A new authentication method based on cryptographic identifiers CGAs

Nahla Abid, Maryline Laurent

▶ **To cite this version:**

Nahla Abid, Maryline Laurent. A new authentication method based on cryptographic identifiers CGAs. [Research Report] Dépt. Logiciels-Réseaux (Institut Mines-Télécom-Télécom SudParis); Services répartis, Architectures, MOdélisation, Validation, Administration des Réseaux (Institut Mines-Télécom-Télécom SudParis-CNRS). 2009, pp.29. hal-01360031

HAL Id: hal-01360031
https://hal.science/hal-01360031

Submitted on 5 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A new Authentication Method Based on Cryptographic Identifiers CGAs

LOgiciels-**R**éseaux	**Nahla ABID** **Maryline LAURENT**	09013 -LOR

A new Authentication Method Based on Cryptographic Identifiers CGAs

ABSTRACT

We define in the context of this work a new EAP method, that we call EAP-CGA. This method performs a mutual authentication of the hosts based on their CGAs. The verification of the authenticity of the provided CGAs decides the failure or the success of the authentication process. We provide also a new way on how to integrate this new authentication method in the cryptographic protocol of HIP, known as the HIP Base Exchange. Because HIP is a recent protocol proposal dedicated mainly to solve the mobility problem in IP networks, such an idea paves the way to authentication extensions related to this solution and has other benefits related to hosts identification, mainly coupling two identifiers of the same identity.

Our protocols proposals are validated using an automatic validation tool and the results we obtained show their robustness mainly against Man-in-the-Middle and replay attacks.

Nahla ABID Etudiant TELECOM & Management SudParis. Département LOR 9, rue Charles Fourier 91011 Evry cedex	Maryline LAURENT Professeur TELECOM & Management SudParis. Département LOR 9, rue Charles Fourier 91011Evry cedex E-mail: Maryline.Laurent@it-sudparis.eu

TABLE OF CONTENT

1 Introduction

Identifiers strongly evolved since the beginning of the Internet. Naming the communicating parties was one of the first issues and it evolved to more consistent services requiring new protocols and having needs for new identifiers.

This paper gives a short description of some cryptographic based identifiers that emerged in the recent past, along with their context, structure and usage. The properties of them are then underlined and compared.

The goal of this analysis is to find out the strength and weaknesses of those identifiers. We do not intend to bring practical solutions to the identification problem. Our objective is to underline several points we find out through analyzing the current identifier situation. These key points could serve as a basis for future works intending to define new identifier formats, possibly more efficient and consistent.

First of all, we need to clearly define the vocabulary – identification, authentication, and localization - used throughout the paper:

- Identification means assigning an identifier to the identity of the host. This identifier must be unique, thus, it can uniquely distinguish the host from others in the network. Two different identities should in no case have the same identifier.
- Authentication refers to the capability of an identifier to serve to authenticate its owner. The authentication here refers to the act of verifying that the presented identifier corresponds to the identity it is claiming to represent. One of the available authentication methods is based on proving the ownership of a secret, such as private keys in the case of the asymmetric cryptography.
- Localization serves to localize a host for traffic routing purpose. The location information can be carried into the identifier itself, for instance under the form of a topological location of the host.

Our study focuses on the three following types of identifiers: Cryptographically Generated Addresses (CGAs) [1], Hash Based Addresses (HBAs) [2], and Host Identity Tags (HITs) [3] [4]. These identifiers have in common to support several of the identity services (identification, authentication, localization) that could help improving the working of the current networks, and satisfying their increasing needs for mobility, security, and multi-homing. The analysis leads to wondering whether one of the identifiers is able to satisfy all the identified identity services.

Another interesting point is the combined usage of identifiers (HIT and CGA) to strengthen the security of the HIP application. Based on the authentication done by SEND

using CGA, we adapted this authentication protocol to define a new EAP method based on CGA, and a way to integrate it into HIP. A validation of EAP-CGA is performed.

This document is organised as follows. Section 2 proposes a survey of the main network identifiers. The context and the use of each identifier are described. Section 3 presents a more detailed study of the properties supported by those identifiers. Section 4 recapitulates the results and gives conclusions. In sections 5 and 6, we present respectively our proposed new EAP method based on CGAs and its integration into the Host Identity Protocol (HIP) [4]. The security analysis of the HIP vs HIP with EAP-CGA is discussed in Section 6.

2 Description of the Selected Identifiers

This section presents the identifiers CGA, HBA, and HIT focusing mainly on their structure. They are all based on asymmetric cryptography, that is, each host is assumed to have a pair of public/private keys, and an identifier generated from its public key. As such, proving the ownership of the private key serves to prove the identity of the host.

2.1 Cryptographically Generated Addresses (CGAs)

A CGA can be simply defined as an IPv6 address [5] closely bound to a public key. It is a networking-level identifier that was mainly introduced in the context of SEND (Secure Neighbor Discovery Protocol) [6] in order to secure the neighbor discovery [7] mechanism. CGA has the same length and structure than an IPv6 address with the slight difference that the last 64 interface identifier bits are calculated over the CGA Parameter Data Structure (CGA PDS) [1] thanks to hashing functions. The CGA PDS as depicted in Fig. 1 includes the public key of the host, the 64 bits subnet prefix and the SEC parameter. The SEC parameter is a value between 0 and 7, which allows the host to fix its CGA robustness against brute-force attacks.

The resulted 128 bits made of the 64 bit subnet prefix and the 64 bit interface identifier is the CGA address of a host. This CGA address helps the neighbors of the host to identify and authenticate the host. The neighbors only need to get its CGA PDS and to check the mapping between the CGA and the PDS. The SEND protocol defines the messages to communicate the PDS.

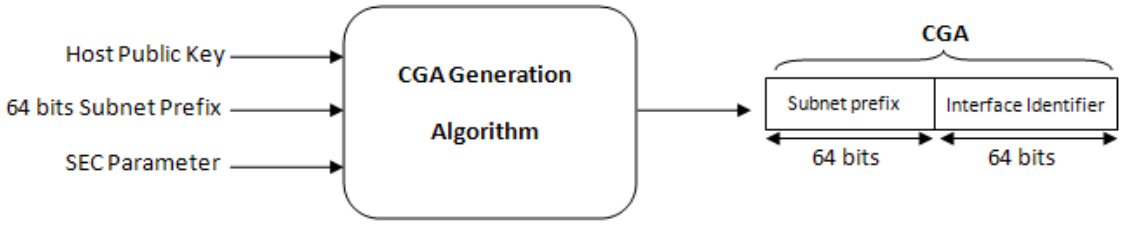

Fig. 1. The CGA Generation Algorithm

2.2 Hash Based Addresses (HBAs)

HBA is also a networking-level identifier having the form of an IPv6 address, and is an IETF draft proposal of 2007 [2]. The principle of HBAs is the same as CGAs: the interface identifier is cryptographically bound to the public key of the host. However, it is adapted to the multi-homing, that is, the possibility for a host to be attached simultaneously to different networks through different interfaces, and having several subnet prefixes.

A multi-homed host can generate a HBA with the same generation algorithm than the CGA. As depicted in Fig. 2, the only one difference is that the N subnet prefixes of the host are contributing to generate the 64 bit interface identifier.

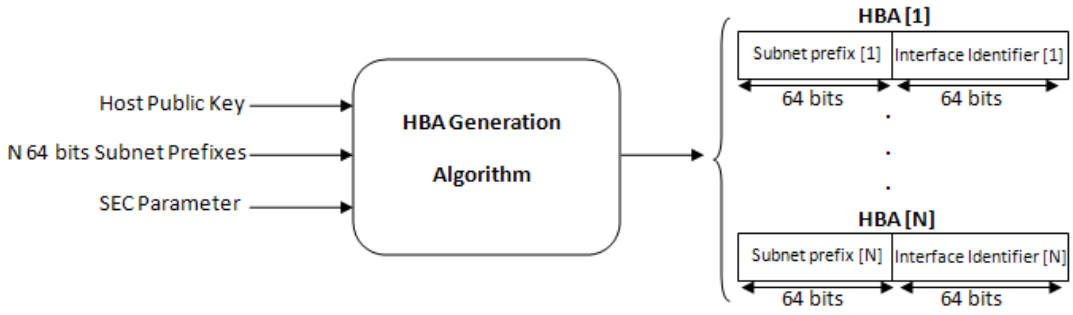

Fig. 2. The HBA Generation Algorithm

At this early level, it is worth noticing that the link between the HBA as an identifier and the identity of the host itself is consistent. Including the multi-homed information of the host in all its identifiers strengthens the representation of the identity by a HBA.

2.3 Host Identity Tags (HITs)

While both CGAs and HBAs were introduced in the context of IP address namespace, HITs belong to a new namespace - the Host Identity namespace [3] – which was proposed in the context of the Host Identity Protocol (HIP) [4]. From a HIP terminology point of view, the public key of the host is referred to as Host Identifier (HI) [3].

The generation of the HIT is explained in Fig. 3, and refers to the ORCHID identifiers [8] as the HIT belongs to the ORCHID family. A context ID value is conventionally fixed to 0xF0EF F02F BFF4 3D0F E793 0C3C 6E61 74EA and the concatenation of the context ID with the public key HI is then hashed with the sha-1 hashing function. Then an encoding function over the resulted hash leads to a 100-bit value chain, and the concatenation of the 28-long prefix value of 2001:10::/28 and the resulted 100 bits gives the 128-bit HIT.

A HIT can be summarized as a 128-bit-long hash over a public key. The link between the identifier and the identity of the host depends exclusively on the strength of the hash function.

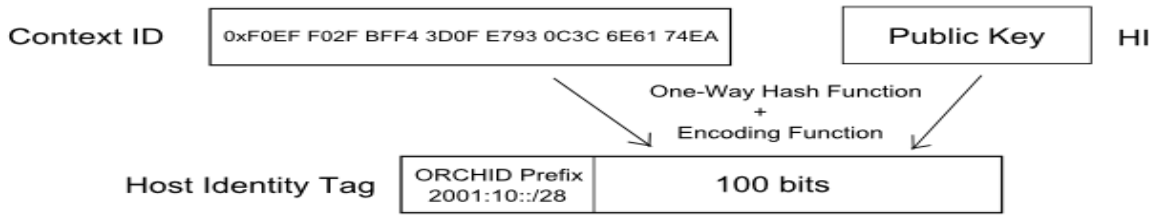

Fig. 3. Generating a HIT as an ORCHID Identifier

Works [3] considers extending the HITs that are today known to represent a single host to a representation of a group of hosts. The collective HIT is referred to as distributed HITs.

So far, the HIT has a flat structure, but in recent works [10], there was an attempt to define a hierarchical structure for HITs. This hierarchical HIT structure is depicted in Fig. 4. The first 32 bits represent the HIP management domain tag of the host, and the last 96 bits represent the host tag that corresponds to the hash over the public key. The hierarchical HIP helps strengthening the link between all the HITs generated by the same authority.

Fig. 4. Structure of a Hierarchical HIT [10]

3 Properties of the Identifiers

Our analysis focuses on the following properties: anonymity, authentication, identification, localization, security, and the requirement for an accompanied heavy structure in the networks.

3.1 Anonymity

The anonymity is the property that enables the hosts to reveal their identities to their communicating parties only. This is valid for the network environments where the level of trust is not that high and it aims to preserve the privacy of the users. For instance, the privacy can guarantee that the users are not traced during their moves.

Anonymity is becoming an ever increasing need and even sometimes, a legal obligation to the operators. Let us analyze now to what extent the identifiers CGA, HBA and HIT are able to satisfy the anonymity property.

HIP designers took into account this need for anonymity by distinguishing “public HITs” [3] and “anonymous HITs” [3]. Public HITs can be stored in public structures or directories, like DNS, LDAP… and are likely to be known by any entity asking for. Anonymous HITs are only known by their owners or their communicating parties. In most of the cases, they are generated by their owner. They are commonly used by HIP initiators, but can be also used by HIP responders. Referring to the protocol specification [4], it is recommended that a host has at least two HITs, one public and one for anonymous usage.

No anonymity is provided with CGAs and HBAs. They were originally designed to secure the neighbor discovery mechanism with SEND, that is giving the hosts the capability to advertise and prove their identity to their neighbors. As such, it makes no sense defining some anonymous CGAs or HBAs. It is contradictory to the original objective of these identifiers.

3.2 Authentication

A host can authenticate by presenting both its identifier and its credential. The verifier has then to check the correct mapping between the identifier and the credential. That is, the credential (usually under the form of a digital signature) is proving that the originator owns the private key bound to the claimed identifier.

CGAs, HBAs and HITs inherit from the identity based-cryptography, and so they all naturally support the authentication service. However, in this section, we propose arguing on the robustness of this authentication. Claiming the ownership of an identifier like CGA, HBA or HIT is not enough to perform authentication. Other operations, protocols and structures are necessary.

For instance, the HIT is a hash of an RSA/SHA-1 or DSA public key, and any malicious node can claim an identity by generating a public key and computing a hash over it. The protocols related to HIP help solving the problem of "HIT ownership". In HIP, the authentication is done in the first phase of the HIP communication, which is known as the Base Exchange (BE) [4]. BE is a cryptographic four-packet exchange that permits to create a HIP context between any two HIP communicating parties and to perform mutual authentication. Signing parts of the BE messages using the private key corresponding to the claimed HITs ensures the host authentication. We will give more details of the BE in Section 6.

The mechanism is pretty similar with CGAs. Any malicious host can claim possessing a fake CGA and thus a fake identity. To authenticate the node and thus verify its identity, there is a first need to check the public key matching between the PDS and the CGA and then the signatures appended to some messages and signed with the private key of the host.

Note that the authentication procedure based on these cryptographic identifiers may be strengthened from a security point of view by introducing some electronic certificates computed over the public keys of the identifiers. The certificate generated by some trusted Certification Authorities will help the communicating parties to more closely bind the identity to its public key, but at a higher cost due to the deployment of authorities, the configuration of new trust anchors on each host, and the maintenance of the certificates.

3.3 Identification vs. Localization: A Real Trade-off

Likely to the persons being identified by their names, the hosts can be identified by their network identifiers. Likely to person names, we expect the network host identifiers to remain the same, wherever the hosts are located. A HIT can be considered as a name, as it

is fixed to a host whatever its location. The host can be reached through this HIT at any time.

A CGA or HBA is an IPv6 address closely linked to the subnet prefix where the host is connected. The first 64 bits of the address are the subnet prefix itself and the second 64 bits are generated over one or multiple subnet prefix (es). As such, moving from one network to another for CGAs requires computing the address again. For HBAs, the recomputation is required only in case the new subnet prefix was not included into the previous HBA computation. This change of identifier weakens the link between the CGA identifier and the identity of the host. From this point of view, we claim that no real identification is provided with CGAs and HBAs.

We can conclude that there is a real trade-off between the localization and identification services in current network host identifiers. If the localization information is favored, then the identification information is disadvantaged and vice versa.

As far as we know, there is no ideal identifier supporting both services. Including the localization information in any identifier without decreasing its link to its fixed corresponding identity seems hard to achieve, but we expect future works to provide possible solutions to this problem.

3.4 Security

Securing hosts identifiers is an important issue to be taken into consideration when generating the identifiers. Also, adjusting the security level of the identifier depending on the trust nature of the environments is also very meaningful and useful.

Keeping the same level of security of any identifier is a negative point. For instance, using unsecure identifiers in untrustworthy environments makes the node vulnerable to many attacks, mostly the ones related to the identity spoofing. Also, making heavy computational efforts to generate a very secure identifier to be used in high-level security environments is not a good idea.

The CGA and HBA generation proposes to adapt the security level of the addresses defining the SEC security parameter (integer between 0 and 7). SEC value equal to 0 corresponds to the lowest security level, while the value 7 matches the highest one. The higher the SEC parameter value is, the more expensive the CGA and HBA generation procedure is and the harder brute-force attacks are.

HITs are all generated with the same security level with a simple hash computed over the public key of the host. As such, all the security issue rests on the strength of the hashing function. Referring to the protocol specification [4], SHA-1 is the official algorithm for HITs, but we have to be careful and adjust the robustness of the hashing

function to the computational strength of the attackers. We also expect some future works to focus on making the relation between the HIT and the public key stronger and stronger. We also think about the alternative of increasing the size of HITs, but this idea is bad because the HIT should remain the same size than the IPv6 address to make the use of HITs transparent to current protocols, APIs and applications. Therefore, a more suitable solution is to include further information about the host into the HIT generation procedure.

3.5 Deployment Requirements

The main advantage of using cryptographic-based identifiers is their self-certifying property that links the public/private key pair to the identifier itself. No Public Key Infrastructures and trusted third parties are required for this certification purpose.

Sometimes, the introduction of new identifiers requires some modifications related to the current networking architecture in order to support their deployment. New network entities and parties or new features added to the existing ones might be needed, mainly for mapping purposes.

CGAs and HBAs are easy to deploy as no modifications are required in the networking architecture. The neighbor discovery mechanisms can be secured with the SEND protocol without any external entities, except the neighboring nodes. This deployment simplicity is a strong positive point for these identifiers.

HITs are cumbersome to deploy and manage in the IP based networks. To make use of HITs possible, [12] introduces a new Resource Record (RR) to the current DNS (Domain Name Services) and a new networking entity, called Rendez-Vous Server (RDV) [4]. The new HIP Resource Record contains the information related to the HITs of the nodes. As such, a mapping between the DNS namespace and the Host Identity namespace can be easily done by any DNS resolver. Another mapping is also required between the HITs and the IP addresses where the hosts are localized. This is the reason why RDV servers were proposed. They are responsible for mapping the HITs to the current IP addresses of the hosts. They are keeping up-to-date localization information of the hosts.

3.6 Mobility and Multi-Homing

Hosts are becoming more and more mobile and multi-homed. Devices integrate more and more technological network interfaces (e.g. 3G, Wi-Fi, Bluetooth). The support of the mobility and multi-homing is one of the criteria for evaluating the identifiers.

HITs are identifiers working above the network layer, that is, the mobility and multi-homing issues that are networking-level are transparent for them. A HIP host is keeping its HIT fixed even when it moves from one network to another or when it is connected to different networks simultaneously. The mapping between the HIT and the IP addresses to which the host is reachable is updated by the host into its RDV servers thanks to some update messages. As such, HITs are offering the full support for both mobility and multi-homing.

HBAs were designed to support multi-homing as the multi-homed network prefixes contribute to their generation. However, HBAs do not support mobility. In case of moves, the new subnet prefix must be integrated into the HBAs (unless it was integrated before), and a new HBA must be computed.

CGAs are very poor in terms of mobility and multi-homing. If the node changes its network, its CGA has to be necessarily recalculated again using the new prefix.

4 Synthesis on the Identifiers

We summarize the results of our analysis in Fig. 5. As a first conclusion, it is clear that HITs seem to be the identifiers offering the highest number of identity services. However, they suffer from the deployment issue in the networks. There is a need to introduce some cumbersome structures and this is costly in terms of administration, deployment and maintenance.

CGAs and HBAs are good identifiers from a localization point of view, but they do not support mobility. A host moving from one network to another is needed to recalculate its identifier.

Referring to the study we did, the following points were found out. There is a need to study further the HITs generation method in order to ensure their uniqueness and to specify in a more consistent way the Host Identity namespace. The proposition of hierarchical HITs can be a solution but this will lead to define another hierarchical namespace, along with the DNS one. Mature experiences with this latter namespace would be useful to help deploying the new proposal of hierarchical HITs in a secure manner.

The length of a HIT is limited to 128 bits. All the strength of this identifier rests on the hashing function used. That is why, it is highly recommended to strengthen the security level of HITs by adjusting periodically the hashing function robustness to the increasing computation capabilities of the attackers.

Also, it should be noted that the direct use of the public key to generate the identifiers CGAs, HBAs and HITs can raise some problems from the identity point of view. Usually, the validity duration of a public key is limited for security reasons, but there is no such

validity limitation for the cryptographic based identifiers. That is, in case of a compromised private key, the bounded identifiers can still be used, and spoofing attacks are possible. If an identity is changing its public key, it would be better that its old HIT is not reused by another entity, but we have no guarantee about this, except with hierarchical HITs. Each time a public key is changed, the identifier supposed to represent the identity is also changing.

For HITs that belong to the same host, there is no possibility so far to define a logical link in between. We strongly believe that identifiers of the same identity should have a clear link relation because finally they represent the same entity.

For HBAs, we outlined that no recent works have been presented. In our own point of view, it could be interesting having a stable HBA taking into account all the subnet prefixes that are visited by the device, and which does not require recomputation in case of moves. However, we must admit that it is hardly feasible to know in advance such subnet prefixes.

We think that a challenging issue is to define another format of identifiers able to support multi-homing but in more elegant and reasonable way than HBAs. Beyond this, it would be ideal defining an identifier gathering all the strong points of CGAs, HBAs, and HITs in one consistent format.

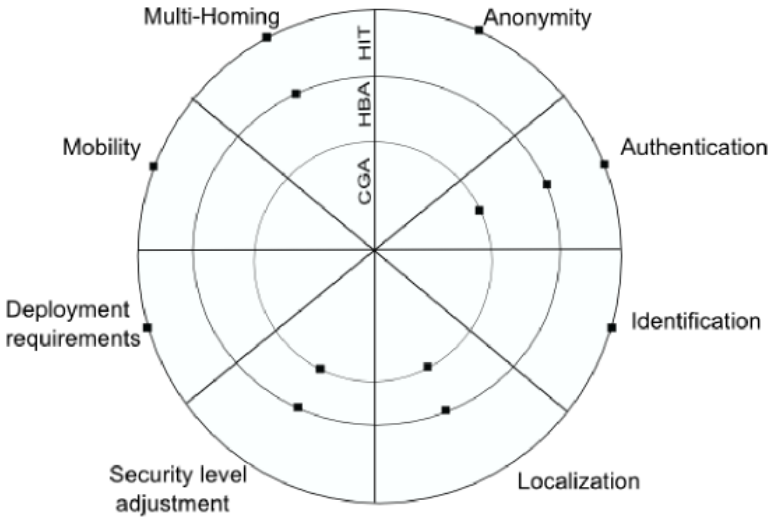

Fig. 5. Analysis Summary

5 Defining a new EAP Method: EAP-CGA

5.1 EAP Protocol Overview

The Extensible Authentication Protocol (EAP) is a generic authentication framework, standardized by the Internet Engineering Task Force (IETF) that provides an infrastructure for network access clients and authentication servers. It is described in [16].
EAP by itself does not perform the act of authentication; it merely provides a mean for the negotiation between the user and the authentication server and it is augmented with an authentication method that has its own requirements and procedures. EAP has gained popularity in the recent years due to the flexibility it provides. In fact, it does not specify any specific authentication mechanism. Instead, it is able to support any existing legacy authentication mechanisms as well as newer and stronger ones able to appear.
EAP permits the negotiation of the parameters needed to process the authentication of the user. So it makes sense that EAP includes start and end messages and a set of middle messages. There are four types of messages in EAP:

- EAP request message (authenticator to user)
- EAP response message (user to authenticator)
- EAP success
- EAP failure

Table 1 shows the format of EAP packets.

Field name	Sub-field	Description
EAP header	EAP code	1 for EAP request,2 for EAP response, 3 for EAP success,4 for EAP failure
	EAP ID	Transaction ID: used for matching request and response messages(1 for first message)
	EAP message length	2 octets for the length of the message
EAP data	Type	Type field is only used for EAP request and response message and indicates the type of information being requested from the peer corresponded to the server.Examples: 1 for Identity
	Type data	

Table 1. EAP Packet Format

Request and response messages are used for all information exchanges and by using an EAP type field. Further EAP request and response messages can be created for exchanging a variety of information types.
For instance, EAP request/identity is a message sent by the server, requesting the user to present his identity, for example, a name or any other type of identity. However most of type numbers are used to identify authentication methods used, such as EAP-MD5, EAP-TLS [17]. . .

The flexibility of using type field for creating a variety of exchanges is the reason for
The so-called "extensible": indeed, new authentication methods can be introduced easily.
EAP success and failure messages serve to indicate whether the authentication process is completed.
Table 2 summarizes the different EAP message types.

Name	Description
EAP Request	Encapsulates a network access control request
EAP Response	Encapsulates a reply to a network access response
EAP Success	Sent by the authenticator to the peer to indicate successful authentication
EAP Failure	Sent by the authenticator to the peer to indicate authentication failed

Table 2. EAP Messages Types

Concerning the communication model in the presence of EAP, it is almost a three-party based model that comprises the following elements like in Figure 6.

Fig. 6. EAP Three-Party Communication Model

• **Peer:** represents the host attempting to connect to the network through the edge device.
• **Authenticator:** is in charge of access control on behalf of the network.
• **EAP Server:** terminates the EAP authentication method and other EAP-related functions.
Note that, although we mostly deal with three-party EAP models, a two party-EAP model can be used for some contexts. At that time, the EAP ends at the authenticator, and hence the authenticator is co-located with the EAP server.

5.2 Motivations

CGAs are classically used for a purpose of authentication in the context of SEND. The EAP framework [16] is designed to support many different authentication mechanisms and allows the peers to negotiate their authentication mechanism of choice rather than to be restricted to specific authentication mechanisms.

This is why the idea of proposing a new EAP authentication method based on the use of CGAs is pretty interesting. We propose a new EAP method called EAP-CGA. It is based on exchanging the CGAs of the peers in order to support the mutual authentication between peers. The verification of the CGAs and the PDS at the two parties' results in the success or failure of the authentication.
However, it is interesting to clear the following architectural point. EAP can run directly at link layer with no requirement for a network layer protocol, such as IP. This is referred to

as EAP over LAN (EAPOL) [16]. In this case, the host uses EAP to be authenticated and to be next able to have an IP address and access the network.
However, in our case, EAP-CGA runs over the IP protocol. Obviously, the host is authenticated using its CGA, i.e. its IP address. The EAP messages are carried into the IP packets. This situation is mainly faced in cases where the host is multi-homed, for instance, when the host needs to be connected to a second network, the authentication process can be processed using this method.

5.3 EAP-CGA Method Description

Our protocol overview here is based on a two-party model, in which the two peers deal with each other in order to perform a mutual authentication based on their CGAs. Thus, every peer acts at the same time as a client and an authenticator.
The exchange of messages for the EAP-CGA procedure is shown in Figure 7 and performed as described in the following.
As known in EAP, the messages from the initiating party to the responding one are carried into EAP-request messages, while in the reverse case, they are carried into the EAP-response messages.

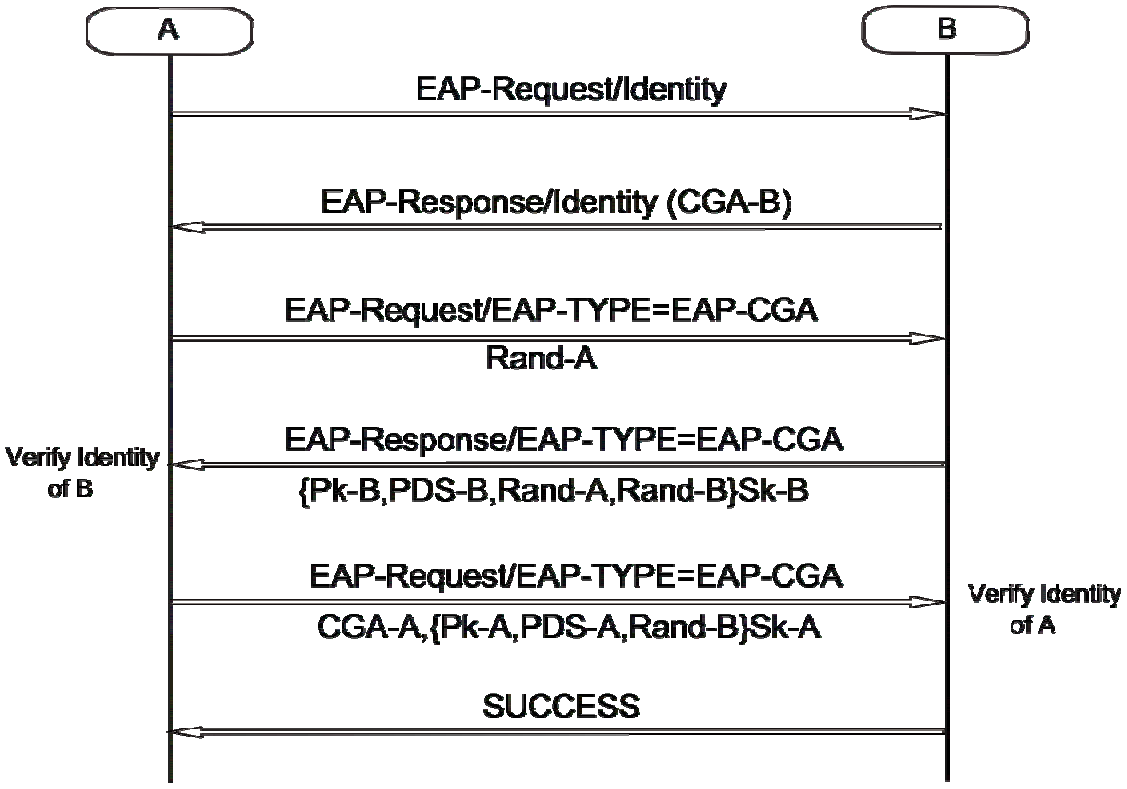

Fig. 7. EAP-CGA Method Messages

- The first exchange provides the first party (party A) with information on the other party identity. This exchange starts with A sending an EAP-request/identity packet to B which in turn responds with an EAP-response identity packet that includes the identity of B, i.e. its CGA in the studied case.

- Once the identity of B is provided to A, the EAP signaling starts with A sending an EAP-request packet with an EAP-TYPE parameter requesting that EAP-CGA is used as the authentication method and informing that an EAP-data field carries a Rand-A nonce, generated by A.

- B responds with an EAP-response message with the same EAP-TYPE field, i.e., EAP-CGA, that encapsulates a signed data field containing its public key, its PDS, echoed Rand-A and another self-generated nonce Rand-B. The signature is issued using the private key of B.

- Upon receiving this message, A decrypts the signed field using the public key of B, it verifies that it has sent the corresponding EAP request packet using the nonce Rand-B, and then it verifies the identity of B by verifying the matching between the sent CGA and PDS as described in [1]. If the verification succeeds, the next step is for A sending an EAP-request message containing its CGA and a signed field including its public key, PDS and the nonce Rand-B. The private key of A is used to sign this field. In the case where the verification fails, an EAP-failure message is sent from A to B to indicate that the authentication failed and to stop the method processing.

- Then B decrypts the signed field, it verifies the identity of A by verifying the CGA and the PDS of A. If the authentication is successfully completed, an EAP-success message is sent from B to A, unless it is an EAP-failure message that indicates the reverse.

6 EAP-CGA Integration into HIP BE

6.1 HIP Communications

Obviously, the new protocol HIP [3][4] specifies a different way for establishing HIP based communications and proposes for this a new packet type, the HIP packet one [4]. If two hosts want to communicate using HIP, two phases are needed: the HIP Base Exchange [4] and the secured data transfer [3].
The BE establishes a security association between two hosts [4]. It consists in a four-way handshake based on the exchange of HIP packets between the host which triggers the communication, referred to as the initiator, and its peer host designated as the responder.
After the BE is finished and the HIP association is established, the two hosts are ready to start the second phase of data exchange in a secure manner.

The main purpose of the BE is to establish what we call a HIP association [4] between the initiator and the responder. It supports mutual authentication based on public/private keys of hosts, symmetric D-H key agreement and DoS (Denial of Service) prevention mechanism. The different messages exchanged during the BE are shown in Figure 8.

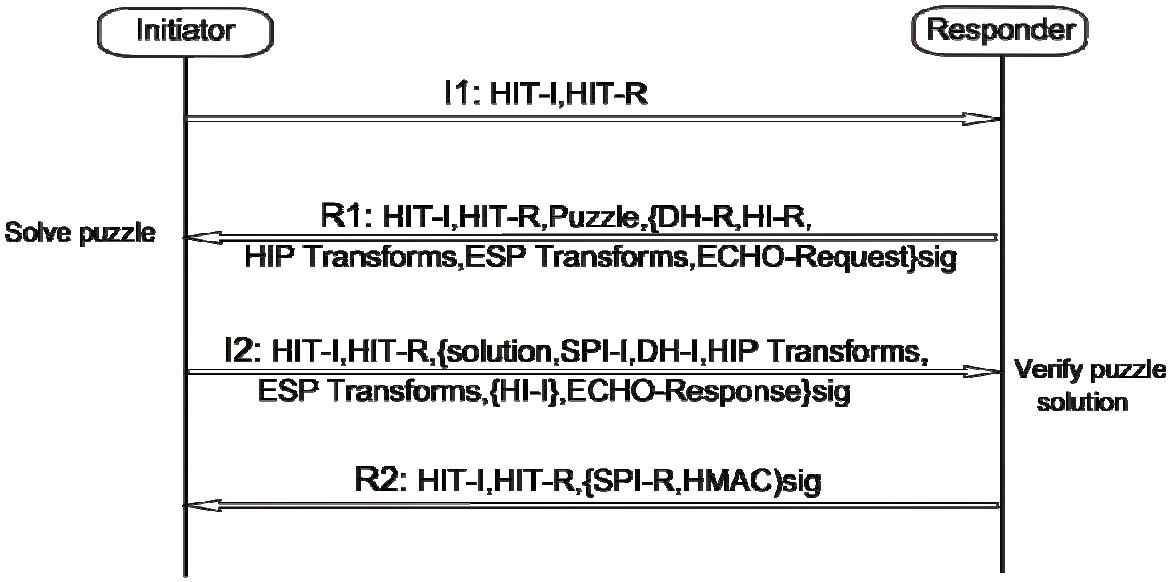

Fig. 8. HIP Base Exchange

The Initiator starts by sending an empty message I1 to the Responder, containing only the initiator and responder HITs (HIT-I and HIT-R).
Even before the Responder receives the I1 message, it precomputes a partial R1 message. The pre-computed R1 includes the HIT-R, the Responder's Diffie-Hellman key, the Responder's host identity HI-R (i.e. a public key), the proposed cryptographic algorithms for the rest of the Base Exchange (HIP transforms), the proposed IPsec algorithms (ESP transforms), and an ECHO Request field. The ECHO Request contains data that the Initiator returns unmodified in the following message I2. This precomputed message is signed by the responder.

After receiving an I1, the responder adds a puzzle and the HIT-R to the message and sent it. The Puzzle parameter in R1 contains a cryptographic puzzle [3][4], which the Initiator is required to solve before sending the following packet I2. This is to prevent against DoS attacks. When receiving R1, the initiator checks the signature, solves the puzzle and starts creating the I2 packet. This latter contains the initiator's Diffie-Hellman key, the HIP and ESP transforms proposed by the initiator, the puzzle and its solution, the Initiator public key (HI-I) encrypted using the new session key, and the Echo Response. All the I2 is signed.

The responder then verifies the puzzle solution, it computes the session keys, it decrypts HI-I, and it verifies the signature on I2. The Responder then sends R2, which contains the SPI for the Initiator-to-Responder IPsec SA, an HMAC computed using the session key, and a signature. For the Initiator, the exchange is concluded by the receipt of R2 and the verification of the HMAC and the signature.

6.2 Motivations

HIP is one of the most prominent solutions for providing secure mobility in future IP networks. Integrating existing authentication mechanisms in HIP communications is a necessity for the deployment of this protocol in current architectures.
Combining EAP authentication mechanisms with HIP is an interesting proposal and can add many useful extensions to HIP like access control mechanisms and interactions with AAA architectures.
In fact, as described in chapter one, HIP BE permits only a mutual authentication based on the hosts' HITs, it means that no real user authentication is provided since many users on the same host are presented with the same HIT. Adding legacy user authentication mechanisms to HIP like EAP can solve this problem.
On the other hand, EAP is most of the time used in current architectures in interaction with AAA architectures. Using EAP with HIP may prepare for future HIP interactions with such architectures.
In our work, we propose to integrate the needed negotiation messages for our method EAP-CGA in the HIP BE. The first advantage of such proposal is the HIP interaction with EAP, and AAA architectures. The second one is the strengthened identification. With the original HIP BE, the identity of the host is represented under the form of HIT. Adding EAP-CGA will strengthen the link between the identifiers of the same host: HIT and CGA. A host can easily check that both HIT and CGA were generated with the same public key.

6.3 Integration of EAP and EAP-CGA into HIP

[16] is a recent IETF draft that proposes a solution to integrate EAP data into HIP messages defining two EAP parameters: EAP-SIGNED and EAP-UNSIGNED. These parameters can be used in R1, I2, R2 and UPDATE control packets. Sometimes the EAP negotiation is longer than the HIP BE, thus it can be continued running just after BE with UPDATE packets.
As described in figure 9, the EAP-CGA messages can be piggybacked into the BE and UPDATE packets. First, the responder announces in R1 that there is a service requiring authentication using the SERVICE OFFER parameter. The initiator transmits the agreement using the SERVICE ACK parameter. The EAP-CGA negotiation messages then starts and completes piggybacked into the BE exchanges and later the UPDATE messages.

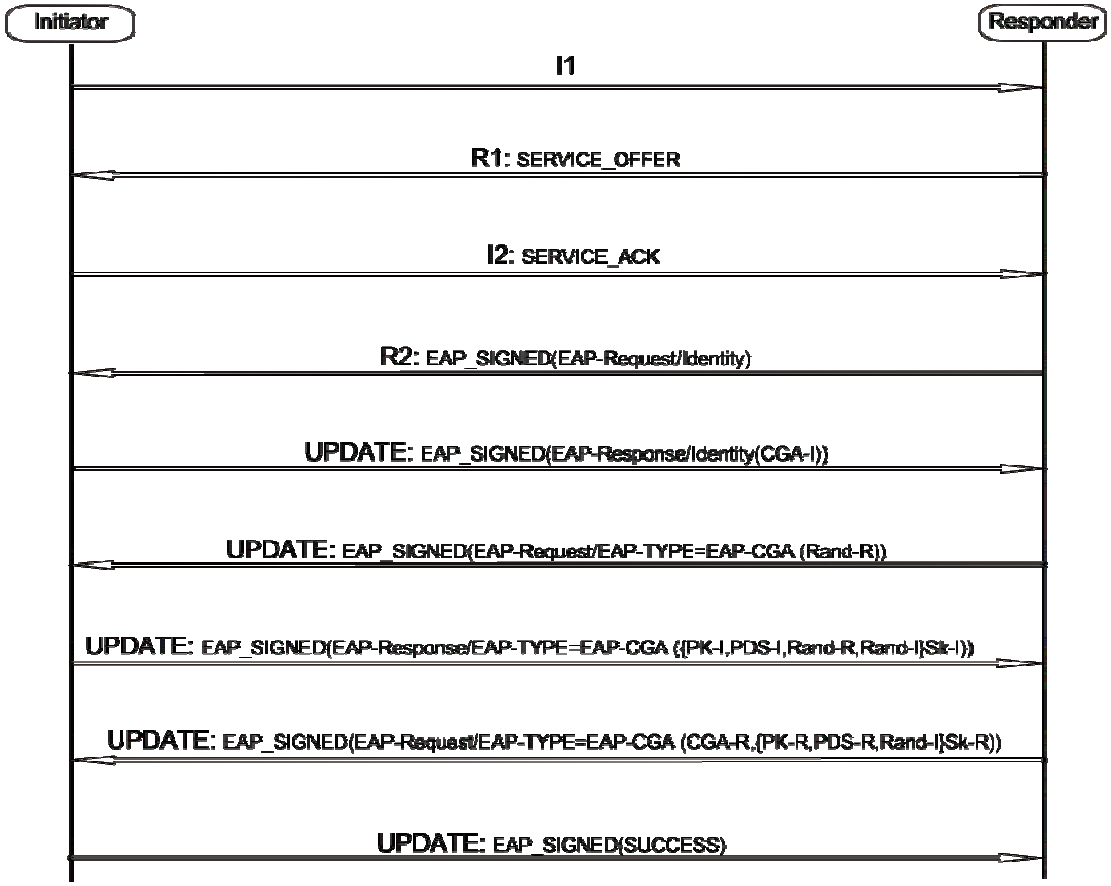

Fig. 9. Integrating EAP-CGA in HIP BE

7 Security Analysis

We propose to use some validation tools to verify the security properties of our resulting protocols, along with a security analysis. First we present the validation tool in use (AVISPA) followed by a security analysis of the EAP-CGA method and HIP BE with EAP-CGA.

7.1 AVISPA Tool

The validation tool we selected is AVISPA [14]. It is a push-button tool for building and analyzing security protocols. It provides a role-based expressive formal language for protocol specification and it integrates four different back-ends that perform the actual analysis of the protocol. As displayed in Figure 10, the AVISPA tool enjoys a graphical user interface that facilitates the editing of the protocol specifications and the selection of the back-end in use.

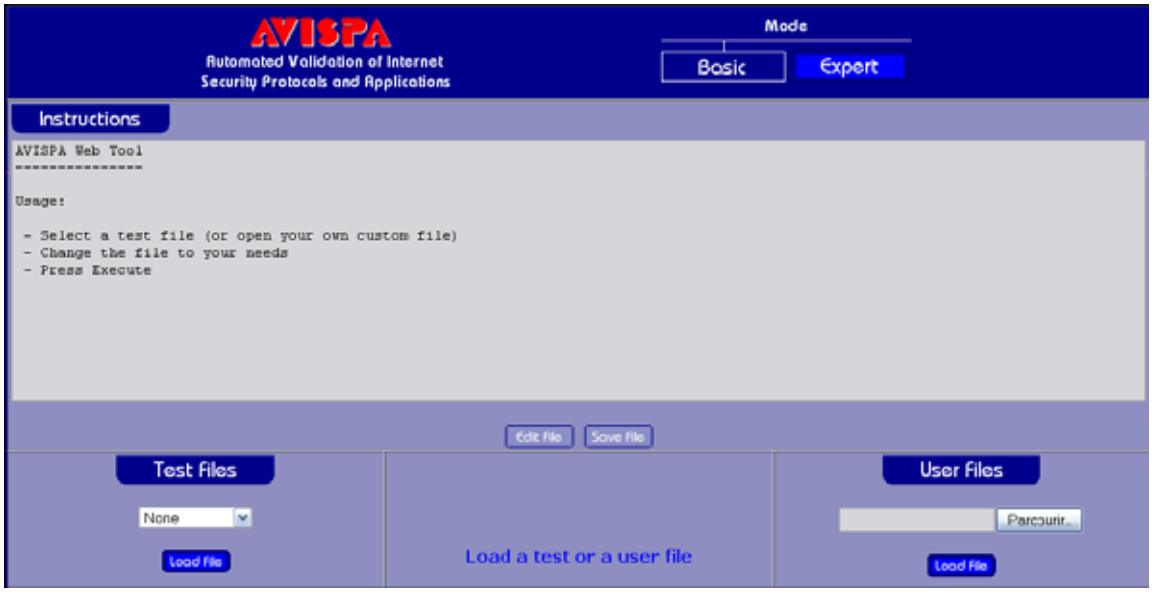

Fig. 10. AVISPA Web Tool

AVISPA Web Tool

The architecture of the tool is presented in Figure 11. The user interacts with the tool by specifying the protocol he is willing to check and the security properties he wishes to verify in the AVISPA language. The High level Protocol Specification Language HLPSL) [14] is used to specify the protocol to the tool.

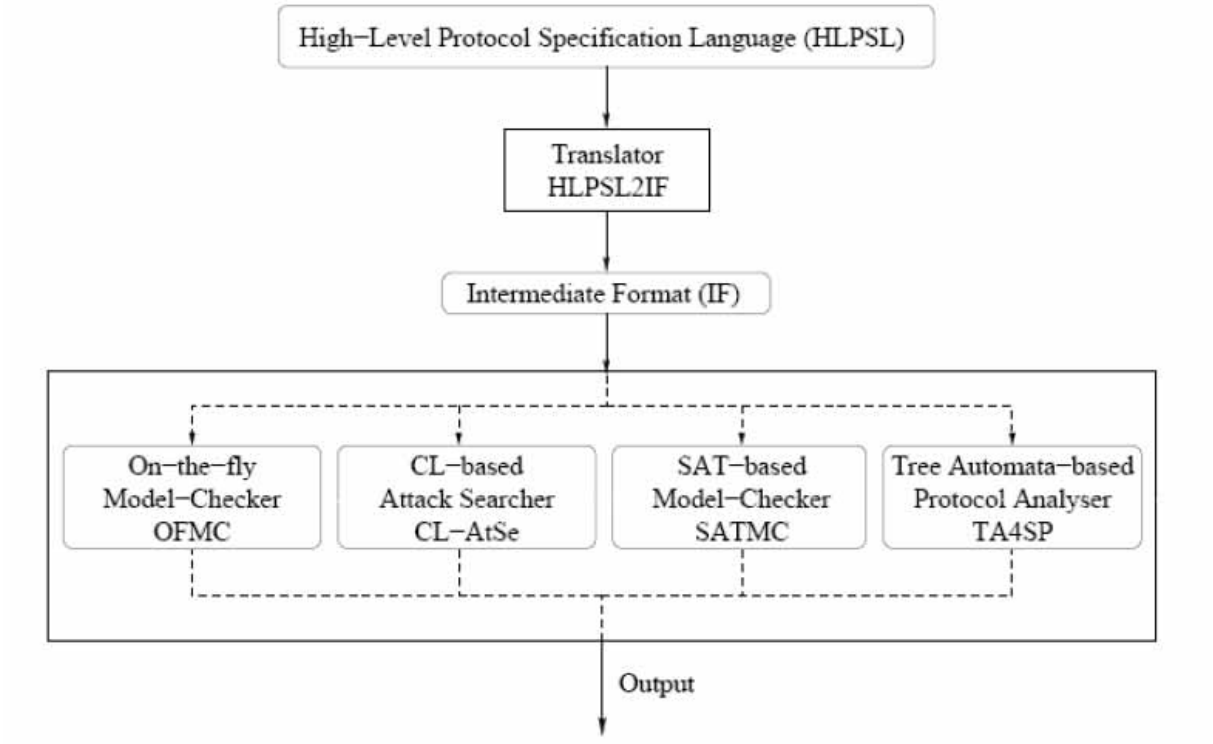

Fig. 11. AVISPA Architecture

This security problem is processed by the tool as follows: The HLPSL specification is translated into Intermediate Format (IF) [14] specification, a lower level language. This translation is performed by a translator called HLPSL2IF [14].
This step is completely transparent to the user. IF specifications are inputs for the different back-ends of the tool, which implement different analysis techniques. In the current version of AVISPA, four back-ends are in use:

• On-the-fly Model-Checker (OFMC) [23]
• Constraint-Logic-based Attack Searcher (CL-AtSe) [24]
• SAT-based Model-Checker (SATMC) [25]
• Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP) [26]

HLPSL

As mentioned above, HLPSL is a role-based language. In fact, each participant in the protocol is presented in a separate role called *"Basic Role"*. In this module, we specify which parameters the participant initially knows and its initial state. It includes also a section called *"transitions"* where we specify the received and sent messages and how they can interfere with the role's state.
In addition to basic roles, there are also composition roles, called *"session"*, that HLPSL defines. They are used to combine different basic roles and to execute them in parallel.
They initiate one instance of each basic role and thus, one run of the concerned protocol.
There is also a top level role, called *"environment role"*, where we can define what information an intruder can have access to and how and when he can interfere with the protocol.
Till now, we presented only the different semantics to use in order to describe the protocol to be analyzed but the security properties need to be verified. They are defined as goal facts in the transition section of each role and in a separate section, called "goals". In this section, we specify which combinations of those goal facts can be considered as an attack.

7.2 EAP-CGA Security Analysis

EAP-CGA specification in HLPSL contains two roles, Alice and Bob that present the two peers trying to authenticate each other using this method. Figure 12 presents an extract of this specification.

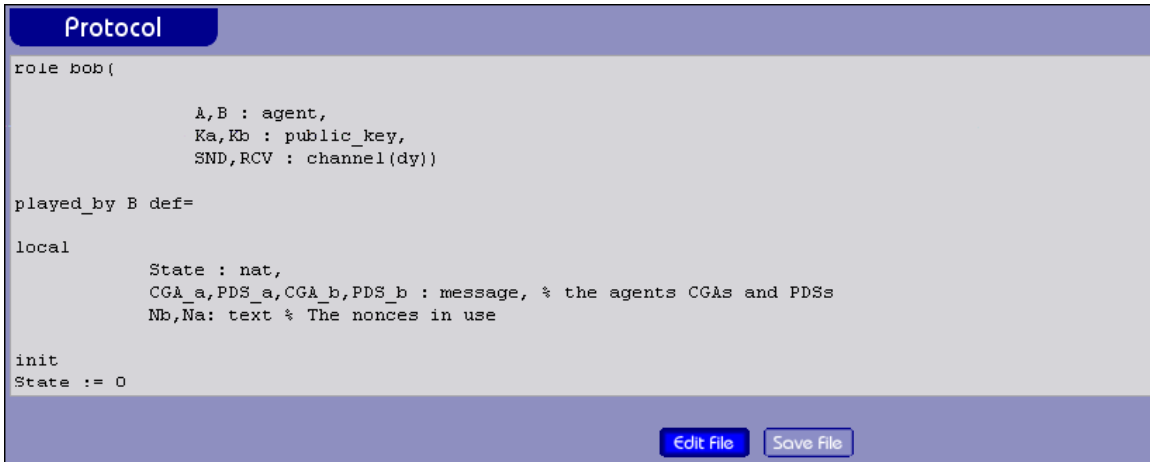

Fig. 12. EAP-CGA Specification in HLPSL

The analysis result is shown in Figure 13.

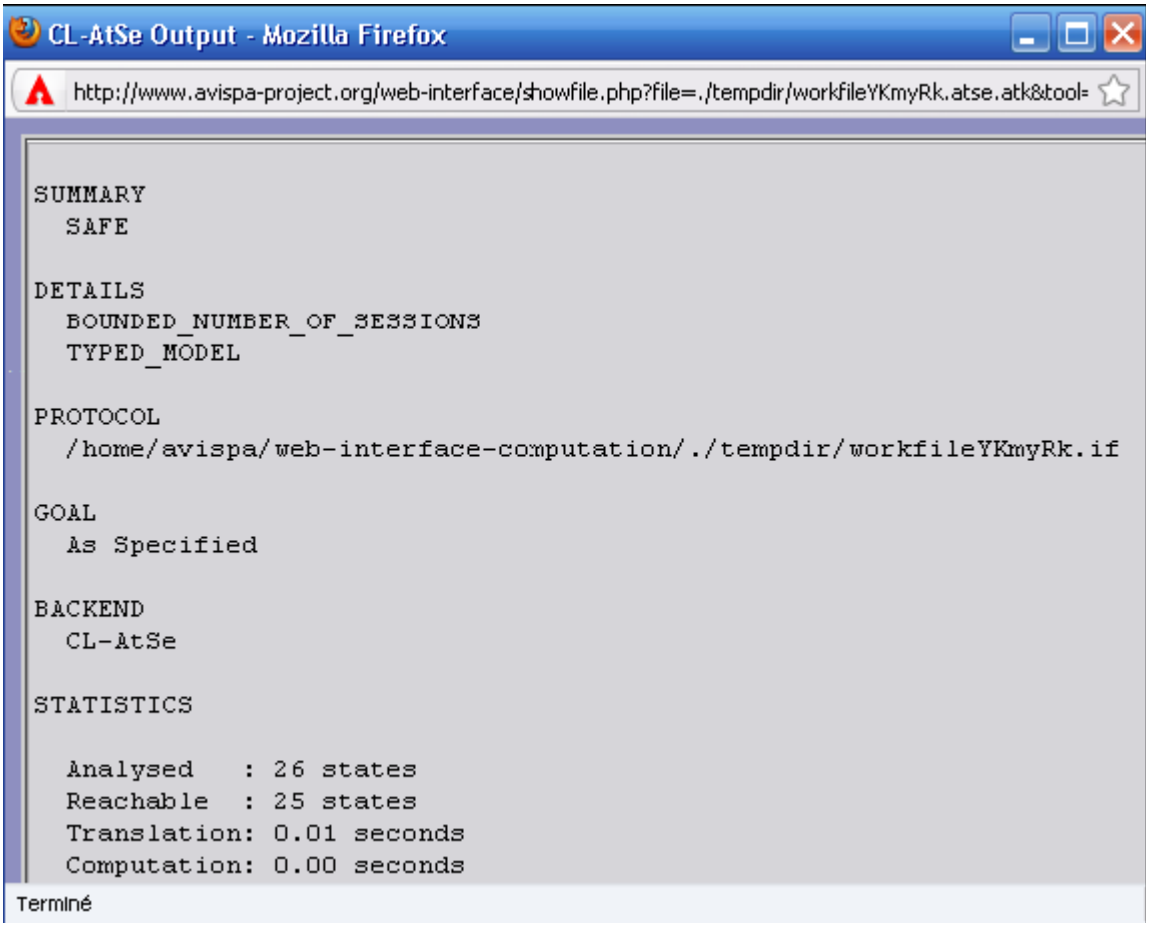

Fig. 13. EAP-CGA Analysis Result

We succeeded to verify the following security properties:

• **Mutual authentication:** Both peers can authenticate each other using their private keys. In fact, in EAP-CGA, every party of the communication sends at a certain time a message encrypted with its private key. Thus, the corresponding party can verify the authenticity of the message by doing decryption using the corresponding public key.

• **Protection against replay attacks:** In order to verify whether the protocol resists to replay attacks, we can declare two parallel sessions at the top-level composed role. In that case AVISPA tries to re-send old messages used in another session to test whether this will be detected.

In our method, the result of the validation shows that EAP-CGA is robust against such type of attacks. The freshness of the sent messages is supported by the two nonces: Na and Nb.

• **Integrity protection:** It provides data authentication and protection against unauthorized modifications. In our case, we are mainly interested in testing the integrity of the parameters used to make the verification of the identity of the peer, i.e., the PDS. Communicating the PDS in a signed field permits to support integrity.
However, the use of the asymmetric cryptography in our method may cause heavy computational consumption at the two parties. This is why EAP-CGA may not be suitable for authentication in environments with low-level consumption.
There is also the problem of the identity protection. In fact, the CGA of the peers are communicated in clear in our method messaging. This can be a serious security problem in contexts where the secrecy of users' identity is of relevant importance.

7.3 Security Analysis of BE with EAP-CGA

The objective of this section is first to check whether the introduction of the new EAP parameters into BE weakens the security of the BE protocol and second to detect possible new security threats related to that.
For that purpose, two steps are performed, validating HIP BE with and without EAP-CGA and then comparing the results.

Validation of HIP BE
HIP specification in HLPSL is based on two basic roles: Alice and Bob representing the initiator and the responder. Figure 14 shows an extract of the HLPSL specification of the HIP BE protocol.

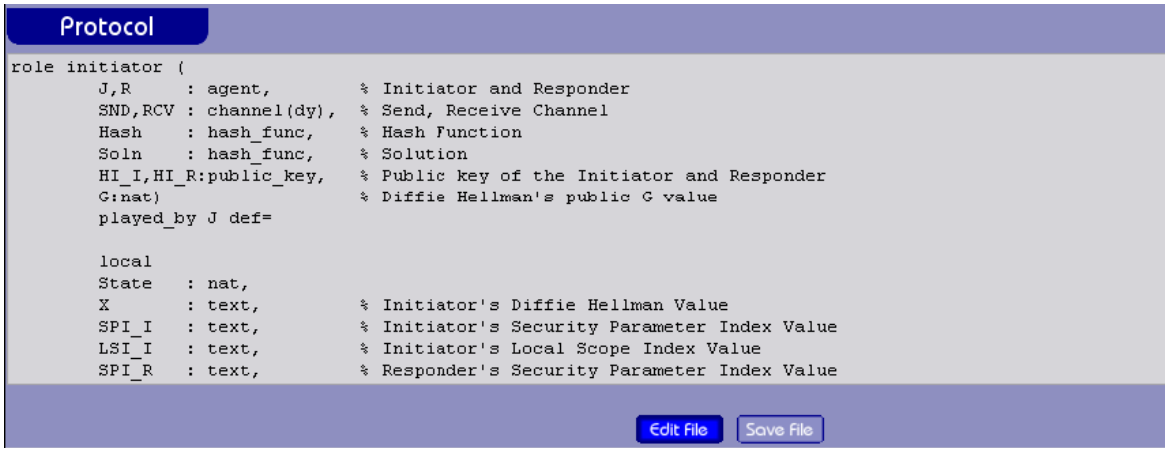

Fig. 14. HIP BE Specification in HLPSL

The security properties we mainly tested are Man-in-the-Middle (MitM) and Replay attacks.
After running AVISPA, we got the results depicted in Figure 15. That is, no security threats were found by AVISPA. That means that the protocol is resistant to MitM attacks. A MiTM attack means that the intruder is attempting to make the two communicating parties believe that they are talking directly to each other over a private connection while the entire communication is under control of the intruder. However, this is obviously not possible between the initiator and the responder.

Fig. 15. HIP BE Analysis result

From the initiator's point of view, the identity of the responder is got from a secure DNS as described in [27] and thus, it can validate the packets coming from the responder using this information.
For the responder, it can verify the HI of the initiator and its level of trust after receiving the I2 packet using any trusted way. However, the HIP BE opportunistic mode where the initiator chooses to use anonymous HITs increases the risk of a MitM attack. This is why it is almost the time not preferred to accept such type of communications.
AVISPA did not detect any replay attack threats on the protocol.

Validation of HIP BE with EAP-CGA
We focus on the integration of EAP-CGA into HIP security issues.

The protocol specifications contain obviously the same basic roles. The only slight difference is that we include in the definition of each role the new EAP parameters. Figure 16 gives an extract of the HLPSL specifications.

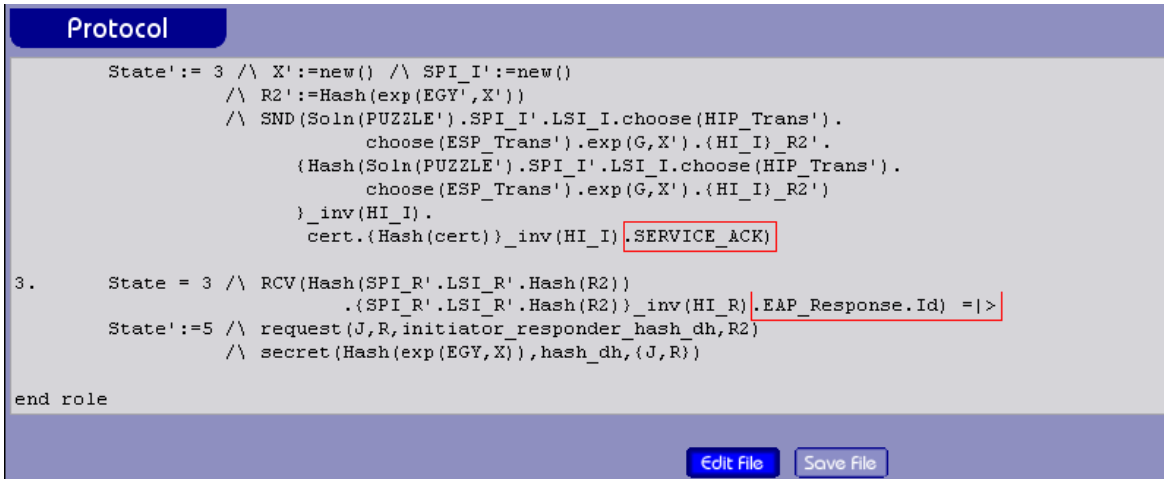

Fig. 16. EAP-CGA Integration into HIP BE Analysis Result

After running this specification with AVISPA, we got the result that no new security threats were detected due to integration of our method into the BE.

Analysis of HIP BE with / without EAP-CGA

The results were interesting as we succeeded integrating EAP-CGA into HIP BE to strengthen the authentication process between the initiator and the responder.
However, note that if new threats were found, this could have been easily solved by encapsulating the new EAP parameters into the ENCRYPTED parameter defined in HIP specifications [4].

Another point is related to an important security property out of scope of the AVISPA tool, which is the location privacy. This property is not supported by the new BE with EAP-CGA method.
Every message is carrying the HIT of the host coupled with its CGA in cleartext format. That means that any one listening to the communication can easily make the relationship between the identity of the host, i.e., its HIT, and its location, i.e., its CGA. In that case, any topological location change of the host can be traced.
This is why, when using our method, this point must be taken into consideration especially in environments where the location privacy is a real security challenge.
To solve this problem, the use of "blind HITs" as defined in [29] can hide this relation and thus ensure the location privacy.

8 Conclusions

In the first part of this report, we presented a study and a comparative analysis of three promising cryptographic identifiers: HITs, CGAs and HBAs. We provided also an overview of the EAP framework as a prominent authentication solution alongside with the new mobility protocol HIP.
We provided later the basic lines of the EAP-CGA method as a new EAP authentication method that consists mainly on providing mutual authentication of the peers using their CGAs verification.
We provided also a new extension to the HIP solution, in which we tried to integrate the EAP-CGA authentication mechanism with the cryptographic protocol that HIP proposes, the BE. This extension paves the way for integrating authentication mechanisms in HIP and it provides a new way to link the different identifiers of the same host.
The last part of the work consisted on validating those new protocols from a security point of view using the AVISPA validation tool. This step was a good opportunity to enhance our knowledge about protocol specification and validation and it allowed us to learn a new specification language, HLPSL. The detailed study that we performed in the first part of this work about HIP BE led us to bring some interesting remarks about the security of this protocol.
Concerning the results of this work and referring to the validation results we obtained, we assert that the new protocols meet the security purposes we had in mind when designing them, mainly robustness against MitM and replay attacks.
Those results seem to be encouraging and we sincerely think that an experimental experience with those protocols would bring important conclusions about their performance in real platforms. This can be the purpose of future work and extensions.

References

[1] T. Aura, Cryptographically Generated Addresses (CGA), RFC 3972 (2005)
[2] M. Bagnulo, Hash Based Addresses (HBA), draft-ietf-shim6-hba-05 (2007)
[3] R. Moskowitz, P. Nikander, Host Identity Protocol (HIP) Architecture,RFC 4423 (2006)
[4] R. Moskowitz, P. Nikander, P. Jokela, Ed., T. Henderson, Host Identity Protocol, RFC 5201 (2008)
[5] S. Deering, R. Hinden, Internet Protocol, Version 6 (IPv6) Specification,RFC 2460 (1998)
[6] J. Arkko, Ed., J. Kempf, B. Zill, P. Nikander, SEcure Neighbor Discovery (SEND), RFC 3971 (2005)
[7] T. Narten, E. Nordmark, W. Simpson, Neighbor Discovery for IP Version 6 (IPv6),RFC 2461(1998)
[8] P. Nikander, J. Laganier, F. Dupont, An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers (ORCHID) (2007)
[9] K. MAEKAWA, Y. Okabe, A Location Privacy Protection Framework with Mobility using Host Identity Protocol, Master Thesis (2009)
[10] S. Jiang, Hierarchical Host Identity Tag Architecture, draft-jiang-hiprg-hhit-arch-01.txt (2008)
[11] P. Nikander, "Denial-of-Service, Address Ownership, and Early Authentication in the IPv6 World," presented at Cambridge Security Protocols Workshop 2001, April 25-27, Cambridge University (2001).
[12] P. Nikander, J. Laganier, Host Identity Protocol (HIP) Domain Name System (DNS) Extension, RFC 5205 (2008)

[13] N.Abid, M.Laurent-Maknavicius, and H.Chaouchi. Experimental experience with host identity protocol (hip). Technical report, Telecom & Management SudParis LOR Department, 2008.

[14] www.avispa project.org. Hlpsl tutorial.
[15] M.Nakhjiri and M.Nakhjiri. AAA and Network Security for Mobile Access: radius,diameter,EAP,PKI and IP mobility. John Wiley & Sons,Ltd, 2005.
[16] B.Aboba, L.Blunk, J.Vollbrecht, J.Carlson, and Ed. H.Levkowetz. Extensible authentication protocol (eap). RFC 3748, June 2004.
[17] D. Simon, B. Aboba, and R. Hurst. The eap-tls authentication protocol. RFC 5216,March 2008.
[18] P. Jokela, R. Moskowitz, and P. Nikander. Using the encapsulating security payload(esp) transport format with the host identity protocol (hip)communication. RFC5202, 2008.
[19] P. Nikander, T. Henderson, Ed., C. Vogt, and J. Arkko. End-host mobility and multihoming with the host identity protocol. RFC 5206, 2008.
[20] J. Arkko, Ed., J. Kempf, B. Zill, and P. Nikander. Secure neighbor discovery (send).RFC 3971, March 2005.
[21] S. Jiang. Hierarchical host identity tag architecture. draft-jiang-hiprg-hhit-arch-01.txt, 2008.

[22] Varjonen. Hip and user authentication. draft-varjonen-hip-eap-00, July 2009.

[23] D.Basin and L.Vigan. Ofmc: A symbolic model checker for security protocols. Nternational Journal of Information Security, 2004.
[24] Y. Chevalier and L. Vigneron. Automated unbounded verification of security protocols.
[25] A. Armando and L. Compagna. Satmc: a sat-based model checker for security protocols.
[26] Y. Boichut, P.-C. Heam, O. Kouchnarenko, and F. Oehl. Improvements on the genet and klay technique to automatically verify security protocols.
[27] P. Nikander and J. Laganier. Host identity protocol (hip) domain name system (dns) extension. RFC 5205, 2008.
[28] M. Stiemerling, J. Quittek, and L. Eggert. Nat and firewall traversal issues of host identity protocol (hip). RFC 5207, 2008.
[29] Y.Jukka and N.Pekka. Blind : A complete identity protection framework for endpoints.