Performance Analysis of Identity Management in the Session Initiation protocol (SIP)

Yacine Rebahi, Nguyen Tuan Minh, Ge Zhang

Abstract

The Session Initiation Protocol (SIP) is a standard for managing IP multimedia sessions in the Internet. Identity management in SIP is a crucial security field that deals with identifying users in SIP networks and controlling their access to the corresponding resources. RFC 4474 describes a mechanism, based on certificates, for dealing with the SIP users identities. This RFC recommends the use of the RSA algorithm as it is currently the most popular public key cryptography system. The proliferation of small and simple devices as well as the need to increase the capacity of the SIP servers to handle the increasing VoIP traffic will make continued reliance on RSA more challenging over time. In this paper, we describe our implementation of the current RFC 4474, our integration of elliptic curves cryptography into this RFC and show that the corresponding performance is much more significant than the one where RSA is used. This paper can be considered as a first step in standardizing the use of elliptic curves in the identity management for SIP.

Index Terms

Identity Management, RFC 4474, SIP, RSA encryption, Elliptic Curves, ECDSA, SPIT.

1 Introduction

IP telephony is an emerging technology enabling a range of new service possibilities. Although, various protocols, underlying this technology, are already deployed, the Session Initiation Protocol (SIP) [4] seems to be the standard that is being widely adopted by the different Voice over IP (VoIP) communities. For a successful deployment of the SIP-based technologies, the services that the VoIP providers offer to their customers need to reach a certain security maturity level.

- *Y. Rebahi and N. T. Minh are with Fraunhofer Institut Fokus, Kaiserin Augusta Allee 31, 10589, Berlin, Germany. E-mail: {yacine.rebahi,tuan.minh.nguyen}@fokus.fraunhofer.de*
- *G. Zhang is with Karlstad University, Universitetsgatan 2, 65188, Karlstad, Sweden. E-mail: ge.zhang@kau.se*

One of the threats that is posed to VoIP technologies, is the weakness of the corresponding identity management. The latter is, in general, a broad field in security that deals with identifying individuals in a system and controlling their access to resources within that system by associating user rights and restrictions with the established identity.

In SIP, the existing security mechanisms are inappropriate for assuring the identity of the end users that originate SIP requests, especially in an interdomain context [5]. This has lead the authors of [5] (known as RFC 4474) to define a new mechanism based on certificates for securely identifying originators of SIP messages. This mechanism enhances the authentication operation by defining two new SIP header fields, *Identity*, for providing a signature used for validating the identity, and *Identity-Info*, for conveying a reference to the certificate of the signer. The work achieved in RFC 4474 is based on the RSA algorithm.

RSA is the most widely used public key technology today, however, the emergence of small devices with some constraining technologies will make the use of RSA in the future questionable. In fact, handheld devices such as PDAs and mobile phones or some other sophisticated devices such as sensors will not have enough resources to provide adequate levels of security if RSA is used. In addition to that, VoIP networks are growing very fast, so a larger volume of VoIP traffic is expected and this will significantly increase the cost of supporting RSA computations. Relying on the RSA algorithm in the future means increasing the corresponding keys sizes in order to cope with the improvement of processors speeds that the attackers might use to attack this algorithm. Unfortunately, increasing the keys sizes will worsen the performance. The just mentioned drawbacks show the need for public key cryptography algorithms that can (first) be deployed on small devices while in the same time ensuring a strong cryptography, (second) improve the capacity of the servers to handle secure connections. The Elliptic Curves Cryptography (ECC) seems to fulfill these requirements. In this paper, we describe on the one side, the integration of the elliptic curves cryptography into the identity management mechanism defined in RFC4474, and on the other side, the implementation of this RFC using both algorithms (RSA and ECC). A performance comparison is also provided.

It should be mentioned that the idea of replacing RSA with some Elliptic Curve Cryptography (ECC) system in the context of SIP identity management does not come form the fact that RFC 4474 is not well specified. This idea is not recent and was already discussed in the context of,

smart cards [1], Radio Frequency IDentification (RFID) technologies [2], and Enhanced Embedded Security [3]. The idea of replacing RSA with ECC was motivated mainly by the superior performance and the high difficulty to be cracked that an ECC system presents with comparison with the RSA algorithm.

The rest of this paper is organized as follows: section 2 gives a brief overview of RFC 4474, however sections 3 and 4 describe respectively the RSA algorithm and the elliptic curve cryptography system and compares between them. Section 5 discusses the implementation undertaken in this work. In section 6, a performance analysis is provided and in section 7.1, a hint of how RFC4474 can be used for detecting SPIT is discussed. Finally, section 8 concludes the paper.

2 IDENTITY MANAGEMENT IN SIP

RFC 4474 tries to identify senders by the set of SIP servers they use to send their requests. The idea behind the domain verification techniques is to get some information from the sender's domain that helps to verify that this domain is exactly the source of the message. As a consequence, a SIP message issued, apparently, from *iptel.org* could not be sent by a user outside *iptel.org*. RFC 4474 assumes that there is a trustful third party that is able to issue certificates, which could be in particular acquired by the SIP proxies in the network. The mechanism described in RFC 4474 is depicted in figure 2 and works as follows,

- Before being able to establish a request, the sender needs to authenticate himself against the SIP proxy of his domain.
- If the authentication succeeds, the SIP proxy in the sender's domain computes a hash on some header fields and body of the SIP message, signs it with its private key and inserts the result as a new header field *Identity* in the same SIP message. The SIP proxy also creates a second header field *Identity-info* and inserts the URL of the certification authority in it. After that, the SIP message is forwarded further. An example of signed SIP INVITE request is illustrated in figure 1.
- When the SIP proxy on the recipient side, receives the SIP message, it uses the URL within the *Identity-info* field to retrieve the public key used by the SIP proxy in the originating domain. This key is used to verify the signature that the originating SIP proxy has inserted

in the forwarded SIP message.

```
INVITE sip:bob@iptel.org SIP/2.0
Via: SIP/2.0/UDP 193.11.155.6:5061 ;branch=z9hG4bk56edct
From: sip:alice@kau.se
To: sip:bob@iptel.org
Cseq: 1 INVITE
Contact: <sip:alice@kau.se:5067;transport=udp>
Identity:"l8wGnkhupOOGoTfjThSlE4G+my1xofK8nxvY1R0FOkjWrkbF
RRVPB+503H5EfZlOiycwgGweD0X1wpshxTDjioOj7kN2I9ViLHKbeOxp
H+VahW0TRRpw5wUToxyA91WpanDolCW90kJUWN7LrqPqR7JossbXInc+
jQfUttNg4="
Identity-info: <https://www.kau.se/my.cert>; alg=rsa-sha1
...
```

Fig. 1. An example of a signed SIP INVITE request with added *Identity* and *Identity-info* headers

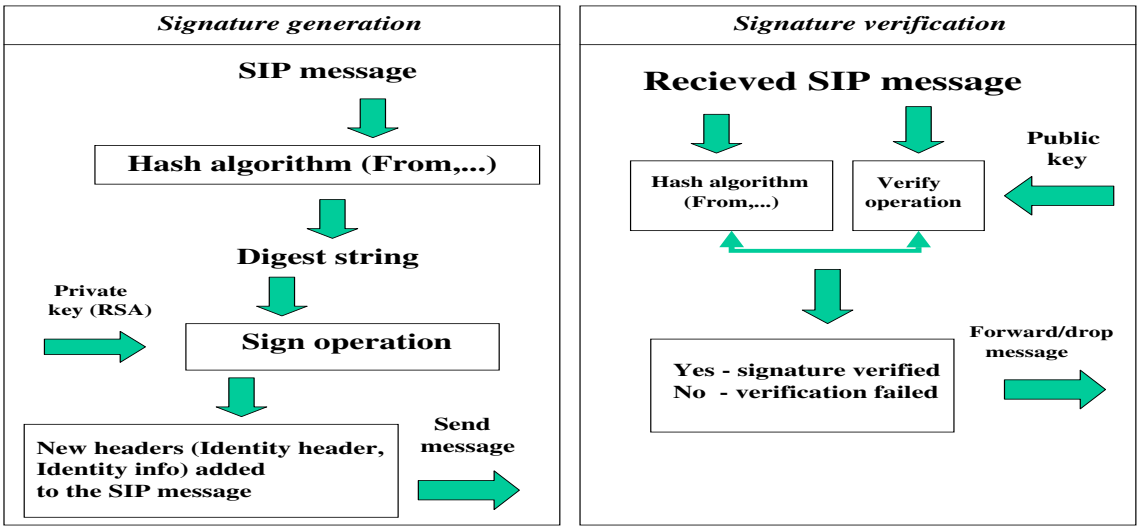

Fig. 2. The authenticated identity management operation

3 THE RSA ALGORITHM

The Rivest-Shamir-Adleman (RSA) algorithm [6] is one of the most popular public-key encryption methods. This algorithm can be used both for encryption and for signing.The importance of this algorithm resides in the fact that it underlies well known systems such as Secure Sockets Layer (SSL) [7] and Private Communication Technology (PCT) [8], as well as many of the firewall and network security products currently available.

The RSA algorithm is considered to be secure when sufficiently long keys are used (512 bits is insecure, 768 bits is moderately secure, 1024 bits is secure and 2048 bits should remain secure

for the foreseeable future).

The security of RSA relies on the difficulty of factoring large integers (100-200 digits), these large integers are the mathematical relationship between the public and private keys. The RSA algorithm can be used in the case of signatures (which is our focus in this paper) as follows,

Key generation algorithm

1) Generate two large random primes, p and q, of approximately equal size such that their product $n = p.q$ is of the required bit length, e.g. 1024 bits
2) Compute $n = p.q$ and $\phi = (p-1).(q-1)$
3) Choose an integer e, $1 < e < \phi$, such that $GCD(e, \phi) = 1$. The latter is the Greatest Common Divisor (GCD) of e and ϕ
4) Compute the secret exponent d, $1 < d < \phi$, such that $e.d \equiv 1 \ mod \ \phi$
5) The public key is (n, e) and the private key is (n, d). The values of p, q, and ϕ should also be kept secret.

Digital signing

Sender A does the following,

1) Creates a message digest of the information to be sent.
2) Represents this digest as an integer m between 0 and $n-1$.
3) Uses the private key (n, d) to compute the signature $s \equiv m^d \ mod \ n$.
4) Sends this signature s to the recipient B.

Signature verification

Recipient B does the following,

1) Uses sender A's public key (n, e) to compute the integer $v \equiv s^e \ mod \ n$.
2) Extracts the message digest from this integer.
3) Independently computes the message digest of the information that has been signed.
4) If both message digests are identical, the signature is valid.

4 ELLIPTIC CURVE CRYPTOGRAPHY

A cryptosystem often requires the use of an algebraic group. A group is a set of elements with custom-defined arithmetic operations on those elements such that certain conditions are fulfilled. Elliptic curves may be used to form elliptic curve groups. Elliptic curve ([9]) groups

used in cryptography are defined over two kinds of Galois fields: $GF(p)$ where p is a prime, and $GF(2^m)$ where each element is a binary polynomial of degree m that can be represented as an m-bit string since each coefficient is either 0 or 1. It is easier to illustrate group operations by examining curves over real numbers.

An elliptic curve over real numbers is a plane algebraic curve defined by an equation of the form $y^2 = x^3 + ax + b$ where a and b are real numbers, with the condition of being non singular, i.e. the discriminant $-16(4a^3 + 27b^2) \neq 0$. An elliptic curve group over real numbers consists of the points on the corresponding elliptic curve, together with a special point O called the point at infinity (for more details, we refer to [9]) .

The security of Elliptic Curve Cryptography (ECC) relies on the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP). While conventional public-key cryptosystems (RSA, Diffie-Hellman (DH), etc) operate directly on large integers in $(Z/pZ)^*$, an ECC operates over points on an elliptic curve. Hence, an attacker needs to resolve the discrete logarithm problem not on $(Z/pZ)^*$ but on some elliptic curve groups. In addition, it seems that using shorter key sizes for elliptic cryptosystems gives similar security as much larger keys that might be used in cryptosystems based on the discrete logarithm problem and integer factorization. As recommended by the National Institute for Standards and Technology (NIST) [10], RSA and Diffie-Hellman use 1024-bit keys while ECC provides equivalent security for a key size of about 160 bits. Table 1 presents the equivalent key sizes for ECC and RSA.

ECC Key Size (bits)	RSA Key Size (bits)	Key Size ratio
160	1024	1:6
224	2048	1:9
256	3072	1:12
384	7680	1:20
512	15360	1:30

TABLE 1
Equivalent key sizes for ECC and RSA

Smaller key size means faster computations, lower power consumption, as well as memory and bandwidth savings. These characteristics not only make ECC appealing for low-computational devices but also for reducing the computational burden on secure servers.

4.1 Elliptic Curve Digital Signature Algorithm (ECDSA)

In the context of this paper, the Elliptic Curve Crytography is a means for generating signatures. As a consequence, the Elliptic Curve Digital Signature Algorithm (ECDSA) will be used. First an elliptic curve E is defined over $GF(p)$ or $GF(2^k)$ with large group of order n and a point P of large order is selected and made public to all users. Then, the following key generation primitive is used by each party to generate the individual public and private key pairs. Furthermore, for each transaction the signature and verification primitives are used. We briefly outline the Elliptic Curve Digital Signature Algorithm (ECDSA) below,details of which can be found in [11].

ECDSA key generation

The user A follows three steps

1) Select a random integer $d \in [2, n-2]$
2) Compute $Q = d.P$
3) The public and private keys of the user A are (E, P, n, Q) and d, respectively

ECDSA signature generation

The user A signs the message m using three steps:

1) Select a random integer $k \in [2, n-2]$
2) Compute $k.d = (x_1, y_1)$ and $r = x_1 \ mod \ n$.

   If $x_1 \in GF(2^k)$, it is assumed that x_1 is represented as a binary number.

   If $r = 0$ then go to step 1
3) Compute $k^{-1} \ mod \ n$
4) Compute $s = k^{-1}(H(m) - dr) \ mod \ n$.

   Here H is the secure hash algorithm SHA.

   If $s = 0$, go to step 1.
5) The signature for the message m is the pair of integers (r, s)

ECDSA signature verification

The user B verifies A's signature (r, s) on the message m by applying the following steps:

1) Compute $c = s^{-1} \ mod \ n$ and $H(m)$
2) Compute $u_1 = H(m)c \ mod \ n$ and $u_2 = rc \ mod \ n$
3) Compute $u_1.P + u_2.Q = (x_0, y_0)$ and $v = x_0 \ mod \ n$
4) Accept the signature if $v = r$

5 IMPLEMENTATION

The RSA and ECDSA algorithms are implemented on top of the Enhanced Identity Management (EIM) mechanism described in RFC 4474. As recommended in this RFC, the EIM implementation is based on the RSA algorithm. To compare the performance of ECDSA against the one of RSA, we have modified the EIM implementation in order to support the ECDSA algorithm. This has been achieved by adding some ECC procedures from the OpenSSL APIs. OpenSSL [7] is an open source implementation of the SSL and TLS protocols. The core library implements the basic cryptographic functions and provides various utility functions.

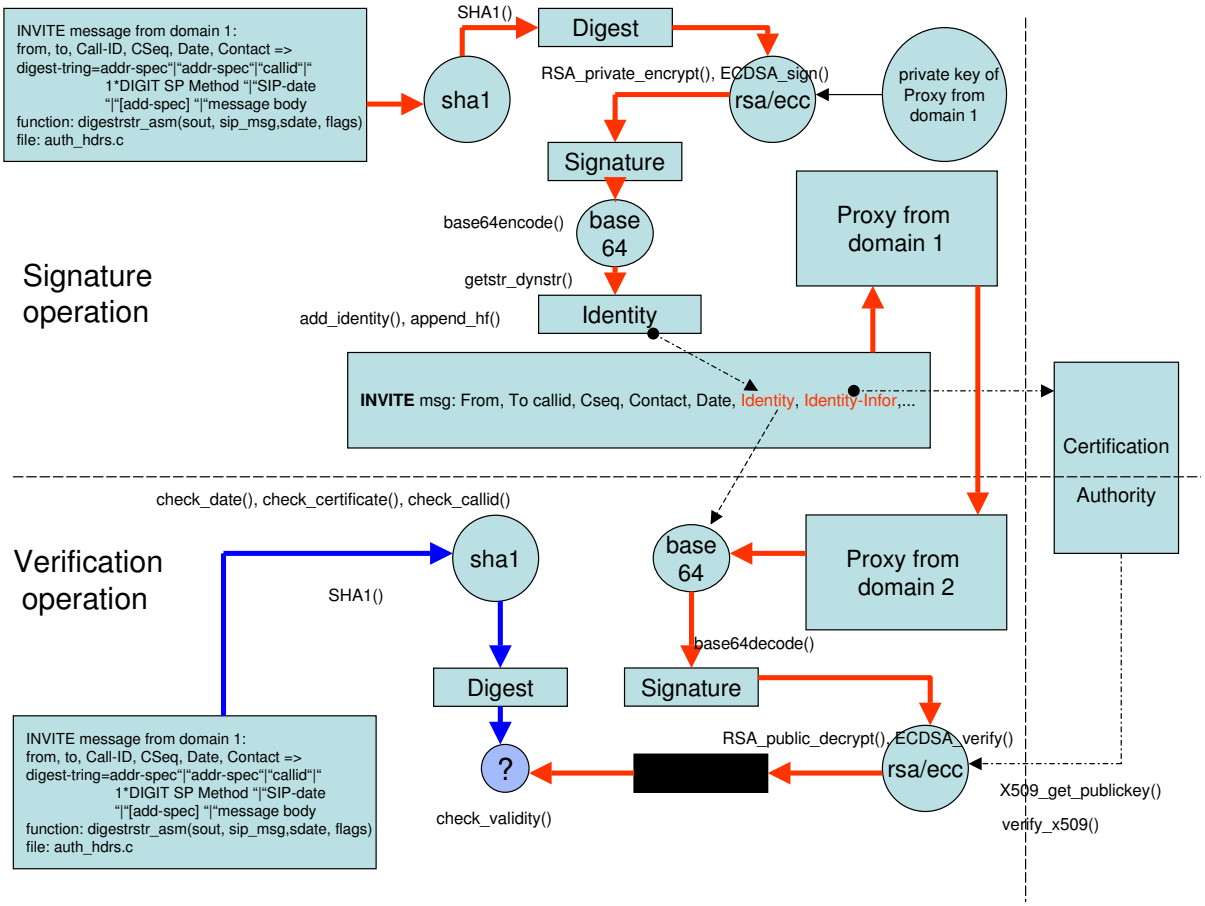

Fig. 3. RFC 4474 implementation

Our implementation of RFC 4474 is depicted in figure 3. The first step in this RFC is the registration phase. The SIP request sender needs to register with the SIP server of his domain using a Digest authentication scheme as described in RFC 2361 [4]. When the registered user sends a SIP request such as an INVITE, the later is intercepted by the SIP proxy of the user's domain, and the EIM mechanism checks the presence and validity of the "Date" header field,

messages with more than 10 minutes of delay should be rejected; the same applies to dates that reflect invalid certificates. The EIM mechanism also recalculates the ”Content-lenght” field, which is an important parameter during hash calculations.

Afterwards, a subset of the header fields from the SIP INVITE method are concatenated and later used to calculate a cryptographic hash by means of the SHA-1 secure hash algorithm [13]. More precisely the fields used during the hash calculation are:

digest-string = addr-spec ”—” addr-spec ”—” callid ”—” 1*DIGIT SP Method ”—” SIP-date ”—” [addr-spec] ”—” message-body

where the first ”addr-spec” field in the concatenated vector corresponds to the ”From” header field, the second to the ”To” field, and the third to the ”Contact” header field, in case this is present in the SIP Request. The ”callid” field corresponds to the ”Call-Id” header information, whereas the ”1*DIGIT” and ”Method” portions are extracted from the ”CSeq” header field. The ”message-body” includes all components of multipart message bodies, and may be left blank in case the message does not contain any message body.

The digest string is then encrypted with the private key of the SIP proxy of the originating domain. After encryption, the signature is base64 encoded and added to the header field *Identity*. Base64 encoding allows encoding arbitrary sequences of octets to a format that can be expressed in short lines of 7-bit characters that represent a subset of the ASCII characters included in all codepages.

The header field *Identity-Info* is then filled with two elements, (first) an URI from which the certificate used for encryption can be acquired and (second) the type of algorithm that has been used for encryption.

The SIP request is then forwarded normally until it reaches the SIP proxy of the recipient domain. This SIP proxy will fetch, if necessary, the certificate indicated in the *Identity-Info* field. Then it will check whether the domain name in the certificate and in the ”From” header of the SIP request are the same. If successful, it recalculates the cryptographic hash from the SIP header parameters. On the other hand, it also base64 decodes the *Identity* field, decrypts it with aid of the certificate’s public key, and compares the result with the cryptographic hash calculated beforehand.

As mentioned earlier, RFC 4474 only supports the RSA algorithm. In order to provide the EIM

mechanism means of knowing which encryption scheme to use, the *Identity-Info* field containing the type of algorithm to be used is inserted. For RSA, the single value of this field is: ”rsa-sha1”. To support different cryptographic primitives this header needs to be extended, namely with ”ecc-sha1” in our case. This extension is the first step of a standardisation procedure that we have been undertaking. We belive that integrating ECC cryptography into identity management in SIP is crutial and we intend to write an internet draft specifying how this integration can be achieved.

Since our implementation opens the door to different encryption algorithms using for instance secure cryptographic hashes from the SHA-2 family (such as SHA-256, SHA-384 or SHA-512), we also would like to propose a method for checking the availability of encryption algorithms. In fact, RFC 4474 specifies that all the servers must implement the ”rsa-sha1” algorithm, then, in case two Signing/Verification servers (supporting ECC cryptography) need to interact and only the signing server supports the ”ecc-sha1” algorithm, then the verification server must send an error response of the form ” Signature Algorithm: Not Supported”. As soon as the signing server receives this response, it will recalculate the *Identity* and *Identity-Info* header values with the failback ”ecc-sha1” algorithm and resend the SIP request.

6 EXPERIMENTS

This section summarizes our testbed setup and the experimental results regarding the performance of using both RSA and ECDSA algorithms within the EIM mechanism.

6.1 Testbed setup

To investigate the impact on the performance of SIP services introduced by EIM mechanism, we built our testbed involving two SIP domains. One is named as the caller domain, which actively initiates SIP transactions. Another is named as the callee domain, which receives SIP requests from the caller domain and makes SIP responses. The components in both domains are introduced as follows:

- SIP proxies: The proxies are implemented according to the mechanisms defined in RFC 3261 and RFC 4474. We employ SIP Express Router (SER) [12] for this task. The proxy is equipped with a certificate cache to store downloaded certificates. On top of SER, the EIM

mechanism, specifying the RFC 4474, is implemented as a separate module. Although, our implementation is SER based, it can easily fit within any other SIP platform. The SIP Express Router (SER) [12] and the EIM mechanism are used for generating the signatures as well as verifying them. In order to have a good performance, both SER proxies, that are going to be used for signature generation and verification, run on Linux machines endowed with 2 CPUs and with the following features: Pentium Core TM 2 (T5600), CPU 1.83 GHz, Cache 2048 kB.

- A web server in the caller domain: A web server is deployed in the caller domain to provide certificate downloading.[1] The URL of this certificate will be added into the *identity-info* field for all outgoing INVITE requests from the caller domain. Then, the proxy in the callee domain will download the certificate from this web server. The web server is implemented based on Apache HTTP server [21].
- SIP user agents: The SIP user agents used in our tests are based on SIPp [14]. It is an open source software for generating SIP signalling messages. SIPp is an appropriate tool for testing especially that it allows the generation of huge SIP traffic volume and provides a simple configuration interface for the SIP URLs, number of calls per second, calls duration, errors, etc. The SIPp clients on both side (caller and callee domains) are installed on Linux machines with the following features: Pentium 4, CPU 3.00 GHz, cache 1024 kB.

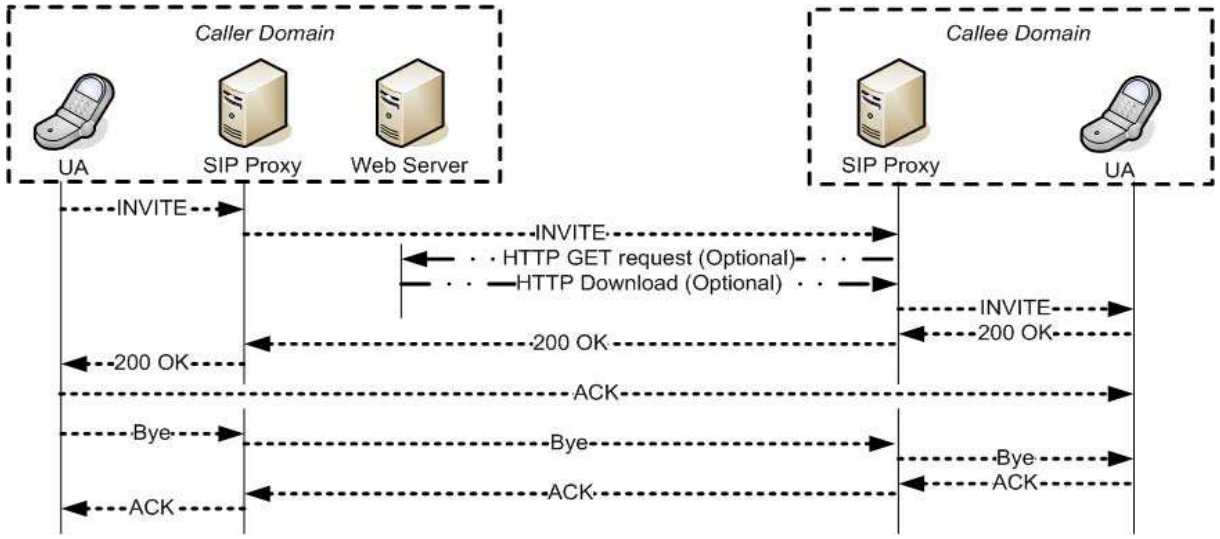

Fig. 4. The testbed configurations with the procedure of establishing and terminating calls

1. In this testbed, we assume that the callee domain will not generate requests to the caller domain. Therefore, there is not need to setup a web server in the callee domain.

The EIM on the SIP proxies can be enabled or disabled. To enable the EIM, a pair of certificate and private key will firstly be generated for the caller domain by a Certification Authority (CA). The CA is supposed to be the key trust entity, which would be for reallife setups a certificate provider like e.g., Versign. For the functionality tests, we built a local CA using OpenSSL. Then, we configured the UA in the caller domain to generate INVITE requests targeting to the UA in the callee domain through the proxies. These INVITE requests will be signed by the caller proxy using its private key according to the steps we introduced above. When the callee proxy receives these INVITE requests, the proxy will fetch the certificate of the caller proxy for message verifying in two ways: (1) If the certificate cannot be found in the local cache, the proxy will download it from the web server in the caller domain. And then, the certificate will be saved in the local cache for a period of time; (2) If the certificate can be found locally, the certificate will be retrieved from the local cache. [2] These requests are supposed to be successfully verified by the proxy in the callee domain and forwarded to the callee UA. The callee UA is configured to make responses with a 200 OK message for each request. As soon as the call is established, the UA in the caller domain sends a BYE request and the UA in the callee domain replies a final ACK repsonse. The testbed and the whole transaction is illustrated in figure 4. *In this experiment, we define that a call (calling dialog) is successfully accomplished as long as the UA in the caller domain sent an INVITE request and received the final ACK response.* We are interested to know how many calls can be accomplished with the EIM enabled or disabled on the SIP environment. In this way, we can profile the performance impact introduced by the identity management mechanism.

6.2 Testing and results

We started first by checking how many SIP calls can be handled on the testbed when the EIM module is disabled. For this purpose, different scenarios were investigated, the UA floods with a huge SIP traffic volume starting from a minimal value (10 call/s) and keeps increasing until it reaches the maximum value (10.000 call/s). We noticed that 920 call/s can be handled at most in the testbed. This maximal value is taken into account when the performance of the EIM mechanism is enabled.

2. This paper only focuses on the performance impact introduced by the cryptographic algorithms in RFC4474, not the certificate downloading. In our experiments, since the certificate is downloaded over a local network instead of the Internet, the time cost for downloading can be neglected and will not impact the experiment result.

For measuring the performance of the mentioned module, we used 3 different RSA keys (1024, 2048, 3072)-bits and their equivalents in ECC. This choice was based on the fact that the keys of size 2048 bits and 3072 bits are supposed to be unbreakable in the future. Thus, a high security level is assumed to be guaranteed by these keys and there is no need to use longer keys in our experiments. The performance measurements for these operations are illustrated in table 2. We observe that when the RSA key 1024 bits is used, the maximum number of calls that the EIM module can handle is only 123 call/s, which means almost 13,4% of the maximal value (920 call/s) established without the use of this module. This can be explained by the fact that RSA signature generation requires a lot of processing. This first measurement value decreases dramatically when the key size increases to 2048 bits and then to 3072 bits. From 123 call/s, the number of the established calls decreases to 28 call/s and 12 call/s in case of 2048 bits and 3072 bits respectively. This shows the poor performance of the RSA algorithm when the corresponding keys are longer.

Table 2 also provides the performance results for the ECDSA algorithm. One can see clearly that replacing RSA with ECDSA improves significantly the number of established SIP transactions. This improvement ranges from 2.14 times comparing ECDSA-160 with RSA-1024, through 4.6 times comparing ECDSA-224 with RSA-2048, up to 8.7 times comparing ECDSA-256 with RSA-3072. These results suggest that the improvement brought by the integration of ECDSA into the identity management in SIP can be seen as a function that increases faster than the exponential function $f(n) = 2^n$ (figure 5). This shows the significant performance of ECDSA over the RSA algorithm.

ECC Key Size	RSA Key Size	Nb Call/s (RSA)	Nb Call/s (ECC)	Nb_ECC/Nb_RSA
160	1024	123	263	2.14
224	2048	28	128	4.6
256	3072	12	104	8.7

TABLE 2
Performance results of RSA and ECDSA

7 THE SECURITY ISSUES AFFECTED BY SIP IDENTITY MANAGEMENT

The SIP identity management mechanism in RFC 4474 is designed to provide a high level security for SIP VoIP services. It can be used to assure the *source integrity* to prevent the identity

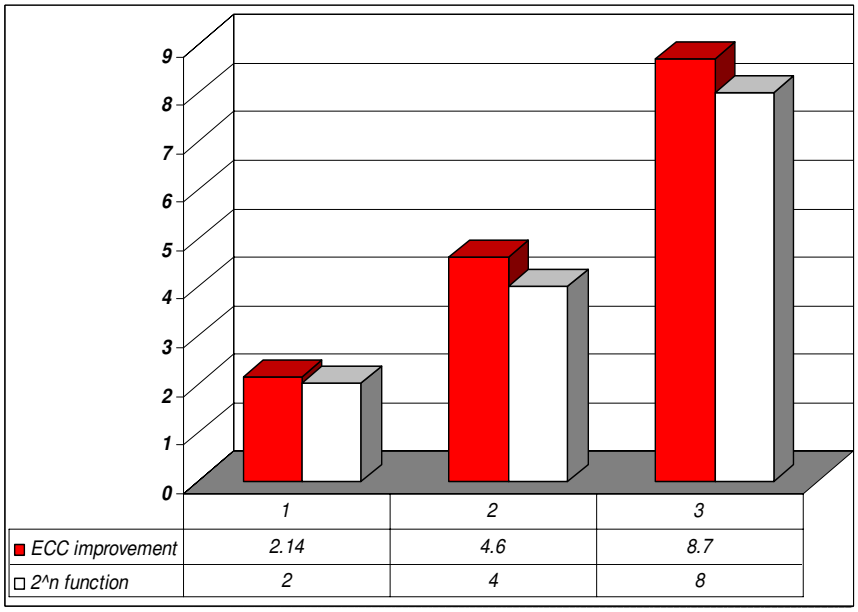

Fig. 5. ECDSA vs. RSA performance

fraud, especially for the inter-domain SIP requests. Unfortunately, however, the efficiency of this mechanism is low since it consists of certificate downloading and cryptographical computations. Considering the significance of the performance for VoIP services, some new security risks can be raised by this mechanism as well. Thus, the SIP identity management mechanism is a double-edged weapon. In this section, we will analyze the advantage and disadvantage of deploying this mechanism respectively.

7.1 The contribution of SIP identity management

The session Initiation Protocol (SIP) is becoming the first standard for managing IP multimedia sessions. Similar to most of email protocols currently used, SIP can suffer from the spam problem [15]. The latter refers in general to any unsolicited information sent to any recipient without its consent. As the spam information is without interest for the majority of people to whom this information is sent, the spammers try to utilize widely used carrier tools that have extremely low cost in order to guarantee high gains. Few years ago, Fax was the main tool used for sending spam, however, the current favorite tools utilized for transmitting spam information are emails and phone calls. For the future, IP telephony seems to be the adequate means for sending spam because of the low cost of the Internet connections and the convergence of data

and voice provided by the corresponding voice over IP protocols [15], [16], [17]. We believe that in the future most of SPIT (SPAM over Internet Telephony) is going to be sent through open relay servers as the latter offer to spiters the possibility to hide their identities. A SIP open relay is a SIP proxy that accepts unauthorized third party relays of SIP messages even they are not destined to its domain. Instead of going through his outbound SIP proxy, a spiter can use these relays to route large volumes of unwanted SIP messages without being detected. SIP open proxies are another kind of third-party relay, however, they are not specifically used for SIP, they could also route some other protocols requests. In fact, SPIT can be seen as an identity management problem [16] or at least a subject to identity management solutions, for instance black and white lists. The RFC 4474 that we have implemented and which deals with identity management in SIP can play a crucial role in mitigating SPIT activities. Indeed, the mechanism described in this RFC ensures that any SIP user utilizes his outbound SIP proxy to establish calls or send instant messages. As a result, the use of open relay proxies is not possible and the SIP servers used for sending SPIT will be blocked, thus, the SPIT activities will reduced significantly.

7.2 The new threats introduced by SIP identity management

Some new threats may be introduced by the SIP identity management in RFC 4474. One serious problem among them is Denial of Service (DoS), which aims to minimized the availability of services to legal users. Considering the cost on infrastructures, it is unlikely for a SIP service provider to offer services with unlimited capacity. Similar to other online applications, SIP service providers deploy SIP servers by assuming a statistic model for the perspective usage (for example, the service can be assured as long as the request rate is no more than α calls/s). It means that an attacker can successfully mount a DoS attack if he floods the server with numerous bogus requests and makes the actual request rate is more than α. It is fairly clear that the higher of α, the more difficult for attack because the attacker has to produce enough requests. Unfortunately, however, the α can be reduced considerably after the SIP identity management deployed. Our experiments above have already shown that the throughput of a SIP proxy can be reduced from 920 calls/s to 12 calls/s when the mechanism is adopted. In this way, an attacker can easily generate 12 calls/s to exclusively occupy the service. Thus, the legal users cannot access the service anymore.

Similarly, the DoS can be caused not only by the step in cryptographical computation, but also by the step in certificates downloading. As we introduced before, a SIP proxy has to download the certificate for message authentication. In the real world application, the time cost of certificate downloading should be taken into account. A recent research [22] has been focused on a new vulnerability in which an attacker can make a SIP proxy to communicate with a high-latency web server for the task of certificates downloading. As a result, the victim proxy has to be waiting for the downloading and temporarily unable to handle other requests. One potential approach to eliminate this kind of threats is to employ certificate cache in case of repeatedly downloading. However, as some experiments shown in [23], a certificate cache can disclose the calling history of a domain to the public. The calling history can be regarded as a kind of sensitive information in some situations. An attacker can then passively observe the Round Trip Time (RTT) of several requests to profile the calling history of a domain.

To enable the identity management mechanism to be widely adopted, these security risks introduced by this mechanism should be tackled.

8 Conclusion

This paper discusses the integration of Elliptic Curve Cryptography (ECC) into SIP identity management schemes. In fact, RFC 4474 that describes a certificate-based mechanism for dealing with SIP users identities is implemented using RSA as well as ECDSA. Our experiments show the superiority of ECDSA over RSA in terms of performance. Due to its computational efficiency, ECDSA can be used in constrained environments where traditional public key mechanisms are impractical. Further, the paper also analyze the security issues related to the identity management mechanism. Although this mechanism is helpful to authenticate the SIP identities, the performance of it poses security threats to SIP services. Thus, it is necessary to optimize the performance of this mechanism from different aspects.

Acknowledgments

The authors would like to mention that this research work is party funded by the SPIDER project [18]

REFERENCES

[1] H. Pietileinen, "Elliptic Curve Cryptography on Smart Cards", link: http://henna.laurikari.net/Dippa/di.pdf

[2] Is Elliptic Curve Suitable to Secure RFID Tags, link: http://www.iaik.tugraz.at/research/krypto/events/RFID-SlidesandProceedings/Wolkerstorfer-ECC%20and%20RFID.pdf

[3] J. Krasner, "Using Elliptic Curve Cryptography for Enhanced Embedded Security", link: http://www.embeddedforecast.com/EMF-ECC-FINAL1204.pdf

[4] J. Rosenberg et al, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[5] J. Peterson t al, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)"', RFC 4474, August 2006

[6] P. Riikonnen, "RSA Algorithm", link: http://silcnet.org/priikone/docs/rsa.pdf

[7] The OpenSSL project, link: www.openssl.org

[8] The Private Communication Technology (PCT) Protocol, link: http://graphcomp.com/info/specs/ms/pct.htm

[9] J. H. Silverman, "An Introduction to the Theory of Elliptic Curves", link: http://www.math.brown.edu/ jhs/ Presentations/WyomingEllipticCurve.pdf

[10] The National Institute for Standards and technology (NIST), link: http://www.nist.gov/

[11] IEEE P1363. Standard specifications for public key cryptography. Draft version 7, September 1998

[12] The SIP Express Router, link: http://www.iptel.org/ser

[13] D. Eastlake et Al, "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, September 2001

[14] SIPp, link: http://sipp.sourceforge.net

[15] J. Rosenberg et al, "The Session Initiation Protocol (SIP) and Spam", draft-ietf-sipping-spam-02, March 6, 2006

[16] Y. Rebahi et Al, "SIP Service Providers and the Spam Problem", In Proceedings of the 2nd Workshop on Securing Voice over IP, June 2005

[17] Y. Rebahi et Al, "SIP Spam Detection", In the Proceedings of the IEEE International Conference on Digital Telecommunications (ICDT06), Cap Esterel, France, August 21-31, 2006

[18] SPIDER Project, link: http:www.projectspider.net

[19] V. Gupta et Al, "Integrating Elliptic Curve Cryptography into the Web's Security Infrastructure", In Proc of WWW2004, May 17-22, 2004, New York, USA

[20] M. Aydos et Al, "Implementing Network Security Protocols based on Elliptic Curve Cryptography", InProc of 4th Symposium on Computer Networks, May 20-21, 1999, pp 130-139, Istanbul, Turkey.

[21] Apache HTTP server, link: http:httpd.apache.org

[22] G. Zhang et Al, "Blocking attacks on SIP VoIP proxies caused by external processing", In Special Issue on Secure Multimedia Services, Journal of Telecommunication Systems, Springer, 2009.

[23] G. Zhang et Al, "Revealing the calling history on SIP VoIP systems by timing attacks.", In Proceedings of the 4th International Conference on Availability, Reliability and Security (ARES 2009). Fukuoka, Japan, 16-19 March 2009.