric encryption system can be used.

the corresponding public key (2) is encoded within the SIP-URI. This conversion needs some discussion: First, the SIP-URI must be formed as specified in [1] in order to enable backwards compatibility with existing SIP-implementations. Second, the resulting SIP-URI shall not be too long, so that it can be typed into the interface of a user agent properly (even with limited user interfaces as in hardphones). Thus, a specified function is needed to convert the hash-value h_1 into a string (3). This string then forms part of the SIP-URI (4).

A SIP-URI is generally of the type “user@domain”. The user part can consist of one or more ASCII-characters. In summary, these characters are limited to alphanumeric values (e.g. a-z, A-Z, 0-9) and special reserved characters. Altogether, 79 different ASCII characters may be used for the user part of a SIP-URI (see [1], pp. 147 and [17] for details). Thus, it is possible to encode at most 6 bits in any ASCII-character of the user part of a SIP-URI ($2^6=64 \leq 79$).

We suggest the following algorithm to convert the hash into a string, assuming a 160-bit hash value is used (as with SHA-1, for example). This algorithm forms the function f (3):

1. Use only the leftmost 65 bits of the hash value
2. From left to right, encode every 5 bits into one ASCII character according to a defined mapping table

We suggest using only lower-case letters and digits for the generated URIs. Although more characters are allowed in SIP, this prevents attacks - similar to common phishing websites - where users are tricked into a wrong URI because of case-sensitivity: An attacker could automatically generate public keys until she/he gets a value that hashes to a URI similar to the URI she/he wants to impersonate except for case-sensitivity. Thus, we only use 32 ASCII-characters (26 lower-case letters plus 6 digits). This results in encoding 5 bits into one character ($2^5=32$). Compared to using all possible characters and encoding 6 bits into one character the user part of the resulting SIP-URI is 13 characters/digits long instead of 11 characters/digits ($65:5=13$; $65:6 \sim 11$). We feel that the added security is worth the two characters.

Fig. V shows the procedure of mapping a public key onto a *Cryptographically Generated SIP-URI*. First, the public key is reduced to a 160-bit hash value by a predefined hash function h . Then, the function f is applied to the leftmost 65 bit of the hash-value, mapping every 5 bits from left to right onto a character/digit. The resulting 13 characters/digits are prepended to the domain of the user, separated by the “@” sign.

6. EVALUATING THE PRESENTED APPROACH

In this section, we discuss disadvantages of the presented approach. Furthermore, we relate our approach to other work that considers options for securing DHT messaging.

6.1 Problems with the Presented Scheme

As shown in the previous section, the presented scheme prevents man-in-the-middle attacks on content in P2P-SIP networks: Using this scheme, any node in the network can verify the authenticity of a SIP-URI/location binding message it receives. However, the presented solution introduces some new - mostly practical - hurdles. In this section we list and discuss these issues:

6.1.1 Readability of SIP-URIs

One of the most obvious disadvantages of the solution presented in this paper is the (un)readability of Cryptographically Generated SIP-URIs. As they are computed by converting the hash of a public key into a string, URIs will look cryptic to the user (e.g. sip:k4h1gfhdh5wdd@sip-provider.de).

6.1.2 Associating a SIP-URI with a user

Because URIs are cryptic, there must be a way to reliably associate a user with a Cryptographically Generated SIP-URI. We suggest using existing authentication infrastructure available to users for this purpose: A certified web service using SSL can publish an “online phone book”, mapping users to SIP-URIs. The certificate of the service can be verified by users in their webbrowser. Note that with Client-Server SIP the same problem exists: How can a user be sure that a SIP-URI indeed belongs to the person he intends to speak to, especially if the person is previously unknown to the user?

6.1.3 Attacks on the hash function

The security of our solution relies on some characteristics of the hash function that is used: The hash function must be collision-resistant and second pre-image resistant. Pre-image resistance is not important for our scheme because the public key gets send along with its hash in each message. If at some time the second pre-image resistance of the hash function is broken (for example due to some new sort of attack), an attacker could generate a public/private key-pair with the same hash as the target URI of his attack. This would enable him to sign falsified messages. Since we only use the leftmost 65 bit of the hash, our SIP-URIs are possibly susceptible to second pre-image attacks in the near future. To circumvent this shortcoming, hash extension techniques can be used: A second hash-value is added, where the s leftmost bits must be 0. Using this technique, the effective hash length can be incremented by s bits (see [15] for details).

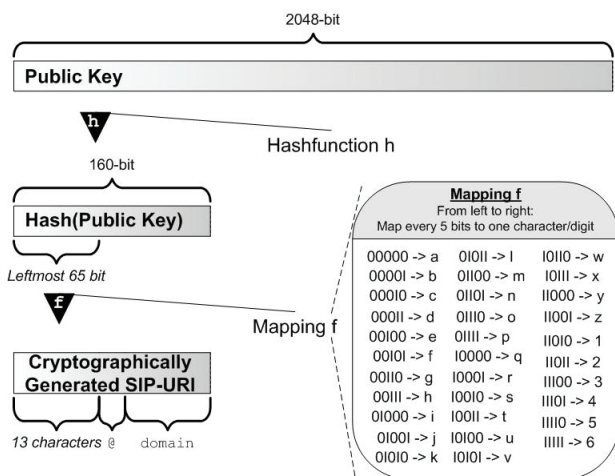


Figure V. Mapping a Public Key onto a Cryptographically Generated SIP-URI

6.1.4 Performance / Denial-of-Service Attacks

Because cryptographic functions are added to the nodes participating in the network, such a network will lose some of its performance. Further, the network will be more susceptible to Denial-of-Service (DoS) attacks: Any node verifying the authenticity of a message must perform cryptographic primitives which consume computational power. An attacker can exploit this by inserting bogus messages into the network to attack the network's availability. However, our technique also prevents some DoS-attacks because each node in the network can detect falsified messages and drop them immediately instead of forwarding them further.

6.2 What is gained?

In any SIP communication, a user is represented by a SIP-URI. We protect the integrity of messages regarding this identity by using Cryptographically Generated SIP-URIs. For these URIs, only the owner of the identity can sign messages. Thus, users can prove their ownership of a SIP-URI.

However, we do not solve the problem of how to bootstrap authentication: How can a caller be sure about the callee's real (physical) identity? Bootstrapping authentication without a central authority or a pre-call trust relationship seems very hard to achieve. We feel it is best to rely on existing authentication infrastructure to achieve this as described in the previous subsection. Even though existing authentication infrastructure (like https-web pages) uses a central authority (and has some problems), this central authority is neither part of the P2P network nor used to authenticate the identity used in P2P-SIP.

With Cryptographically Generated SIP-URIs, users cannot choose the characters in their SIP-URI. Hence, they cannot include some semantic information in their URI which would relate to their real identity. Essentially, this is the price being paid for the ability to prove ownership of a SIP-URI. We feel that the ability to prove ownership of a URI to any entity in the system is worth this drawback.

6.3 Related Work

Some of the work on DHT security (listed in Section III.A) alludes to self-certifying data in P2P systems. Sit and Morris [12] briefly mention self-certifying data as an option to prevent the return of incorrect data from the overlay to the application. Castro et al. [14] discuss self-certifying data as an add-on to their secure routing techniques. They consider self-certifying data as an alternative to their measures in some cases.

CFS [18] is a distributed read-only file system that uses self-certifying data for securing the content stored in a Distributed Hash Table. A cryptographic hash of a file's content is used as the key during insertion and lookup of the file. The publisher of the file signs the root block with his/her private key.

Two important factors distinguish the work in this paper from the work listed above: First, previous work on DHT security primarily considers the availability of content in P2P networks and attacks on node-ID generation. Our work focuses solely on the integrity of the content stored in a P2P network. Second, because the content stored in the network is application specific, our work presents a solution unique to SIP communications. We discuss specific problems of using self-certifying data for this

particular application of a Structured Overlay Network and propose a solution intrinsic to SIP. However, our solution is neither specific to P2P networks nor to Voice-over-IP: it can be used in any application that uses SIP as a signaling protocol and which has a need for protecting the integrity of SIP-URI/location bindings.

7. CONCLUSION

We have presented a solution to protect the integrity of location bindings stored as content in a P2P-SIP network. Our approach uses Cryptographically Generated SIP-URIs as self-certifying data for user registration and location lookup in a P2P network. The proposed scheme requires no central authority, enables verification of message-integrity on all routing hops, and is independent of the DHT or routing strategy being used. Further, it can be used on top of any existing SIP-deployment because all added cryptography is encoded within the SIP-URI.

The proposed scheme has been described in detail. We have shown how our solution prevents man-in-the-middle attacks on content stored in the network. In addition, we have discussed problems that arise when using self-certifying content in a P2P-SIP network and evaluated our approach.

We see the scheme presented in this paper as a step towards a security framework for P2P-SIP. Self-certifying SIP-URIs can be combined with other security measures for P2P-SIP. For instance, they can be used in conjunction with multiple lookup paths [16] or other secure routing techniques [14].

8. REFERENCES

- [1] J. Rosenberg et al., "SIP: session initiation protocol", RFC 3261, 2002
- [2] K. Singh, H. Schulzrinne, "Peer-to-Peer Internet Telephony using SIP", Proc. of the international workshop on Network and operating systems support for digital audio and video, Stevenson, Washington, USA, 2005, pp. 63-68
- [3] D.A. Bryan, B.B. Lowekamp, C. Jennings, "SOSIMPLE: A Serverless, Standards-based, P2P SIP Communication System", Proc. of the International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications, Orlando, FL - June 15, 2005
- [4] D.A. Bryan, B.B. Lowekamp, C. Jennings, "A P2P Approach to SIP Registration", internet draft, work in progress, March 2006, <http://www.p2psip.org/drafts/draft-bryan-sipping-p2p-02.html>
- [5] D.A. Bryan, E. Shim and B.B. Lowekamp, "Use Cases for Peer-to-Peer Session Initiation Protocol (P2P SIP)", internet draft, work in progress, November 2005, <http://www.p2psip.org/drafts/draft-bryan-sipping-p2p-usecases-00.html>
- [6] J. Posegga, J. Seedorf, "Voice over IP: Unsafe at any Bandwidth?", Proc. of Eurescom Summit 2005 - Ubiquitous Services and Applications, Heidelberg, April 27-29, 2005, pp.305-314
- [7] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, "A Scalable Content-Addressable Network", Proc. of SIGCOMM '01, San Diego, USA, August 27-31, 2001

- [8] I. Stoica et al., "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications", IEEE/ACM Transactions on Networking, Vol. 11, No. 1, February 2003
- [9] A. Rowstron, P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems", Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms, Heidelberg, Germany, November 2001
- [10] Ben Y. Zhao et al., "Tapestry: A Resilient Global-Scale Overlay for Service Deployment", IEEE Journal on Selected Areas in Communications, Vol. 22, No.1, January 2004
- [11] R. Steinmetz, S. Götz, S. Rieche, "Distributed Hash Tables", in P2P Systems and Applications, R. Steinmetz and K. Wehrle (Eds.), LNCS 3485, pp.79-93 & pp. 95-117, 2005
- [12] E. Sit, R. Morris, "Security Considerations for Peer-to-Peer Distributed Hash Tables", 1st International Workshop on Peer-to-Peer Systems (IPTPS), March 2002
- [13] J. R. Douceur, "The sybil attack", Proc. of the IPTPS02 Workshop, Cambridge, MA (USA), March 2002.
- [14] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, D.S. Wallach, "Secure routing for structured peer-to-peer overlay networks", Proc. Of the 5th Usenix Symposium on Operating Systems Design and Implementation, Boston, MA, December 2002
- [15] T. Aura, "Cryptographically Generated Addresses (CGA)", Proc. of Information Security, 6th International Conference, ISC 2003, Bristol, UK, October 1-3, 2003, LNCS 2851, pp.29-43
- [16] G. Danezis, Ch. Lesniewski-Laas, M.F. Kaashoek, R. Anderson, "Sybil resistant DHT routing", Proc. of ESORICS 2005, pp. 305-318
- [17] D. Crocker, P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997
- [18] F. Dabek, M.F. Kaashoek, D. Karger, R. Morris, I. Stoica, „Wide-area cooperative storage with CFS", Proc. of SOSP '01, October 21-24, 2001, Banff, Canada