SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2011)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.387

RESEARCH ARTICLE

SIPA: generic and secure accounting for SIP

Alexandros Tsakountakis*, Georgios Kambourakis and Stefanos Gritzalis

Laboratory of Information and Communication Systems Security, Department of Information and Communication Systems Engineering, University of the Aegean, Karlovassi, GR-83200 Samos, Greece

ABSTRACT

Authentication, authorization, and accounting services provide the framework on top of which a reliable, secure, and robust accounting system can be built. In a previous work of ours, we have presented a flexible and, most importantly, generic accounting scheme for next generation networks. In this paper, we substantially improve our previous work by providing the required Diameter application namely SIP-Accounting (SIPA) that enables the use of our accounting scheme for Session Initiation Protocol (SIP) services. Additionally, in an effort to protect the service providers and the end users against accounting frauds, we implement an add-on mechanism referred to as SIPA+ to combat attacks targeting the core accounting functions and the integrity of the respective accounting messages. Using the implemented SIPA and SIPA+ prototypes, we conducted a complete set of experiments testing several configurations and two distinct scenarios. The results reveal that the proposed accounting system and its security add-on are fully operable in SIP environments without incurring much cost in terms of performance and overhead. Copyright © 2011 John Wiley & Sons, Ltd.

KEYWORDS

SIP; diameter; accounting; Diameter application; security

***Correspondence**

Alexandros Tsakountakis, Laboratory of Information and Communication Systems Engineering, University of the Aegean, Karlovassi, GR-83200 Samos, Greece.
E-mail: atsak@aegean.gr

This paper is part of the 03ED675 research project, implemented within the framework of the "Reinforcement Programme of Human Research Manpower" (PENED) and co-financed by national and community funds (25% from the Greek Ministry of Development—General Secretariat of Research and Technology and 75% from EU—European Social Fund).

1. INTRODUCTION

Recent increase in the demand for voice and multimedia delivery has raised great interest in the Session Initiation Protocol (SIP) [1]. SIP constitutes an application layer control signaling protocol able to establish, control, and terminate multimedia sessions. Without doubt, SIP is the cornerstone for most emerging communication services. This is because SIP-based applications can be deployed by all communications service providers from traditional fixed line and mobile operators to Internet service providers. During the last years, SIP is under extensive attention by researchers in terms of performance, mobility, provided services, and security [2–8]. In fact, security constitutes a key aspect in the effort to establish SIP as the predominant protocol in multimedia session management. In addition, recently, there is a growing interest in studying SIP in conjunction with authentication, authorization, and accounting (AAA) [9]. The significance of providing proper bindings between SIP and AAA is further emphasized by the choice of SIP by the Third Generation Partnership Project consortium as the multimedia management protocol of 3G networks IP Multimedia Subsystem (IMS) [10], and Diameter [11] as the default protocol to provide AAA services. That is, while SIP can be used to manage multimedia sessions, Diameter can provide the necessary AAA services. In such an environment, the SIP server operates as a client of the Diameter server (also known as AAA server).

Accounting refers to the tracking of the consumption of network resources by users. This information may be used for planning, management, billing, or other purposes. However, although accounting is imperative for any AAA infrastructure, it is often neglected in the literature as researchers mostly focus on authentication, authorization, and performance issues. In our opinion however, the penetration of any new technology in the market is severely affected by the trustworthiness of the underlying accounting mechanisms with particular emphasis to billing. To put it another way, accounting is very important for both the service provider, as his revenue relies upon it, and the subscriber in order to keep his faith in the network operator with whom he holds a contractual agreement. At the same time, accounting mechanisms are always an attractive target to attackers who can potentially exploit any vulnerability in the system to gain profit or cause economic frauds just for fun [12,13].

In this paper, we focus on accounting-related issues in the context of SIP services. This is because, although SIP is very capable of managing multimedia sessions, all details regarding accounting are beyond its scope. On the other hand, modern AAA protocols such as Diameter can be used to convey accounting-related information between a SIP client and an accounting server but fail to define how security needs to be considered or how horizontal and vertical handoffs should be treated. For example, secure delivery of the user profile or Service Level Agreements (SLAs) between users and operators or between operators is not guaranteed during an inter-domain handoff. This means that a more generic accounting scheme is necessary, able to both fulfill security requirements and support smooth network and service integration.

Our Contribution: Motivated by the aforementioned problems, this work capitalizes on our generic accounting system discussed in [14] and significantly enhances it to provide compatibility with SIP services and cope with SIP-specific requirements. On top of that, in an effort to strengthen security, we implement a mechanism to ensure the validity and accuracy of the accounting services. The outcome of this work is a fully fledged accounting solution for SIP, namely SIP-Accounting (SIPA). SIPA comes into two versions; the standard one and SIPA+, which delivers increased security features. Similar to [14], the SIPA system is also implemented as a Diameter application. We evaluate SIPA and SIPA+ in terms of performance and overhead using a properly designed test bed. The results show that our Diameter application is sound, is relatively lightweight, and can be easily implemented in real-life network architectures. Additionally, we demonstrate that the overhead imposed by SIPA is acceptable in comparison with the ordinary case, that is, standard AAA SIP installations. To the best of our knowledge, this is the first work to (i) offer a fully fledged next generation networks (NGN)-aware SIP accounting system and (ii) present a comprehensive performance evaluation of a SIP accounting system inside the AAA terrain.

The rest of the paper is structured as follows: The next section provides the necessary information on how AAA can be used in SIP realms. All mandatory procedures ranging from user authentication to accounting information delivery are presented. Moreover, the inability of the default Diameter protocol to handle complex accounting procedures is pointed out. A brief discussion of our previous work is given in Section 3. This is considered necessary because we use the same principles and tools to design and implement SIPA. Section 4 elaborates on the implementation of the SIPA prototype. The analysis includes all the new custom-made Diameter commands and attribute–value pairs (AVPs), which SIPA embeds. Section 5 discusses security issues related to SIP accounting and presents some available solutions. This section also presents the selected solution and discusses how it can be implemented in the form of Diameter AVPs and commands. Our test-bed architecture and performance evaluation are given in Section 6. Section 7 surveys related work. Finally, Section 8 concludes and gives directions for future work.

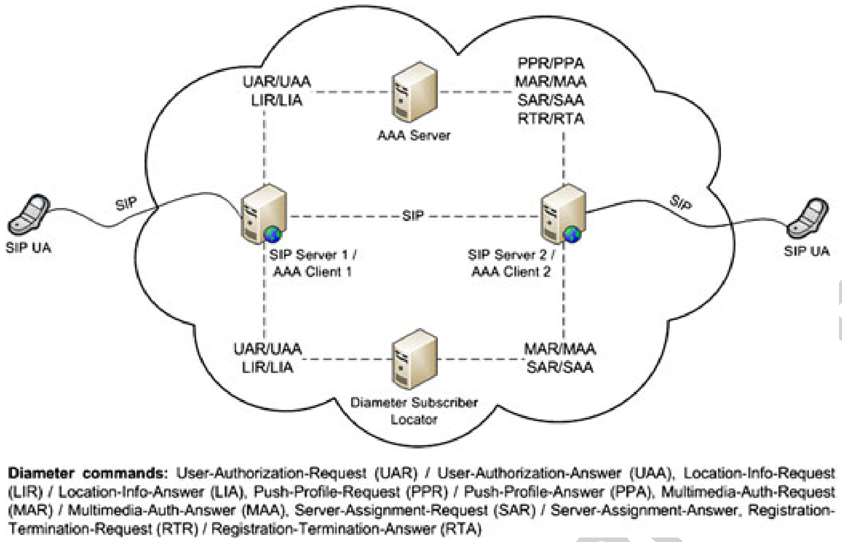
2. AUTHENTICATION, AUTHORIZATION, AND ACCOUNTING IN SESSION INITIATION PROTOCOL

The Diameter AAA standard is built to provide a base protocol framework that can be easily extended to provide further operability. The Diameter base protocol [11] defines the minimum requirements for an AAA protocol and provides the basic functionality. A Diameter application is able to extend the base protocol by adding new Diameter commands and AVPs to support even the most sophisticated AAA functions. Examples of Diameter applications include the Diameter Network Access Server (NAS) application [15], the Diameter Mobile IPv4 application [16], the Internet Engineering Task Force (IETF) Diameter SIP application [17], and others. In practice, when the need for a new AAA-related functionality arises, it can be provided by extending the base protocol (or another existing Diameter application) using new AVPs and/or commands. Furthermore, it may be implemented in the form of an entirely new Diameter application.

When SIP is chosen to manage multimedia sessions, the Diameter SIP application is used to provide all the necessary Diameter functionality. More specifically, this is a Diameter application that allows an AAA client, which acquires SIP-based IP multimedia services, to request AAA information from an AAA server. Other supported capabilities include rudimentary routing and management of updated user profiles. Overall, the Diameter SIP application can be used by SIP configurations where an interface to an AAA infrastructure is required to authenticate, authorize, and support the accounting of consumed SIP resources. Note, however, that no open-source implementation of Diameter SIP application exists until now. In fact, the first such implementation is provided in the context of this work.

In such an environment, SIP users create SIP requests in order to access SIP resources and acquire services. Usually, the respective home network (HN) needs to authenticate and/or authorize the usage of these resources. In addition, the SIP server and the AAA client are co-located in the same network node, which practically means that a SIP server actually implements an AAA client. Therefore, it is imperative that all network elements support the Diameter SIP application apart from the base Diameter protocol. Figure 1 depicts a generic architecture for the Diameter SIP application. SIP server 1 receives a SIP request from a SIP user agent (UA) and proxies it to SIP server 2, which in turn will deliver the actual service to the end user. The AAA server serves both AAA clients, which coexist with SIP servers 1 and 2, respectively. Finally, the Diameter

DOI: 10.1002/sec



Diameter commands: User-Authorization-Request (UAR) / User-Authorization-Answer (UAA), Location-Info-Request (LIR) / Location-Info-Answer (LIA), Push-Profile-Request (PPR) / Push-Profile-Answer (PPA), Multimedia-Auth-Request (MAR) / Multimedia-Auth-Answer (MAA), Server-Assignment-Request (SAR) / Server-Assignment-Answer, Registration-Termination-Request (RTR) / Registration-Termination-Answer (RTA)

Figure 1. General Session Initiation Protocol (SIP) authentication, authorization, and accounting (AAA) architecture.

subscriber locator (SL) serves the purpose of locating AAA servers that store user-related data (i.e., user profile and SLA) on behalf of the requesting AAA or SIP servers. The figure also presents the Diameter commands that can be possibly exchanged between the network elements. Section 6 discusses all these commands in detail.

The Diameter SIP application provides only basic support for accounting services. Specifically, the only functionalities that can offer are as follows: (i) an AAA client is able to request the network addresses of accounting servers from an AAA server, and (ii) it is possible to locate the AAA server that keeps user-related data through the Diameter SL that implements the Diameter redirect mechanism [11]. Another potentially interesting function related to accounting is the ability of a SIP server to asynchronously access a user profile whenever it has been updated. Furthermore, the Diameter SIP application is often used in conjunction with the Diameter credit-control (DCC) application [18], which implements a real-time credit-control mechanism for prepaid users. In this case, the Diameter SIP application is capable of informing the engaged network elements about the addresses of credit-control servers that can perform the respective credit-control functions. In a nutshell, the Diameter SIP application supports basic connectivity and security functionalities, but accounting details need to be identified and implemented by the network operators or service providers *per se*. Under these circumstances, a generic and secure accounting system that could provide all the required accounting functionality (see Section 3.1) without resorting to custom-built solutions would be highly appreciable.

Motivated by the aforementioned observations, we implement SIPA. SIPA actually builds upon the Diameter SIP application and extends its functionality. By doing so, SIPA retains all the properties of the Diameter SIP application being generic and flexible at the same time. Hence, it can be straightforwardly incorporated into existing infrastructures and is able to satisfy all requirements imposed by SIP implementations. Table I presents the basic Diameter commands defined in the context of the Diameter SIP application. Note that the same commands are the keystones in SIPA implementation process.

3. SIPA PRELIMINARIES

In our previous work [14], we presented and evaluated a generic accounting scheme that capitalizes on state-of-the-art AAA technology and can be deployed for delivering proper accounting services to NGN [19]. This accounting system was designed with flexibility and adaptability in mind and thus can be straightforwardly incorporated into current and future providers' infrastructures. Moreover, it is applicable regardless of the underlying network access technology. It was built as a new Diameter application by extending the Diameter base protocol and utilizing information from the NAS application. For its realization, we assumed the generic architecture of NASs acting as AAA clients as described in [15]. Thus, no inherent support for SIP network elements and SIP-specific functions was available, but the generic nature of the accounting system makes possible future extensions to accommodate any modern service such as SIP. This section describes the main components and internal mechanics of the original accounting system. This is considered necessary as SIPA takes the original system as a reference and extends it where necessary.

3.1. Accounting system requirements

As already pointed out, the motivation behind the implementation of a new AAA accounting system stems from the observation that, so far, little has been carried

Table I. Diameter Session Initiation Protocol (SIP) application commands reused in the context of SIPA.

Command	Abbreviation	Functionality
Location-Info-Request	LIR	The AAA client in a SIP server issues this command to request routing information from the corresponding AAA server.
Location-Info-Answer	LIA	The AAA server sends this command in response to a previously received LIR command.
Multimedia-Auth-Request	MAR	A Diameter client issues this command to request user authentication from the Diameter server.
Multimedia-Auth-Answer	MAA	It is issued to acknowledge a previous MAR command.
Push-Profile-Request	PPR	The AAA server sends this command to an AAA client to update the user profile or the corresponding accounting directives.
Push-Profile-Answer	PPA	It is used to acknowledge a previously received PPR command.
Registration-Termination-Request	RTR	The AAA server sends this command to an AAA client to request deregistration of a specific user.
Registration-Termination-Answer	RTA	It is issued to acknowledge a previous RTR command.
Server-Assignment-Request	SAR	It is issued by an AAA client to inform the corresponding AAA server about the allocation of the SIP server to a given user name (URI). Also, this command may be used from an AAA client to request several services from an AAA server, for example, user deregistration. The SIP-Server-Assignment-Type AVP defines the type of service request.
Server-Assignment-Answer	SAA	It is used as a response to a previous SAA command. It is frequently issued to transfer the user profile to the requesting AAA entity.
User-Authorization-Request	UAR	It is issued by a Diameter client (collocated with a SIP server) to request from a Diameter server authorization for the SIP UA to route a SIP REGISTER request.
User-Authorization-Answer	UAA	The Diameter server sends this command in response to an UAR to indicate the result of the requested registration authorization. Additionally, it can inform of a collection of capabilities that may assist the Diameter client to select a SIP proxy for the UA.

AAA, authentication, authorization, and accounting, AVP, attribute–value pair; URI, Universal Resource Identifier.

out for the accounting part of the AAA framework. Secondly, it is driven by the fact that current custom-built accounting solutions deployed by network operators are not capable of dealing with sophisticated modern requirements imposed by the multi-domain heterogeneous and ubiquitous network terrain. For example, existing accounting systems treat inter-domain handoffs as if they occur inside the home Administrative domain. Towards this direction, it is necessary to define the basic requirements that any new accounting system should be able to cope with, taking into consideration the following: (i) the heterogeneous network access environment; (ii) the multi-network operator relationship model; (iii) the existence of many innovative technologies possibly incompatible with each other; and (iv) the large number of mobile user population; note that each user is a potential customer or service-requiring entity for all existing network operators and service providers. In a nutshell, the desirable requirements that any novel accounting system must meet are the following:

Generic: The new accounting system should be applicable irrespective of the underlying network access technology. In this way, forthcoming technologies should be easily incorporated.

Distributed: The magnitude and complexity of current demands for accounting services can only be tackled through distributed architectures. A distributed architecture also helps mitigate future problems and technical failures and/or bottlenecks.

Secure: Without doubt, security is critical during the accounting procedure. Data privacy, confidentiality, and integrity should be ensured. On top of that, the protection of user's private information, that is, confidentiality, must also be kept to an acceptable level. Private data should be safely stored and never be transmitted to any party other than the one that the user has a contractual relationship with. At the same time, the accounting data collected on behalf of a user should be securely and reliably communicated between the Administrative parties involved. Therefore, the confidentiality and integrity of accounting data in transit are of major importance here.

Transparent to users: Users must receive a single billing report regardless of the number of operators or other charging parties involved in the process of accounting.

3.2. Accounting system details

This section presents the basic components and the internal mechanics of the accounting system presented in [14]. Figure 2 depicts the general architecture of this system. All network elements communicate using the Diameter protocol and support our Diameter accounting application. An AAA client is a device residing at the edge of the network that provides access control and forwards any user queries towards the AAA server in charge. Moreover, it generates AAA messages to request AAA services on behalf of the user. In terms of accounting, AAA clients are the network elements that perform the tasks of gathering accounting metrics and sending them to the corresponding AAA servers.

The AAA servers on the other hand receive AAA messages (also called commands) from the AAA clients and perform the appropriate AAA services. According to [14], the AAA servers are responsible to calibrate accounting settings on AAA clients; request, receive, and transform accounting metrics into accounting records; communicate with other AAA servers inside or outside the local Administrative domain; manage user-related information; and finally, store accounting data. Table II summarizes the functionality of all the engaged network entities in the context of SIPA.

Moreover, during the accounting process, an AAA server can take either the role of the Root server or that of the Administrative server. The Root server is an AAA server inside the home domain responsible for a specific user. That is, the server that has already successfully completed the authentication and authorization process and granted access to the user through the AAA client. In several scenarios, the AAA server that the user initially attaches to might not be suitable to provide the required services; thus, AAA requests may be proxied to a new AAA server that will be granted the role of the Root server. From now on, in terms of accounting, the Root server will be responsible for that specific user. Therefore, the same AAA server will be used for collecting accounting records from the respective Administrative servers throughout the entire user session. In short, the Root server initializes and terminates the accounting process for a given user.

Upon granting network access to a user, the Root server creates a unique identification number (ID) and, at the same time, stores in its database a record mapping the newly created ID with the actual user ID as shown in Figure 3, case A. The actual user ID may be a permanent one, such as the user's International Mobile Station Identifier or a Universal Resource Identifier (URI), or a temporary ID, such as the Network Access Identifier [20], or even a pseudonym. The first ID that the Root server creates is called *Master ID*. This ID can be changed, updated, or deleted only by the Root server. The Root server is also responsible for accepting frequent requests for accounting information by the Administrative servers as well as for the preparation of the final invoice to be sent toward the subscriber. Any Administrative server, on the other hand, will respond to an accounting query sent by a Root server.

The Administrative server is initially the same as the Root server. As the user roams from one domain to another, handoffs occur, and the user may need to attach to a different AAA client or even require the services of a new AAA server. Consequently, the Administrative server is the local AAA server, which is at the given moment responsible for the user. It is important to note that the Administrative server can be an AAA server that is located in the Administrative domain of a foreign network operator. This server is responsible for collecting accounting records and keeps track of the user activities while he or she remains under its supervision. Practically, the Administrative

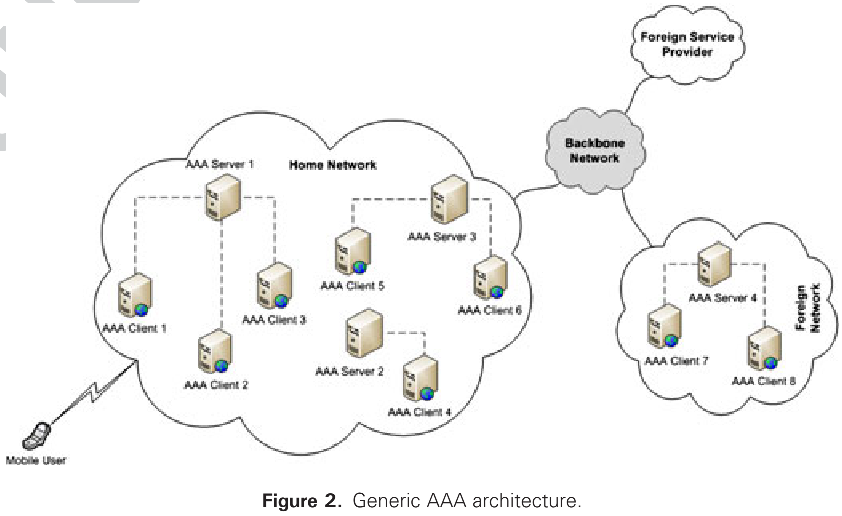


Figure 2. Generic AAA architecture.

Table II. Network entities functionality

Network entity	Functionality
AAA client	Provides access control and generates AAA messages to request AAA services from the AAA server in charge on behalf of the user. In terms of accounting it is additionally responsible for gathering accounting metrics to be forwarded to the respective AAA server.
Generic AAA server	Receives AAA messages from the AAA client, performs the appropriate AAA services and informs the AAA client of the outcome.
Root server	In terms of accounting it is the AAA server inside the HN being responsible for the accounting procedure of a specific user. Is responsible for the following: (i) managing and storing user-related information; (ii) collecting accounting records from the corresponding Administrative servers; (iii) creating the Master ID and storing the correlation between the user profile, the Master ID, and the accounting records; and (iv) preparing the final invoice to be sent to the end user.
Administrative server	In terms of accounting, it is the local AAA server, which is at the given moment responsible for the specific user. Also, it may be an AAA server outside the HN and therefore keeps limited information (i.e., user SLA and other routing or accounting data). Configures accounting parameters on the respective AAA clients, triggers the initiation or termination of the accounting metrics collection process and receives the accounting metrics from the AAA clients. Then, the accounting metrics are transformed into accounting records and the correlation between the accounting records, the Event ID, and the Reference ID is securely stored and forwarded to the Root server when asked.

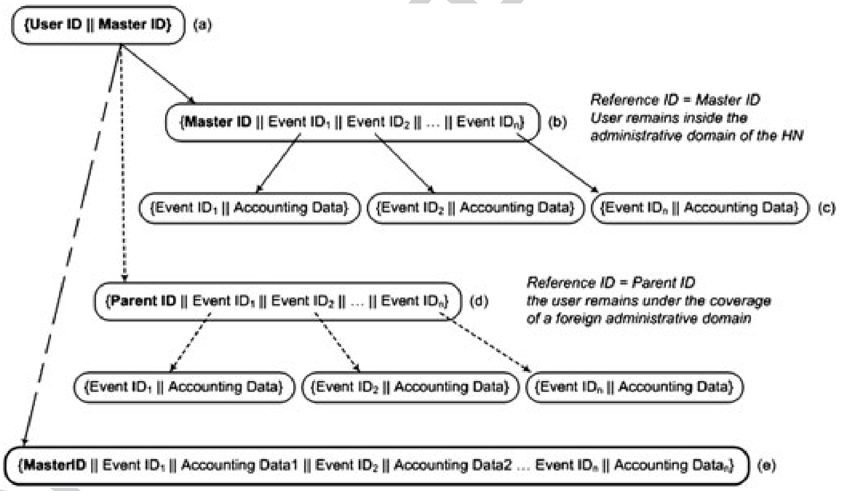

{User ID || Master ID}
(a)
{Master ID || Event ID1 || Event ID2 || … || Event IDn}
(b)
Reference ID = Master ID
User remains inside the
administrative domain of the HN
{Event ID1 || Accounting Data}
{Event ID2 || Accounting Data}
{Event IDn || Accounting Data}
(c)
{Parent ID || Event ID1 || Event ID2 || … || Event IDn}
(d)
Reference ID = Parent ID
the user remains under the coverage
of a foreign administrative domain
{Event ID1 || Accounting Data}
{Event ID2 || Accounting Data}
{Event IDn || Accounting Data}
{MasterID || Event ID1 || Accounting Data1 || Event ID2 || Accounting Data2 … Event IDn || Accounting Datan}
(e)

Figure 3. (a) Mapping user information with the Master ID, (b) mapping the Master ID with Event IDs, (c) mapping the event ID with accounting data, (d) mapping the Parent ID with Event IDs, and (e) record stored in the database by the Root server.

server configures accounting parameters on the AAA client and triggers the initiation or termination of the accounting metrics collecting procedure from the same entity. Moreover, it receives all accounting metrics that are later on converted to accounting records to be sent towards the Root server. While the user moves from one AAA client to another, the current Administrative server terminates accounting on the old AAA client and asks the new one to take control and initialize the proper accounting procedures. Each Administrative server holds only limited information about the actual user. That is, it keeps only the required SLA parameters needed for charging as well as a reference to an ID sent to it by the previous Administrative server.

Each time the user initializes an event that needs to be tracked and metered, the Administrative server will create a new unique ID, called *Event ID*, mapped to that particular event. Each Event ID must be globally unique, so for instance, it could take the form of a triplet: {Administative_-Server_Name_or_IP || Event_ID || Current_time_in_milliseconds}. The server will securely store, in the corresponding database, the correlation between the newly created Event ID and the received *Reference ID* as shown in Figure 3, cases B and D. For multiple events created by the same user, the corresponding IDs will be utilized to track all user activities. A database is accessed to securely store records binding the user Event IDs with accounting data as shown in Figure 3, case C. When a user leaves the current Administrative server or when required for other purposes, all accounting gathered records will be sent towards the Root server. The Root server will eventually combine all events and store an accounting record in the form of that shown in Figure 3, case E.

The notion of the Reference ID contains two discrete entities. While the user remains inside the Administrative domain of the HN, the Reference ID is the Master ID created by the local Root server. On the other hand, while the user remains under the coverage of a foreign Administrative domain, a *Parent ID* takes the role of the Reference ID, as described further down. Thus, for a given user, the same Master ID is used during a session, whereas several Parent IDs may be utilized in parallel. This happens because it is vital that, every time a foreign AAA server is involved, a new ID should be used as a reference. The Root server in the case of the Master ID and the previous Administrative server in the case of the Parent ID can be extracted from the Master and Parent ID values correspondingly so that the current Administrative server knows where to send the accounting records.

In case the user moves to the domain of a foreign network operator, the same principles apply, but confidentiality requirements suggest the use of a new identifier other than the Master ID to be utilized as a reference for any new Event IDs. This is accomplished by a new identification number that we call Parent ID. The Parent ID is created by the Administrative server in the home domain to be sent to the new Administrative server inside the foreign domain. This Parent ID will thereafter be used as a reference to any newly created Event IDs. The Parent ID notion serves a dual purpose. First, it constitutes a completely new reference neither created nor relevant to the initial Master ID or the actual user ID. Thus, even when the Master ID is used as a reference to Event IDs, it remains inside the home domain and is never become available to a server inside the FO. This assists to further protect the user real identity and other related confidential information. Secondly, the Parent ID helps to clearly distinguish the role of the Parent ID from that of the Master ID.

It is required that, in case of a vertical handoff, the new Administrative server does not contact directly the previous Administrative server inside the foreign domain. Instead, it requests all necessary information to be sent during authentication by the lattermost Administrative server inside the home domain. This server will send the required SLA and any other charging instructions as well as the user's Parent ID. The latter is a new Parent ID different than any other previously occupied for the same user. As the user terminates all actions or when asked for other purposes, each engaged Administrative server inside the foreign domain that tracked user activities for some time will send the relevant accounting records to the corresponding Administrative server inside the home domain. The Administrative server will later on forward them along with its own collected accounting records to the Root server inside the home domain.

The aforementioned accounting system was implemented in the form of a Diameter accounting application. As it is pointed out in [14], this scheme requires no modifications to hardware and minor in the software of the involved network entities (i.e., AAA server, router, and AAA client). Additionally, it attains full compatibility with the base Diameter protocol, can be easily conveyed into any present or future AAA protocol, and is able to support modern brokering environments [11]. The security analysis presented in [14] shows that the accounting system provides at least the same level of security that the default Diameter protocol mandates. In addition, a complete set of experiments testing several configurations shows that the imposed penalties in terms of service time and resource utilization are considered rather insignificant, if not negligible in some cases.

The accounting system described in this section provides the basis on top of which SIPA is designed. The next sections present the necessary amendments and additions that make SIPA a good choice for any SIP realm.

4. SIPA: A NOVEL ACCOUNTING SYSTEM FOR SIP

4.1. Requirements and architecture

Before everything else, SIPA needs to take into consideration: (i) any new requirements stem from the SIP environment like Quality of Service (QoS) issues and/or real-time service delivery; (ii) the different practices and mechanisms behind core SIP functions, like user authentication, registration or session termination; and (iii) any disparities in the SIP architecture Voice Service Providers (VSP) may implement [17].

Generally, in order to deliver a fully fledged SIP-oriented accounting mechanism, we need to implement the following: (i)the basic SIP operations mandated by the Diameter SIP application; this will provide the necessary connectivity and other functionality required by SIP, for example, authentication, authorization, and session management; and (ii) the actual accounting system by incorporating the previous functionality and adding new behavior where necessary; this can be realized by incorporating new AVPs into the existent Diameter commands in the first case and through new Diameter commands in the latter. Figure 4 shows the general architecture of SIPA. Note that the accounting system depicted supports both SIP-only and general purpose NAS as AAA clients.

Before we continue with the analysis of the SIPA system, it is important to note a few assumptions and observations stemming from the nature and requirements of the SIP architecture.

(1) Several real-life VSPs do not implement all SIP functions mandated by the Diameter SIP

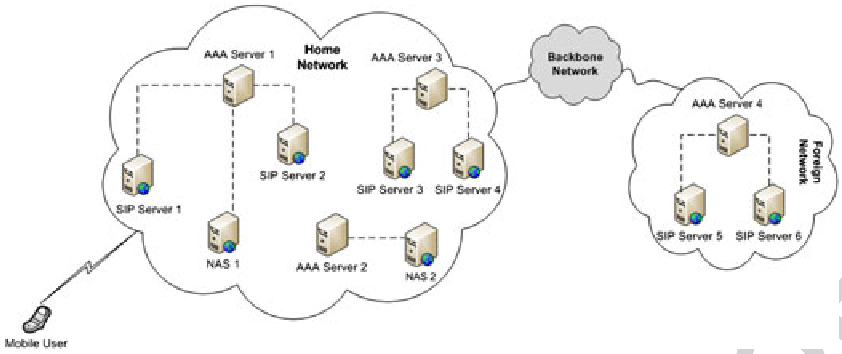


Figure 4. Accounting system configuration supporting Network Access Server and SIP servers as AAA clients.

application. It is therefore important for SIPA to only utilize those that are necessary and thus expected to be implemented by all VSPs. At the same time, it is desirable to offer several methods to perform certain critical operations and not rely on specific architectures and practices. For example, SIPA offers to network entities more than one methods to discover and update user profile~~s~~ and/or track user IP address.

(2) The Diameter SIP application advices the actual authentication of the user requesting SIP services to be performed either by the SIP or the AAA server. The *SIP-Auth-Data-Item* AVP inside a standard Diameter *Multimedia-Auth-Answer* command denotes the decided choice. Without loss of generality, we have chosen the AAA server as the entity to provide the authentication service.

(3) It is possible that AAA clients implementing the RADIUS protocol [21] need to communicate with an AAA server utilizing the Diameter protocol. This is a common practice in real-life SIP scenarios. Migration from RADIUS is generally feasible, but several incompatibilities prevent the accounting system to cover all possible accounting requirements. Thus, SIPA focuses on the Diameter protocol with the requirement that all entities utilize Diameter as the AAA protocol in charge. Diameter is regarded a successor to RADIUS, fixing all RADIUS deficiencies [11]. Specifically, Diameter is more extensible than RADIUS and consolidates many of the features that vendors have already implemented. It unifies accounting more closely within its structure, works with reliable transport protocols such as Stream Control Transmission Protocol (SCTP) [22], and is an open, IP-centric protocol that will enable network vendor choice and other improvements that will benefit billing. Diameter has its primary applications in novel network designs, such as wireless AAA, rather than in applications where RADIUS is well established. Therefore, we can argue that SIPA is based on most modern technology.

(4) A basic architecture assumption within the Diameter SIP application is that all the data related to a user is kept in a unique Diameter server, typically operating in a redundant manner. To circumvent this limitation, the Diameter SIP application mandates the implementation of a Diameter SL that is able to locate the requested SIP or AAA server through the Diameter redirection mechanism. SIPA will adhere to that specific guideline as well.

(5) The SIP servers 1 and 2 depicted in Figure 4 are actually logical units, meaning that in real life, VSPs utilize farms of probably stateless SIP servers that operate in a redundant configuration. There should therefore be no guarantee that two consequent requests will arrive at the same SIP server 1 or 2. If false, the standard Diameter commands *User-Authorization-Request* (UAR)/*User-Authorization-Answer* (UAA) (see Table I) are used to discover the correct SIP server.

(6) An AAA server does not necessarily perform every possible AAA function. Instead, a server may be dedicated to only perform SIP user authentication or billing (i.e., acting as a billing server). For simplicity reasons, in the following, we only use the term "AAA server" although it is possible that several AAA servers exist in the background. This is a general assumption that should be taken into consideration when AAA architectures are deployed.

4.2. SIPA in detail

Conforming to the IETF guidelines [11] for the introduction of new Diameter applications, we have tried to keep the number of new commands and AVPs to the minimum possible and instead reuse any predefined one when applicable. This section presents the Diameter commands and AVPs that SIPA incorporates. In fact, these commands are responsible for offering SIP functionality. This results in an accounting system that can additionally support SIP service delivery. All amendments and additions that we made to the Diameter SIP application are summarized in Table III and explained in greater detail subsequently. In

Table III. Proposed extensions and additions to the Diameter SIP application.

New Diameter command/AVP	Added to command	Purpose
Accounting-Records-Request (ARR)	–	Diameter command used to transfer accounting records between AAA servers.
Accounting-Records-Answer (ARA)	–	Diameter command used to acknowledge the delivery of the accounting records.
Accounting-Records	ARR	AVP to convey the accounting records between AAA servers.
Accounting-Response	ARA	AVP to acknowledge the correct delivery of accounting records or request retransmission.
Event-Id	MAA, PPR	AVP to convey a new Event ID.
Force-Delivery	RTR	AVP to request from a AAA client to send all accounting metrics to the respective AAA server prior to the deregistration of the user.
IP-Address-Realm	CER, CEA	AVP to store the IP address to be assigned to a newly arrived user. Used to inform the AAA server of the users that the AAA client currently services.
Peer-Accounting-Role	CER, CEA	AVP to define the role of the AAA peer regarding accounting. It can take one of the three values, namely "Root Server", "Home_Administrative Server" or "Foreign_Administrative Server".
Reference-Id	CER, CEA	AVP to transfer the reference ID between the AAA servers.
Request-Accounting-Details	CER	AVP used to request accounting directions for a specific user and/or service.
Root-Server-Name	CER, CEA	AVP to store the identity of the Root server.
Setup-Accounting	MAA, PPR	AVP to inform the AAA client to initiate or terminate the accounting process.
SIP-User-Data AVP	CEA	AVP that allows an AAA server to transport user-specific data (e.g., a user profile) to the AAA client inside the SIP server.

addition, Figure 5 depicts a typical message flow in a SIP AAA environment.

An important process that takes place several times and necessarily before a connection is established between two AAA nodes is the negotiation of entities capabilities. It allows the discovery of a node's identity and its capabilities, for example, Diameter version and supported Diameter applications. Through this process, every AAA node inside

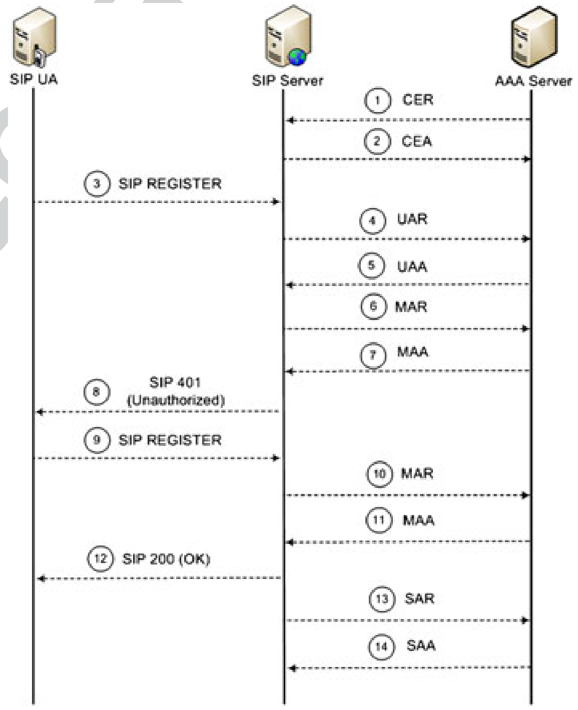


Figure 5. Typical SIP AAA message flow.

DOI: 10.1002/sec

a domain is aware of which AAA clients are initially serviced by which AAA servers, and which users are serviced by which AAA clients. The aforementioned functionality is realized through the exchange of the Diameter commands *Capability-Exchange-Request* (CER) and *Capability-Exchange-Answer* (CEA). A new AVP namely *IP-Address-Realm* has been incorporated into those commands to store a group of user IP addresses that the sender (acting as an AAA client) currently services. This AVP is not set as mandatory as it is only applicable when the issuer of the CER or CEA command is an AAA client. On the contrary, when an AAA server creates these messages, it leaves the *IP-Address-Realm* AVP empty (null). A second custom-made AVP that we added in these two commands is called *Peer-Accounting-Role*. This AVP is used to inform the receiving AAA entity about the exact role that the AAA server would have concerning accounting. This AVP is used only by AAA servers and can take one of three distinct values, namely "Root Server," "Home_Administrative Server," or "Foreign_Administrative Server."

Taking advantage of the fact that the exchange of CER and CEA is always performed before session establishment, we have utilized these commands to transfer the Master ID or Parent ID from one AAA server to another. This is achieved through the new *Reference-Id* AVP. Additionally, the *Request-Accounting-Details* AVP is added to the CER command, making it able to request specific accounting instructions for a given user and/or a given service. Likewise, the CEA command now carries the *SIP-User-Data* AVP that contains the actual accounting instructions, that is, those derived from the user-profile. Finally, in order to store the identity of the current Root server inside the home domain, a *Root-Server-Name* AVP is appended. This only provides an additional method that AAA servers can utilize to discover the correct Root server. It is actually an effort to provide more than one method for critical operations. The same may be achieved through the *Reference-Id* AVP as it also contains the identity of the Root server.

The aforementioned new AVPs do not interfere with the CEA/CER functionality nor add any significant time or computational overhead. On the contrary, they are generic in the sense that they are applicable regardless of the nature of the AAA client or the acquired service. At the same time, they do not raise any compatibility issues with other common Diameter applications that may affect accounting tasks (e.g., a Credit-control application).

Before a user can obtain any SIP service, authentication and authorization need to be successfully completed. This process is triggered when a SIP UA sends a SIP REGISTER request to a SIP server. The AAA client inside the SIP server needs to contact the AAA server in charge, in order to (i) decide if the user is allowed to receive the requested service and (ii) be informed about routing information and specifically the address of a local SIP server to service the user. This is achieved through the Diameter UAR command. A Diameter UAA command is issued in response by the AAA server informing the AAA client (co-located with the SIP server) about SIP servers capable of serving the user. Finally, the initial AAA server will decide about the corresponding SIP server and forward the SIP REGISTER request to it.

At this point, the AAA client inside the new SIP server needs to request authentication from the AAA server. A Diameter *Multimedia-Auth-Request* (MAR) command is issued to request authentication and to inform the AAA server about the SIP URI of the SIP server. This is carried out in order the SIP server to be reserved as the entity in charge for the specific user. The AAA server responds by issuing a Diameter *Multimedia-Auth-Answer* (MAA) command. Note that the authentication procedure does not terminate at this point. In fact, the actual authentication is now initialized. The MAA command includes a challenge that the SIP server will use to map into the WWW-Authenticate header in a SIP 401 (Unauthorized) response to be sent back to the initial SIP server and eventually to the SIP UA.

Afterwards, the initial SIP server receives a new SIP REGISTER, which this time includes the user credentials. The initial SIP server becomes aware of the SIP server in charge through the exchange—with the AAA server—of a new pair of UAR and UAA commands. The initial SIP server is able now to forward the SIP REGISTER request to the SIP server in charge that in turn will extract the user credentials. Upon that, it will forward the user credentials to the AAA server by issuing a new MAR command. At this point, the AAA server is able to authenticate the user and inform the SIP server about the outcome by issuing a new MAA command. Finally, the SIP server in charge can generate a SIP 200 (OK) message and inform the initial SIP server and, eventually, the SIP UA.

Some additional authentication/authorization mechanisms applicable in a SIP AAA mixed realm are available in the Diameter SIP application, but all of them make use of UAR/UAA and MAR/MAA commands. Note that this work does not emphasize on the authentication/authorization part. In fact, SIPA is applicable regardless of how the actual authentication is performed and only amends the MAR/MAA commands to support the new accounting system requirements.

Regardless of the authentication/authorization mechanisms in place, the related activities terminate with the delivery of a MAA command originated from an AAA server and sent towards an AAA client inside a SIP server. From this point on, the accounting process needs to be initiated. This can be achieved with a new AVP incorporated inside the MAA command, namely *Setup-Accounting*, to inform the AAA client to either initiate or terminate the accounting process. Another addition to the MAA command is a new AVP called *Event-Id*, which carries an Event ID that the AAA client will use to assign accounting metrics. In our previous work, it was stressed that whenever a new Event ID is utilized, authentication is also mandatory. Thus, the MAA command is ideal to convey the Event ID.

The default method to forward accounting metrics from an AAA client to an AAA server is through the base Diameter *Accounting-Request* (ACR) standard command. An *Accounting-Answer* (ACA) command is sent in response by the AAA server to notify the AAA client of a successful delivery or a (re)send request. It should be noted here that the ACR only handles accounting metrics originating from an AAA client and sent towards an AAA server.

Session Initiation Protocol-Accounting also requires a mechanism to enable AAA servers to forward accounting records towards other AAA servers. Specifically, an Administrative server needs to reliably transfer the accounting records constructed after processing the accounting metrics log files. The log files are sent by the respective AAA clients to another Administrative server or the Root server. As the available Diameter commands cannot cope with this situation, we introduce a new pair of commands, namely *Accounting-Records-Request* (ARR) and *Accounting-Records-Answer* (ARA). The first one carries the actual accounting records through the new *Accounting-Records* AVP. The latter is sent as an acknowledgement to a previous ARR command; note that the newly created *Accounting-Response* AVP will notify the receiver about the delivery result.

During a user session, accounting instructions (i.e., the rules behind the way accounting is performed for the specific user and/or the given service) may need to change. This may be because service parameters have been changed, a new service is requested, the user is now receiving service from a foreign provider, or the user profile has been updated. It is therefore not sufficient to only set up accounting once at the beginning of a user session through the CER/CEA commands. To deal with such a situation, we utilize the *Server-Assignment-Request* (SAR) and *Server-Assignment-Answer* (SAA) command pair. A SAR command is sent by an AAA client to an AAA server to (i) indicate the completion of the authentication process and (ii) request the AAA server to identify from now on the AAA client (through his URI) as the entity in charge allocated for the specific user. At the same time, the AAA client may ask to download the user profile by also issuing a SAR command. A SAR command can also be utilized by an AAA client to request user deregistration from the AAA server in charge. This is achieved through the already defined *SIP-Server-Assignment-Type* AVP. In response to a SAR command, an AAA server will issue a SAA command that, among other information, can convey (forward) the user profile to the AAA client through the *SIP-User-Data* AVP. Note that when the AAA server is notified of a user deregistration via a SAR command, it is mandated that an ACR/ACA command exchange must be triggered. This is necessary in order for the AAA client to forward all gathered accounting metrics for the specific user to the Administrative server in charge.

The aforementioned transfer of a user profile can only be initiated by an AAA client. Additionally, SIPA supports an on-the-fly transfer of a user profile once it has been updated. This method allows an AAA server to send an updated user profile and forces the AAA client to perform accounting according to the new parameters. This is achieved through the *Push-Profile-Request* (PPR) command originated by an AAA server. The *SIP-User-Data* AVP carries the new profile, and the *Setup-Accounting* and *Event-Id* AVPs are now incorporated inside this command. A *Push-Profile-Answer* (PPA) command is issued as an acknowledgment to a previous PPR command.

A complementary function supported by the Diameter SIP application allows an administrator to cancel the registration of a specific user. This is achieved through the *Registration-Termination-Request* (RTR) command. Although not important for SIPA, we require that when this command is used, it also conveys a demand for the AAA client to send all accounting metrics to the respective AAA server prior to user deregistration. To do so, we have populated the RTR command with a new AVP called *Force-Delivery*. When an AAA client receives this command, it triggers a new exchange of ACR/ACA commands between itself and the AAA server in charge, in an effort to transfer all accounting metrics for the specific user. Finally, all routing mechanisms are implemented through the existing *Location-Info-Request* (LIR) command and its acknowledgement, namely *Location-Info-Answer* (LIA).

The aforementioned amendments and additions certainly affect SIP-related AAA operations by creating new requirements and raising performance issues and security considerations. All these concerns will be addressed in the following sections.

5. SIP ACCOUNTING SECURITY

Considering the security of the proposed accounting architecture it is important to notice that the underlying AAA technology materialized by the Diameter protocol guarantees the confidentiality and integrity of the messages in transit. Nevertheless, SIPA like any other accounting system built on top of SIP infrastructure, may be susceptible to inherent SIP flaws or other related attacks [3]. Thus, this section elaborates solely on accounting security and presents the integration of a fully fledged solution into SIPA. Attacks on the SIP protocol in general and corresponding solutions remain out of the scope of this paper.

5.1. Accounting-specific threats in SIP

Every accounting system being part of the AAA architecture needs, among others, to be accurate and reliable. By doing so, it is able to protect both the service provider and the end users against malicious actions undermining the accuracy of billing records. Such malicious actions may originate either by the service provider or the end users. Hence, by proactively repelling such incidents one can increase the trustworthiness level and ensure the survivability of the service provider. Generally, users receiving voice-over-IP (VoIP) services are greatly worried about the accuracy and correctness of their billing records;

DOI: 10.1002/sec

thus, a secure accounting system can boost user confidence in the service provider and, at the same time, leaves little space for disputes among users and service providers over the charging process.

Session Initiation Protocol-Accounting offers native support for secure setup, transfer, processing, and storage of accounting records. This is due to the use of IPsec [23] or Transport Layer Security (TLS) [24] as explained later on in Section 6.1. Nevertheless, similar to any other accounting system, SIPA provides no mechanism to control or validate the actual gathering of the accounting metrics—that will later produce the accounting records—from the AAA clients. More precisely, the accounting system allows an AAA server to send accounting instructions to an AAA client and trigger the accounting process or request the delivery of the accounting records but does not interfere with the actual collection of the accounting metrics performed by the AAA client. Thus, an AAA server and, consequently, the accounting system are not able to control the collection of the accounting metrics and have no means to know that this process has been performed correctly.

Once accounting has been configured and triggered on an AAA client, this entity is responsible to track and store any accounting events. For every new accounting event, for example, a service the user may receive, generally, the AAA client tracks the respective Call Detail Records (CDRs) [25]. That is, the "start" and "end" commands produced whenever a new accounting event initializes or terminates. For instance, a SIP server acting as an AAA client in a SIP environment needs to track for "200 OK" messages, which corresponds to the "start" CDR, and "BYE" messages that corresponds to the "end" CDR. The accurate tracing of the previous messages results in accurate and correct accounting metrics and consequently billing.

As VoIP services are becoming popular, several attacks targeting the accounting functions have been discovered [4,12,13]. These attacks aim to maliciously alter the accounting metrics through the manipulation of the corresponding CDRs and may result in forged charges. For example, in [12,13], the authors demonstrated a way that a malicious user could exploit to manipulate VoIP signaling data in order to avoid charging. Denial-of-service attacks and man-in-the-middle attacks are also feasible as presented in [13]. Recall that such a malicious event may be performed either by the end user or the service provider, but because of the lack of a non-repudiation mechanism in VoIP services, none of them is able to prove if the accounting process was performed correctly.

5.2. Proposed security solution: SIPA+

To cope with accounting-related weaknesses and attacks discussed in the previous section it is highly desirable to incorporate into SIPA an on-demand mechanism that can offer non-repudiation and non-usurpation. Several custom-tailored solutions to deal with specific threats can be implemented, but the choice was to focus on a generic and simple mechanism that does not entail additional or complex network entities or protocols. In this way, the accounting system is able to preserve compatibility with the requirements already identified in Section 3.1.

The choice was therefore to implement the mechanism presented in [26] with a few only minor modifications to cohere with our specific architecture requirements. This mechanism utilizes the underlying AAA infrastructure to provide robust time-stamping services to SIP network entities. Because of its nature, it can be implemented through the use of any AAA protocol such as Diameter and can be amended to cover any VoIP protocol besides SIP.

More specifically, the selected mechanism relies upon the transfer and secure storage of a triplet in the form {OriginHost || Session_Id || Event_Timestamp} by all the engaged network entities, whether it is the local AAA server, the SIP proxy, or other SIP servers. The OriginHost field holds the user's device IP address. It is stressed that this is not the same entity as the standard Diameter AVP *Origin-Host* that denotes the identity of the local host or the host from which a Diameter command has been originated. The Session-Id field is used to correlate an event or a Diameter command with a user session, whereas the Event_Timestamp field is used to record the time that a reported event occurred, in seconds. Through careful concatenations and comparisons of triplets stored in different entities, CDRs can be reliably tracked, and the resolution of disputes becomes therefore feasible. A work in [26] explains the mechanism in greater detail and provides an extended security analysis.

The selected security mechanism can be embedded in SIPA through the following: (i) the incorporation of the required AVPs to convey the triplets in the Diameter commands in use and (ii) the implementation of the functions that perform the secure storage of the triplet, its verification, and any other secondary function required.

In this context, we extend the corresponding Diameter commands with the respective AVPs to make sure that every involved entity generates/receives the abovementioned triplet correctly. It is worth noting that different Diameter commands may already carry part of the triplet. For instance, the ACR and ACA commands contain the fields Session_Id and Event_Timestamp by default. Additionally, it is required that the *Accounting-Record-Type* AVP is set to carry the value "EVENT_RECORD" denoting the tracking of a one-time event. A new AVP namely *Retransmit-After* is created to store the time interval after which a retransmission is allowed as suggested in [26]. Moreover, a new AVP namely *Security-Check* is used to inform about the result of a triplet check. All the aforementioned new AVPs are summarized in Table IV.

This security mechanism comes as an add-on to SIPA and can be easily enabled or disabled upon request. For this reason, in the following sections, we refer to SIPA that employs the security mechanism as SIPA+. Performance

DOI: 10.1002/sec

Table IV. Proposed additions to SIPA

Diameter AVP	Added to command	Purpose
OriginHost	ACR, ACA, MAR, MAA	AVP to store the current IP address of the user's device.
Retransmit-After	ACA, MAA	AVP to store the time interval after which a retransmission is allowed.
Event-Timestamp	MAR, MAA	AVP to store the time a reported event occurred.
Security-Check	ACA, MAA	AVP to inform the engaged AAA nodes about the result of a triplets check.

results presented in the next section show that SIPA+ does not severely penalize performance and resource utilization in the AAA server side.

6. EVALUATION

The SIPA system described in the previous sections has been implemented in the form of a Diameter application. We utilized OpenDiameter v. 1.0.7-I [27], which implements the base Diameter protocol and provides an API to create Diameter applications. The SIPA prototype allows us to evaluate both its compliance with the requirements given in Section 3.1 and its performance within a pilot test bed.

As already mentioned, SIPA adds support for SIP service delivery. Therefore, its evaluation in terms of performance and processing overheads was based on several scenarios representing common SIP services. Specifically, SIPA prototype has been evaluated considering the successful fulfillment of each tested scenario requirements and the correct preparation of the final invoice representing the acquired services. Moreover, extra complexity was added by forcing several handoffs to happen during a user session. Results show that SIPA does not interfere with or impedes normal AAA and SIP operations, and the accounting records were correctly and reliably created and assigned to the appropriate user.

6.1. General remarks

Session Initiation Protocol-Accounting is generic in the sense that it does not rely on specific architectures or mechanisms, and at the same time, its core functionalities are operable regardless of the underlying network technologies used. Additionally, it is highly adaptable and thus is able to support new requirements or services that may emerge in the future.

More specifically, when examining the requirements set in Section 3.1, we can say that our system adheres to the AAA-distributed nature by providing and supporting several distributed network architectures. This is actually straightforward because SIPA is built as a new Diameter application. Moreover, by using the Diameter SL, SIPA is able to dynamically locate the proper AAA server as the case may be. At the same time, SIPA extends the distributed characteristics of AAA architectures by providing additional mechanisms to support core AAA functionalities. For instance, SIP server access to an updated user profile is implemented through several mechanisms, as denoted in Section 4.2. This is performed to prevent the creation of single points of failure and allow network operators or service providers to choose the most expedient mechanism where and when applicable. For example, this is very important in the case of the authentication procedure, which can be implemented in various ways.

The demand for high level of transparency comes from the desire of both network operators and the end users to deliver/receive only one invoice that corresponds to all services obtained. This must be carried out irrespectively from which network the user acquired each service. In any case, SIPA does not affect charging issues or the way the final invoice is prepared. Thus, it produces the same output that network operators currently employ as an input for the preparation of the final invoice to be sent to the end users.

Considering SIPA security it is stressed that all guidelines behind the design of new Diameter applications were adopted. Our accounting system supports the use of IPsec for data exchanges inside the same Administrative domain and TLS in case of data crossing Administrative domains as suggested in [11]. In general, the designed prototype retains the same level of security that a real AAA system built on Diameter can provide. The actual challenge however, when building a new Diameter application, is to secure its inner functions because all communication among network elements and between different Diameter applications is considered *a priori* secure. Toward this goal, we propose using the security scheme described in Section 5.2. SIPA+ implements this mechanism as an add-on and thus offers a method to ensure the validity and accuracy of the produced accounting records. Moreover, SIPA+ facilitates the resolution of disputes between service providers and end users. This is due to the timestamping service that it provides. On top of that, as described in [26], several other attacks targeting the accounting service mainly through denial-of-service and spoofing can be combated when this mechanism is active.

In addition to the accounting process efficiency and security, another challenging issue associated especially with heterogeneous wireless networks is the privacy of the end user. Without privacy-preserving mechanisms in place, the end user can be easily tracked and profiled in the mid or long term. That is, network or service operators—especially colluding ones—may collect user information and keep them for a long time in order to profile their users and eventually sell these profiles to say advertising companies for profit. After that, the user is left defenseless to spamming and/or other related threats that violate his private sphere.

In general, privacy is a complex concept that affects aspects such as location, identification, and authentication

[28]. Whereas location privacy requires that the location of a mobile user is untraceable to unauthorized parties (including the network), identification privacy mandates users anonymity except for authorized parties. As we can see, these types of privacy are interrelated. If user's identity is private, then location data is useless. At the same time, both types of privacy strongly depend on the authentication process where user permanent identity must be exchanged. If the authentication mechanism has no adequate level of privacy to protect identification-related data, the location can be revealed to unauthorized third parties. Therefore, one of the primary targets of SIPA was to guarantee the privacy of the mobile end users receiving services. Certainly, SIPA is not responsible to offer a privacy-preserving mechanism for SIP. This can be provided by other mechanisms such as the one described in [29]. On the other hand, SIPA should guarantee that users' anonymity is assured throughout the accounting process. This is actually achieved considering the fact that no user's direct or indirect personal information (e.g., permanent identity and Master ID) leaves the possession of Root AAA server in charge inside the home domain. At the same time, it is regarded impossible for an attacker (e.g., an eavesdropper) to correlate a series of Event IDs or associate a specific Event ID with an accounting event or a user real identity. This is because of the following: (i) as already mentioned, all communications between AAA servers are IPsec/TLS protected, so having access to an event ID means getting access to the AAA itself, and (ii) the generated event IDs can be sufficiently random.

6.2. Performance evaluation

The aim of this section is to determine and evaluate the performance penalty imposed by the introduction of our accounting system inside an AAA SIP environment. SIPA extends the Diameter SIP application and is thus expected to add extra complexity and, consequently, increase resource consumption and/or time delays, which in turn may affect performance. Note that direct comparison of our findings with similar mechanisms cannot be performed as similar experimental results are not available. Instead, this section aims to determine the performance of SIPA compared with a standard AAA SIP installation. The second goal is to measure the extra cost the add-on security mechanism (i.e., SIPA+) imposes.

With providers' requirements and expectations regarding SIP services, a major priority for the designed accounting system is to keep resource utilization at an acceptable level. Factors such as real-time service delivery, complex and multiple services accessed simultaneously, and increased user population require that, apart from being robust and effective, the accounting system should keep the overheads at an affordable level. Additionally, we must assess the overall trade-off between the security add-on described in Section 5.2 and the overall SIPA performance. If this add-on imposes a high penalty in SIPA performance then may be considered a sumptuosity for most realms.

Authentication, authorization, and accounting systems comprise several time-consuming and resource-dependent procedures, whereas SIP functionality and performance is mainly affected by network delays and large roundtrips. Additionally, SIPA, besides the actual accounting functions, affects several other procedures, that is, user authentication or session termination. Thus, it is difficult to isolate and measure the direct impact of the accounting system on the AAA SIP functionality. The test bed presented in the following is therefore specially designed to test pilot scenarios that allow us to identify and assess how the accounting-related additions affect the AAA operations. Note that we are able to receive measurements from three different configurations:

- *Standard*: The default AAA SIP environment that includes the default base Diameter protocol provided by OpenDiameter and our implementation of the SIP Diameter application. As already stated, this is the first open-source implementation of the Diameter application to this point.
- *SIPA*: The SIPA system that combines our modified base Diameter protocol and our implementation of the SIP Diameter application.
- *SIPA*+: The SIPA+ system having the security add-on described in Section 5.2.

In an effort to better understand and assess the imposed overheads, we have utilized both a high-end system configuration and a low-end one. In this context, the experimental test bed comprises the following elements:

- One laptop incorporating an Intel Mobile Core 2 Duo T7500 (Intel Corp., Santa Clara, CA, USA) processor along with 2048 MB of 333 MHz RAM. This is mainly used as a high-end SIP UA.
- One laptop machine incorporating an AMD Mobile Athlon 4 CPU (Advanced Micro Devices, Sunnyvale, CA, USA) along with 256 MB of 133 MHz RAM. The CPU frequency was downgraded to 350 MHz from the original 1200 MHz using the Powersave daemon v. 0.14.0 [30]. This is performed to provide us with a low-end machine to be used as a SIP UA.
- One Intel Core 2 Duo 8200 desktop PC having 2048 MB of 666 MHz RAM and one AMD Athlon 64 ×2 3800+ desktop PC with 2048 MB of 333 MHz RAM to act as high-end SIP servers.
- One Intel Core 2 Duo 8200 desktop PC having 4096 MB of 800 MHz RAM and one Intel Core 2 Duo P8800 laptop PC having 4096 MB of 800 MHz RAM to act as high-end AAA servers.
- Two Intel Pentium III 733/800 MHz desktop PCs that incorporate 512/348 MB of 133 MHz RAM. These machines are used as low-end SIP servers.
- Two Intel Pentium III 800 MHz desktop PCs incorporating 512 MB of 133 MHz RAM. Both machines are used as low-end AAA servers.

DOI: 10.1002/sec

All previous machines utilize Ubuntu Linux v. 9.04 (Jaunty Jackalope) [31]. MySQL v. 5.0.45 was used as a local database when needed (e.g. for the AAA servers) and OpenDiameter v. 1.0.7-I was used to provide communication between network entities. The Linux library *schedutils* was used to force all processes to run on a single CPU in multiprocessor systems. This is important when measuring CPU utilization in an environment of single and multi processor systems. The Hewlett-Packard (HP, Palo Alto, CA, USA) SIPp [32] v. 3.0 was used to generate SIP traffic. It also allows for the use of XML files to customize SIP call flows and create custom scenarios. Twinkle [33] v. 1.4.2 was used as a SIP softphone. Finally, SIP Express Router [34] v. 0.9.6 provides the basic SIP server functionality.

6.2.1. Scenario I: evaluation of the authentication procedure

The first scenario examines a common mechanism to offer user authentication. This scenario message flow is depicted in Figure 6, which in turn corresponds to the architecture presented previously in Figure 1. First, a SIP UA sends a SIP REGISTER request towards SIP server 1. AAA client 1 inside SIP server 1 contacts the local AAA server through a UAR message to (i) determine if the user is allowed to receive service and (ii) become aware of available SIP servers capable of providing the requested service to the user. The AAA server responds with a UAA message that informs SIP server 1 of the existence of SIP server 2. In turn, SIP server 1 will now forward the initial SIP REGISTER request to SIP server 2.

Upon reception, SIP server 2 contacts AAA server through the MAR message and requests the initialization of the authentication procedure. The latter is usually the user's HN AAA server or a local one in case a fast-handoff scheme is employed. The AAA server responds with a MAA message that includes a challenge that will be forwarded back to the SIP UA via SIP server 1 through a standard SIP 401 (Unauthorized) message.

The next SIP REGISTER request that SIP server 1 receives contains the user credentials. SIP server 1 contacts AAA server through a new UAR message to become aware of the IP address of the entity that is currently in charge of this specific user (i.e., SIP server 2). The AAA server responds with a UAA message that includes the IP address of SIP server 2. The SIP REGISTER message is now forwarded to SIP server 2. After extracting the user credentials, the AAA client in SIP server 2 contacts the AAA server by issuing a MAR message. The AAA server authenticates the user and informs SIP server 2 of the result

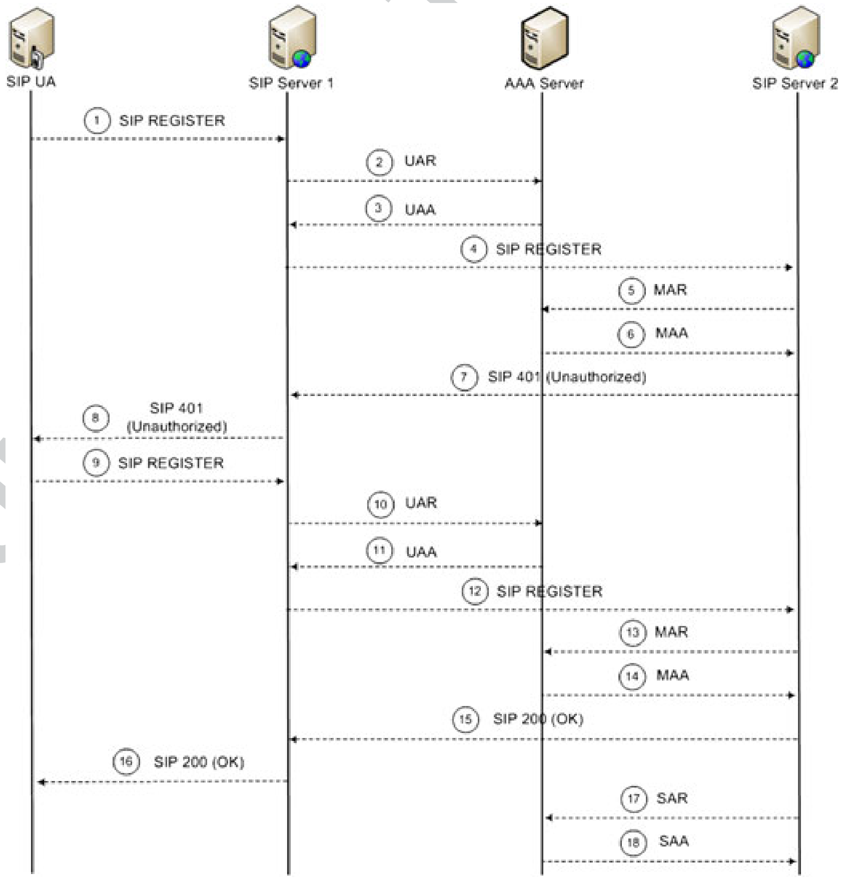


Figure 6. Scenario I message flow.

through a MAA message. SIP server 2 now constructs a SIP 200 (OK) message and forwards it to the SIP UA via SIP server 1. Finally, SIP server 2 contacts the AAA server to request delivery of the user profile information through a SAR message. The current user profile will be sent to SIP server 2 through a SAA message.

Note that this scenario does not trigger any accounting processes but the exchanged messages (i.e., MAR/MAA, SAR/SAA) convey accounting-related information. Moreover, as shown in Figure 6, this scenario does not require capabilities exchange by network elements prior to communication. The aforementioned scenario allows us to assess the overhead imposed from the augment of the Diameter commands by adding accounting-related AVPs to them. In this way, we determine if the additional cost due to the creation, exchange, and process of our more complex Diameter commands is significant.

The AAA server and SIP servers 1 and 2 reside in the same 100-Mbps LAN, whereas the communication with SIP UA is through the backbone network. The connection is realized via a 1-Mbit ADSL connection with 1024-Kbps maximum downlink and 256-Kbps maximum uplink speed. An estimate of an average ping time between SIP UA and SIP server 1 is 21.9 ms, but this value can only be considered as an indication.

The aforementioned scenario was executed 500 times, and we captured the overall time required for the completion of the process starting from the creation of the SIP REGISTER request and terminating the moment that the Diameter SAA command is received by SIP server 2. Table V shows the recorded timings for each of the three different configurations and each of the two different setups. Apart from the mean delay, we have included in the table the standard deviation of the taken measurements and the 95% confidence interval. Table VI presents the same measurements but for the CPU workload produced in the AAA server.

The results reveal that SIPA produces a minor time penalty on the authentication procedure. For instance, we witness an increment of ≈448 ms (25.7%) in the average authentication time when we use SIPA instead of the standard configuration. This extra overhead diminishes by more than 50% (≈198 ms) when moving from the low-end system configuration to the high-end one. This is not the case for SIPA+, which seems to significantly increase the imposed time penalty in the environment of the low-end system configuration. Specifically, an increment of ≈1728 ms (99.1%) in the average authentication time is witnessed when we use SIPA+ instead of the standard configuration in the case of low-end system setup. A slighter but still significant increment of about ≈1003 ms (55%) is also witnessed in the case of high-end system configuration. In addition, standard deviation of all values remains low, showing that their majority is spread near the mean delay. Specifically, taking low-end configuration as example, the standard deviation of all values for SIPA and SIPA + is ≈13.8% and 8.8% of the mean time, respectively. This observation is further supported by the calculated 95% confidence interval; for example, for the high-end system, the overall authentication time for SIPA and SIPA+ remains near 2 and 2.8 s, respectively.

Table VI. Central processing unit utilization (%) in authentication, authorization, and accounting server for scenario I.

	Low-end system		High-end system	
Configuration	Mean time	Standard deviation	Mean time	Standard deviation
Standard	14.05 ± 0.16	1.87	11.91 ± 0.09	0.98
SIPA	19.01 ± 0.18	2.01	17.12 ± 0.20	2.32
SIPA+	28.70 ± 0.18	2.08	19.90 ± 0.20	2.29

Overall, when estimating the time delay imposed by SIPA and SIPA+ it is safe to say that the total time is certainly acceptable and does not impede overall operability or performance in real-life scenarios. Keep in mind that scenario I is probably the most time-consuming part of the real-life operation circle for an AAA server and was specially used here to represent a worst-case scenario.

Regarding the imposed resource consumption, both the Diameter SIP application and SIPA implementations perform efficiently even in the case of low-end systems. On the other hand, as expected, SIPA+ seems to perform better in high-end systems and is thus regarded as CPU dependent. This is due to queue management functions and frequent data manipulation. More specifically, we witness an increment of ≈4.96 (35.3%) and ≈5.21 (43.7%) when we utilize SIPA instead of the standard configuration in the case of low-end and high-end system configurations, respectively. SIPA+ produces a penalty of ≈14.65 (104.2%) and ≈7.99 (67%), respectively. Nevertheless, the overall performance in terms of resource consumption is satisfactory, and we can argue that even relatively weaker systems operating as AAA servers would be able to cope with the new requirements imposed by SIPA and

Table V. Service time results for Scenario I (in ms).

	Low-end system		High-end system	
Configuration	Mean time	Standard deviation	Mean time	Standard deviation
Standard	1742.93 ± 18.74	213.76	1801.67 ± 17.40	198.51
SIPA	2191.35 ± 26.47	302.01	2000.05 ± 18.07	206.21
SIPA+	3471.12 ± 26.74	305.12	2805.57 ± 17.95	204.73

SIPA+. This remark is further validated by the calculated standard deviation and confidence interval values. In summary, we can argue that the amendments and additions to non-accounting functions (i.e., authentication) that the SIPA system introduces do not have an inhibiting impact on the authentication times and resource utilization.

6.2.2. Scenario II: evaluation of the accounting procedure

The second scenario aims to evaluate the actual accounting process and, most precisely, everything being triggered by the exchange of the modified or entirely new Diameter commands. Bear in mind that these are the {ACR, ACA} and {ARR, ARA} (see Table III). Figure 7 depicts the network architecture used in this scenario.

The current scenario resembles a common accounting incident involving several intra-domain and inter-domain handoffs as the user receives service from different network providers. Recall from Section 2 that, in such an environment, the SIP server and the AAA client are co-located in the same network node. The user initially registers to his home domain by connecting to AAA client 1 (SIP server 1), but sometime later, a handoff occurs, and the user is attached to a foreign network (visited domain 1) through AAA client 2 (SIP server 2). Afterwards, an intra-domain handoff occurs, and the user is attached to AAA client 3 (SIP server 3). A new handoff takes place, and the user is attached to a new foreign network (visited domain 2) through the AAA client 4 (SIP server 4). The scenario terminates when a last handoff happens and the user is again attached to his home domain through the AAA client 5 (SIP server 5). At each step, the user initializes a SIP call to another user who resides outside the current network domain. For the sake of simplicity, we only utilize predetermined call parameters, that is, call duration, call recipient, and cost.

Furthermore, AAA server 1 inside the HN takes over the role of the Root server, and AAA servers 2, 3, and 4 take over the role of Administrative servers 1, 2, and 3, respectively. AAA server 1 communicates with the AAA server 2 through the backbone network. The connection between the different network domains is realized through a 1-MB ADSL line, that is, 1024-Kbps downlink and 256-Kbps uplink maximum speeds. The average ping time between the two sub-networks is 24.7 ms, but again, this value can only be considered as an indication.

Figure 8 presents the message flow between all the network entities involved in this scenario. Note that we have omitted several messages necessary in real-life network operation (e.g., capabilities negotiation) as we focus solely on accounting. The scenario begins when SIP server 1 dispatches an ACR message and terminates the moment that Administrative server 1 receives the corresponding ARA message.

During the execution of this second scenario, we only measured the mean CPU workload produced in the AAA/SIP servers in an effort to identify the overhead caused by both SIPA and SIPA+. The scenario was also repeated 500 times. It is stressed that server time delay for this scenario is not important because real-life network operators implement complex batch accounting records transfer schemes by dynamically adjusting a time window (i.e., how often the accounting records need to be transferred between the involved domains). This is necessary in order to reduce network bandwidth usage. Generally, as discussed in [35], batch transfer of accounting data is more CPU and bandwidth efficient than real-time transfer, so providers follow this tactic except in very special cases

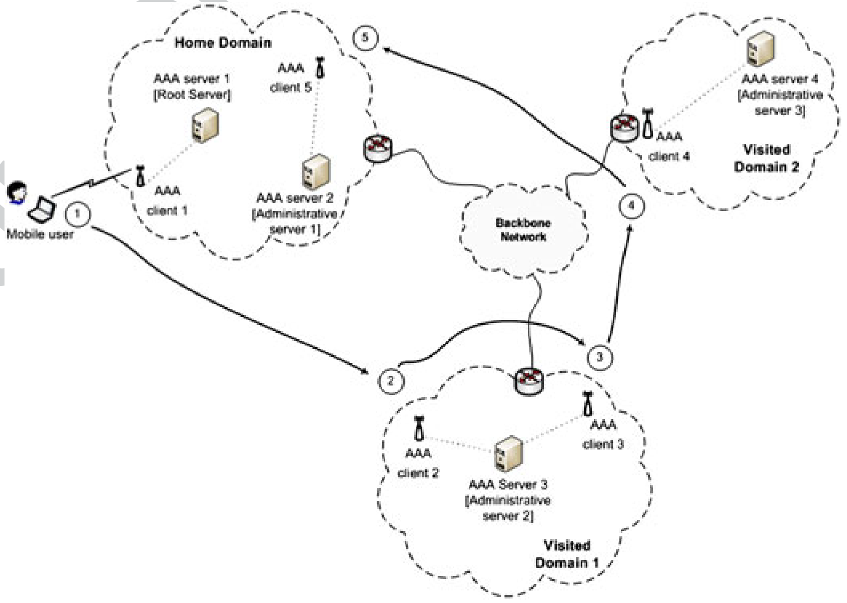


Figure 7. Scenario II network architecture.

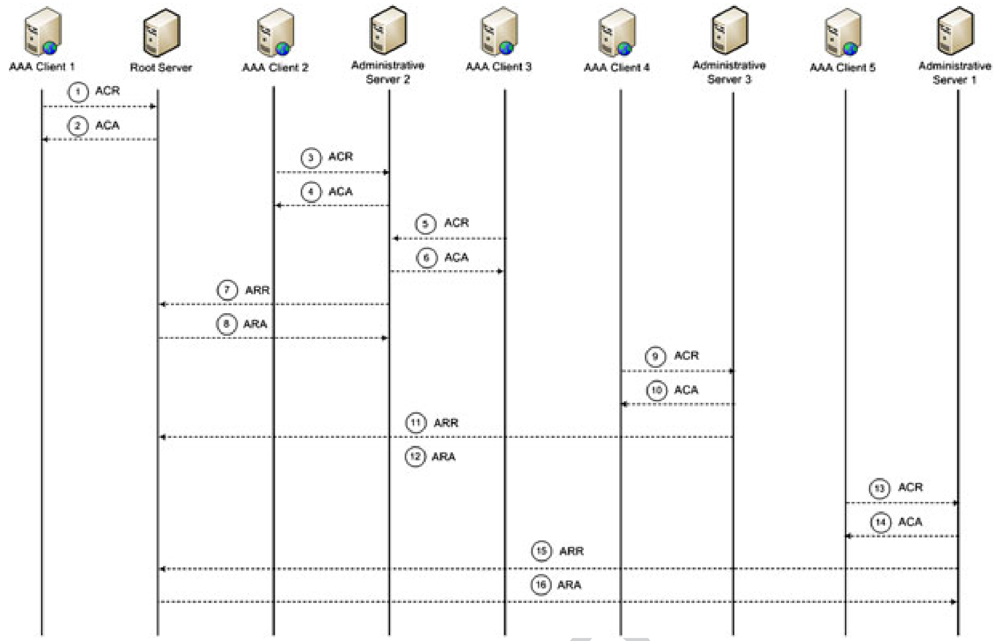


Figure 8. Scenario II message flow.

where real-time (also referred to as on-line) accounting is needed, e.g., due to the needs of credit limit checks. Real-time accounting remains out of the scope of this paper and is considered for future work. This means that SIPA *per se* does not cope with real accounting. Instead, this functionality, where needed, is delivered by standard Diameter as normal.

Figure 9 gives a comparison of the mean CPU utilization (%) between different configurations. The *X*-axis represents the hardware configuration used by each category of entities, whereas the *Y*-axis shows the mean percentage of utilization. In each point, we have also included the corresponding confidence interval as error bars. Table VII shows the standard deviation for each configuration/entity and the calculated 95% confidence interval. Note that the table does not include any findings recorded in the SIP UA side. This is because all the corresponding operations in the SIP UA are straightforward and the results infer nothing important (i.e., the focus is on accounting operations on the server side, and the client simply consumes services).

Also note that the Standard configuration does not include support for the ARR and ARA commands and is thus tested (logged) for messages 1 to 6 only. This means that this configuration is not directly comparable with SIPA variations, but in absence of any other literature results, we decided to measure it as well in order to have at least a rough comparison with SIPA.

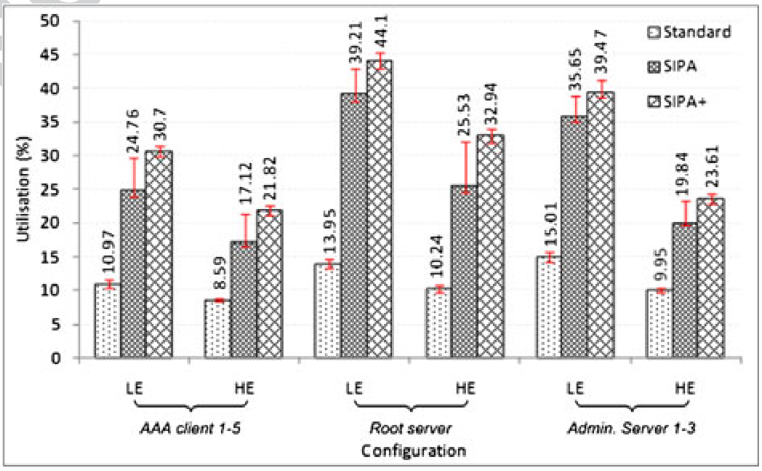


Figure 9. Comparison of mean CPU utilization (%) for scenario II.

DOI: 10.1002/sec

Table VII. Central processing unit utilization (%) for scenario II.

	Network elements					
	AAA clients 1 to 5		Root server		Administrative servers 1 to 3	
Configuration	Low end	High end	Low end	High end	Low end	High end
Standard						
Standard deviation	7.45	2.99	7.90	5.82	8.04	3.69
Confidence interval (95%)	10.97 ± 0.65	8.64 ± 0.26	13.95 ± 0.69	10.24 ± 0.51	15.01 ± 0.70	9.95 ± 0.32
SIPA						
Standard deviation	10.82	6.98	13.10	11.13	7.78	4.02
Confidence interval (95%)	24.76 ± 0.95	17.12 ± 0.61	39.21 ± 1.15	25.53 ± 0.98	35.65 ± 0.68	19.84 ± 0.35
SIPA+						
Standard deviation	8.71	7.79	13.21	12.02	8.87	7.86
Confidence interval (95%)	30.70 ± 0.76	21.82 ± 0.68	44.00 ± 1.16	32.74 ± 1.05	39.47 ± 0.78	23.51 ± 0.69

The results in Table VII reveal that, in case of high-end system configuration, the CPU utilization remains at a considerably low level. When low-end systems take the role of the SIP UA, SIP, and AAA server(s), the workload increases but not to an unacceptable level. The increment is greater in the case of AAA servers (Root and Administrative servers) and smaller for AAA clients. For instance, in the case of Standard configuration, we witness an increment of ≈2.38 (27.7%), ≈3.71 (36.2%), and ≈5.06 (50.8%) when we utilize low-end system configuration for the AAA clients, Root server, and Administrative servers, respectively.

Likewise, in the case of SIPA, the imposed penalty when shifting from high-end to low-end system configuration becomes more significant, that is, ≈7.64 (44.6%), ≈13.68 (53.5%), and ≈15.81 (79.6%) for the AAA clients, Root server, and Administrative servers, respectively. Nevertheless, this penalization diminishes by ≈12% for every entity in the case of SIPA+. In every case, however, such results prove that our proposals are viable even in low-end systems, which of course is not the case for modern service providers.

Furthermore, SIPA+ also imposes similar overhead penalties when we utilize low-end system configuration instead of high end. At this point, it is very interesting to assess the penalty imposed by the security mechanism. Considering the low-end system configuration, the utilization of SIPA+ instead of SIPA produces a penalty of ≈5.94 (23.9%), ≈4.89 (12.4%), and ≈3.82 (10.7%) for AAA clients, Root server, and Administrative servers, respectively. In the case of high-end system configuration, the respective values are ≈4.70 (27.4%), ≈7.41 (29.0%), and ≈ 3.77 (19.0%). Although these differences are significant, we can argue that these are not interdictory, especially for high-end systems, to sustain. Standard deviation of all values in this scenario is increased. Specifically, taking SIPA and Root server as example, the standard deviation for both configurations is ≈33.4% and 43.5% of the mean time, respectively. Apart from casual instabilities of the ADSL connection, this augment of standard deviation values can be explained by the fact that, every time a new experiment run initiates, Diameter initializes almost randomly several accounting parameters (queue size, intervals, etc.). The difference incurred by the random adjustment of these parameters to the measured values of each run is more noticeable in complex scenarios similar to this one. Certainly, in a real accounting environment, these parameters are fine tuned using information such as the available network bandwidth and workload; this is not possible however in controlled experiments because Diameter does not allow to manually adjust these parameters.

Generally, the overall performance in terms of resource consumption is satisfactory, and we can safely say that even relatively weak systems operating as AAA servers would be able to cope with the new requirements imposed by SIPA. Even SIPA+ that includes complex security procedures does not impose any serious penalties, and both system configurations respond well to the new requirements. In conclusion, the findings from the two scenarios prove the soundness and robustness of SIPA. Furthermore, by delivering sound accounting results, we can argue that the accounting extensions and the security add-on do not include any inherent design flaws.

7. RELATED WORK

Several studies examine the coexistence of AAA and SIP, but accounting as part of the AAA framework is usually regarded of secondary importance; thus, most works do not offer complete accounting schemes. Instead, the IETF draft directions are followed, and network operators are responsible to incorporate their proprietary accounting extensions into their AAA system. However, in most cases, these custom-built systems are not exactly compatible with current AAA procedures and protocols. In addition, every change happening in the underlying infrastructure may cause several minor or major implications to the accounting system, especially to the poorly designed ones.

Some recent studies focus on accounting systems capable of dealing with modern networks and future requirements. In [35], the authors build on top of an accounting system earlier presented in [36] and provide the necessary modifications to support accounting for session mobility in a SIP environment. More specifically, the authors discuss the required interaction between the signaling protocol and the chosen accounting system in support of session mobility in inter/intra-domain scenarios. However, the accounting system presented in [36] does not specifically focus on SIP environments. Instead, it presents a generic accounting system based on discrete accounting roles an authentication, authorization, accounting, auditing, and charging (A4C) server may have. The authors provide Diameter extensions to implement the accounting system and study handoffs inside an Administrative domain or between different Administrative domains. In case the user moves to a foreign Administrative domain, the same principles apply and the accounting management treats the new handoff as if it occurred inside the home Administrative domain.

In [37], the authors proposed an AAA and billing solution in which VSPs provide their users with tokens containing all the information that Internet access providers need for verifying their authorization rights and activating the accounting and billing procedures. A third provider, named Guarantor, signs the tokens and supports accounting and billing mediation services. SIP is chosen as the VoIP signaling protocol in use.

The work in [38] presents an accounting system based on AAA infrastructure. The authors study different accounting scenarios and elaborate on their analysis by implementing a prototype. The accounting system discussed in this work is not specifically designed for SIP; however, the authors also consider the case of a VoIP SIP-based service.

Contrariwise to the aforementioned solutions, SIPA does not require extra network elements or complex security mechanisms to exist (i.e., a public key infrastructure). At the same time, it can cope effectively with the most complex accounting scenarios. SIPA chooses to distinguish inter-domain handoffs from intra-domain ones and takes into careful consideration security and end-user privacy. Overall, as already pointed out, this is the first work to offer the following: (i) a fully fledged NGN-aware SIP accounting system, along with its performance evaluation, and (ii) an open-source implementation of the SIP Diameter application.

Last, we would like to make some clarifications regarding the relation of SIPA with IMS. IMS is built on IMS SIP as a signaling protocol, which is an enhanced version of SIP, incorporating several extensions as described in [39]; Diameter as the AAA protocol; COPS as the policy enforcing protocol, and others, including Media Gateway Control Protocol (MeGaCo or H.248), Real-time Protocol, and Real-time Control Protocol. Regarding accounting, IMS depends on the default Diameter and only adds the required functionality that enables IMS nodes to convey accounting information. More specifically, the IMS charging Diameter application reuses default Diameter messages and introduces two interfaces namely *Rf* and *Ro* [40] used for off-line and on-line charging, respectively. These interfaces enable the IMS nodes to communicate inside the IMS environment and respond to accounting queries upon reception of SIP messages. This means that accounting in the IMS architecture is performed according to the default Diameter specifications and thus retains the same weaknesses that we have pointed out in this work. On the contrary, our work enhances the default Diameter accounting functionality and thus may be used in the context of IMS. That is, because SIPA is built over SIP and IMS also utilizes an enhanced version of the same protocol, SIPA could support the requirements of IMS. In fact, our implementation of the accounting system focuses on a generic and adaptable framework capable to service IMS installations as well as less disciplined NGN IP-based proprietary configurations, for example, by supporting additional protocols and setups. It is important to note that such loose configurations are more commonly deployed in real-world scenarios at the moment.

8. CONCLUSIONS

As service providers try to continuously adapt to the multi-domain heterogeneous wireless environment and the increased competition and market challenges, they require a flexible and robust accounting system that will be able to cope with the fast-evolving market demands and enable them to accurately bill for next generation services. In this context, traditional accounting systems are proved insufficient to deal with modern services, the many-to-many relationship model between network providers, the accumulation of non-correlated data distributed across a large number of systems, and the heterogeneity of the wireless network terrain.

This paper studies the introduction of a complete and generic accounting system inside a VoIP realm. We choose SIP because it is considered the predominant protocol for NGN. In this context, we elaborate on our previous work by presenting SIPA, which incorporates all the necessary functionality to deliver proper billing for SIP services. The SIPA Diameter application comes into two flavors the standard one and SIPA+, which delivers increased security features. SIPA relies on the current AAA architecture and utilizes Diameter as the default AAA protocol and SIP as the VoIP protocol in charge. A prototype is constructed and a test bed is designed to allow us to determine the performance penalties imposed by both SIPA and SIPA+. Extensive experimentation infers that the proposed solution is sound, is robust,

DOI: 10.1002/sec

and can be easily implemented in real-life network architectures.

Future work aims to thoroughly study the security of the AAA SIP coexistence and to address open topics such as user location privacy in inter-domain handoffs and support for real-time credit-control applications. Moreover, despite that our Diameter application and extensions require a minimal extra computing time on the involved AAA servers, we will analyze the behavior of SIPA and SIPA+ under stress, that is, having a large number of clients requesting SIP services from servers.

REFERENCES

1. Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Sparks R, Handley M, Schooler E, SIP: Session Initiation Protocol. RFC 3261, June 2002.
2. Geneiatakis D, Kambourakis G, Dagiuklas T, Lambrinoudakis C, Gritzalis S. SIP security mechanisms: a state-of-the-art review, In *Proceedings of the Fifth International Network Conference (INC 2005)*. Samos: Greece, July 2005.
3. Geneiatakis D, Dagiouklas A, Kambourakis G, Lambrinoudakis C, Gritzalis S, Ehlert S, Sisalem D. Survey of security vulnerabilities in Session Initiation Protocol. *IEEE Communications Surveys and Tutorials* 2006; **8**(3): 68–81.IEEE.
4. Thermos P, Takanen A, *Securing VoIP networks: threats, vulnerabilities, and countermeasures*. Addison-Wesley Professional. Aug. 2007.
5. Camarillo G, Marshall W, Rosenberg J. Integration of Resource Management and Session Initiation Protocol (SIP). IETF RFC 3312, October 2002.
6. Wedlund E, Schulzrinne H, Mobility support using SIP, The Second ACM International Workshop on Wireless Mobile Multimedia, 76–82, Aug. 1999.
7. Vakil F et al.. *Host Mobility Management Protocol Extending SIP to 3G-IP Networks*. IETF Internet Draft: October, 1999.
8. Nakajima N, Dutta A, Das S, Schulzrinne H, Handoff delay analysis and measurement for SIP based mobility in IPv6, In *Proc. of IEEE International Conference on Communications (ICC) 2003*, Vol. **2**, 1085–1089, 2003.
9. IETF Status Pages. Authentication, Authorization and Accounting services. http://tools.ietf.org/wg/aaa/.
10. Technical Specification Group Services and System Aspects. IP Multimedia Subsystem (IMS), Stage 2, V8.6.0, TS 23.228, 3rd Generation Partnership Project, September 2008.
11. Calhoun P, Loughney J, Guttman E, Zorn G, Arkko J. Diameter Base Protocol, RFC 3588, September 2003.
12. Zhang R, Wang X, Yang X, Jiang X, Billing attacks on SIP-based VoIP systems In Proc. of First USENIX Workshop on Offensive Technologies, Aug. 2007.
13. Sisalem D, Kuthan J, Ehlert S, Sept.–Oct. 2006.Denial of service attacks targeting a SIP VoIP infrastructure: attack scenarios and prevention mechanisms, *Network, IEEE*. **20**(5): 26–31.
14. Tsakountakis A. Kambourakis G. Gritzalis SA generic accounting scheme for next generation networks. *Computer Networks* 2009; **53**(14): 2408–2426, Elsevier.
15. Diameter Network Access Server Application. http://www.ietf.org/rfc/rfc4005.txt.
16. Calhoun P, Johansson T, Perkins C, Hiller T, McCann P. Diameter Mobile IPv4 Application, RFC 4004, Aug. 2005.
17. Garcia-Martin M, Belinchon M, Pallares-Lopez M, Canales-Valenzuela C, Tammi K, Diameter Session Initiation Protocol (SIP) Application, RFC 4740, November 2006.
18. Hakala H, Mattila L, Koskinen J-P, Stura M, Loughney J, Diameter Credit-Control Application, RFC 4006, August 2005.
19. NGN definition by ITU-T. http://www.itu.int/ITUT/studygroups/com13/ngn2004/ working_definition.html.
20. Aboba B, Beadles M, Arkko J, Eronen P. The Network Access Identifier, RFC 4282, December 2005.
21. Rigney C, Willens S, Rubens A, Simpson W. Remote Authentication Dial In User Service (RADIUS), RFC 2865, June 2000.
22. Stewart R, Xie Q, Morneault K, Sharp C, Schwarzbauer H, Taylor T, Rytina I, Kalla M, Zhang L, Paxson V. Stream Control Transmission Protocol, RFC 2960, October 2000.
23. Kent S, Atkinson R. Security Architecture for the Internet Protocol, RFC 4301, December 2005.
24. Dierks T, Rescorla E. The Transport Layer Security (TLS) Protocol, Version 1.2, RFC 5246, August 2008.
25. Call Detail Records (CDR). http://www.voip-info.org/wiki/view/CDR.
26. Geneiatakis D, Kambourakis G, Lambrinoudakis C, A mechanism for ensuring the validity and accuracy of the billing services in IP telephony. Proceedings of the 5th International Conference on Trust, Privacy and Security in Digital Business, 59–68, Sept. 2008, LNCS 5185, Springer.
27. DIAMETER protocol open source implementation. http://www.opendiameter.org/.
28. Askwith B, Merabti M, Shi Q, Whiteley K. Achieving user privacy in mobile networks. *In Proc. of 13th Annual Computer Security Applications Conference, ACSAC*. IEEE Computer Society: San diego, CA, USA Dec. 1997. 108–116.

29. Karopoulos G, Kambourakis G, Gritzalis S, Konstantinou E, A framework for identity privacy in SIP. *Journal of Network and Computer Applications* 2010; **33** (1): 16–28, Elsevier.
30. Powersave Daemon. http://powersave.sourceforge.net/powersave/index.html.
31. Ubuntu Linux v. 9.04 (Jaunty Jackalope), http://www.ubuntu.com/.
32. SIPp, open source performance testing tool for SIP, http://sipp.sourceforge.net.
33. SIP softphone, open source, available at http://www.twinklephone.com.
34. SIP Express Router (SER), free, open source SIP server, http://www.iptel.org/ser.
35. Thakolsri S, Schaefer C, Walter T, Kellerer W, Accounting management for session mobility in an ubiquitous environment, In *Proc. of International Conference On Wireless Communications and Mobile Computing (ICWCMC)*, ACM Press: Vancouver, British Columbia, Canada 2006.
36. Eyermann F, Racz P, Stiller B, Schaefer C, Walter T, Diameter-based accounting management for wireless services. In Proc. of IEEE Wireless Communications and Networking Conference (WCNC 2006), Las Vegas, April 2006.
37. Polito SG, Schulzrinne H Forte A, Inter-provider AAA and billing of VoIP users with token-based method. In Proc. of Global Information Infrastructure Symposium, 2007. GIIS 2007. First International, July 2007, 159–166.
38. P Racz, B Stiller, A service model and architecture in support of IP service accounting, In Proc. of Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP, April 2006, 1–12.
39. IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP), 3GPP TS 24.229, http://www.3gpp.org/ftp/ Specs/html-info/24229.htm.
40. Telecommunication management, Charging management, Charging architecture and principles, 3GPP TS32.240, http://www.3gpp.org/ftp/Specs/html-info/32240.htm.
41. Aboba B, Arkko J, Harrington D. Introduction to Accounting Management, draft-ietf-aaa-acct-06.txt, IETF work in progress, June 2000.