Proceedings of The IEEE Region 8, EUROCON 2009, International Conference (IEEE EUROCON'09), St. Petersburg, Russia, May 18-23, 2009.

ARCHITECTURE & SYSTEM DESIGN OF AUTHENTICATION, AUTHORIZATION, & ACCOUNTING SERVICES

N. Papatheodoulou, N. Sklavos, *IEEE Member*

Abstract: **One of the most important issues in traditional and modern networks architecture is security. Data integrity and authenticity are the most critical points that a network security model should protect and ensure. Authentication, Authorization and Accounting model (AAA Protocol) is one of the most portable security concepts. Authentication acts providing proof of authenticity for stored data and verifying proof of authenticity for received. Authorization acts providing privileges to those clients that present specific credentials. Accounting acts collecting accounting metrics for two reasons; to forward them to the billing server for billing results, and to keep them saved locally for the procedure of trend analysis. In this paper, the generic AAA architecture is introduced, personalized and practically designed for usage in modern networks. The most efficient way, using supported protocols and cryptographic algorithms, for administrating AAA in practice, is proposed for mobile networks or administrative domains. In this work, a web based application scenario using the AAA protocol is proposed, with the server-side developed in PHP, SQL and Java, implementation platform.**

Index Terms: **AAA, Security, Networks.**

I. INTRODUCTION

In both traditional and modern networks, security is one of the most critical issues. Many people have the appropriate knowledge to steal or destroy data, in the middle of telecommunication cables. From point to point we can find these entities called MITM (Men In The Middle). Networks security should be built in the best way to protect data interactions [1-5].

AAA Protocol [6] is one of the most portable and trustful protocols for data integrity and security [5]. Authentication consists of two acts. The act of providing proof of authenticity for the information that is being delivered or stored, and the act of verifying the proof of authenticity for the information that is being received or retrieved [5]. Authorization is defined as the act of determining whether a particular privilege can be granted to the presenter of a particular credential. That privilege can be right of access to a resource, such as a communications' link, an information data base, a computing machine or many other network resources. Accounting is something different than billing. Accounting involves more than tracking a user's total number of data packets for billing.

In this paper, the generic AAA architecture is introduced, personalized and practically designed for usage in modern networks. The most efficient way, using supported protocols and cryptographic algorithms, for administrating AAA in practice, is proposed for mobile networks or administrative domains. In this work, a web based application scenario using the AAA protocol is proposed, with the server-side developed in PHP, SQL and Java, implementation platform.

II. AAA PROTOCOL

AAA is a network security model for mobile access, based on RADIUS [5], [7] and EAP [5], [8]. This model is also called "3A", standing for Authentication, Authorization and Accounting [5-6]. In the first phase (Authentication) client interacts with Authenticator (Authentication Server), producing some credentials of trust, in order to gain access to the network. In the second phase (Authorization) [9] the Authorization Server checks the client's authority level to access the network services. In the last phase (Accounting), Accounting Server collects a set of accounting metrics, which will be forwarded to the Billing Server, and saved in the Accounting Server for Trend Analysis [5-6].

This is the phase where an EAP conversation is produced [5],[6] between Client and Authenticator. Authentication process can be devised in two basic categories: 2-party authentication and 3-party authentication. In 2-party authentication, a client interacts directly with the authenticator, making an EAP conversation. In 3-party authentication a Network Access Server (NAS) is connected between the Client and the Authentication Server. In small networks, an authentication server could be configured by the system administrator at the authenticator, so the authenticator and the authentication server are co-located. However in large networks or administrative domains, this is not practical. In that scenario, many network points of presence, acting as NAS are deployed and the authentication must be done according to the three-party model [5],[10].

Client presents a set of credentials (as username – password), ciphered using hash functions such as [8-9] (MD5, SHA), in order to gain access to the network. A multiplexer encapsulates the credentials in a 128-bit length (or 256-bit for SHA-2) cipher-text and passes it through the network to the authenticator. Inside the Authentication Server, the credentials are being checked for authenticity, using the same hash functions. Afterwards, are being saved in a database (e.g. SQL) for a specific period of time (until the end-time of the connection). In Figure 1, the conversation between Client and Authenticator is shown.

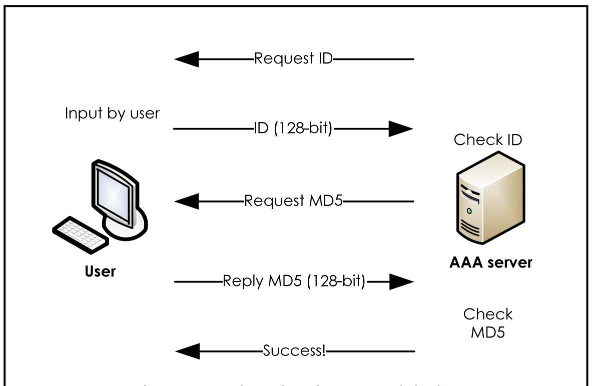

Fig.1 – Authentication Handshake.

In a 3-party authentication model, there would be the NAS (as an AAA Client) between user and AAA server [5].

Authorization is the procedure, in which the Authorization Server checks the authority level of each client, to use the network provided services [11]. In this phase the Client interacts with Authorization Server. When that interaction is finished the authorized access, for each level of the network provided services begins for each case. In Figure 2, the Authorization Process is shown.

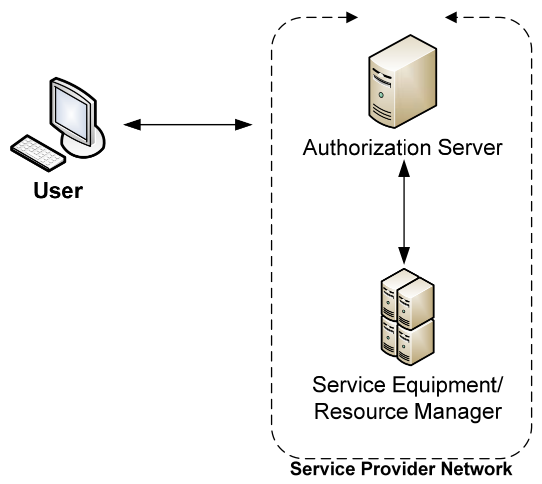

Fig.2 – Authorization Process.

There are three different sequences, which represent three alternatives scenarios. Agent Sequence, Pull Sequence and Push Sequence. In the first sequence, the transaction entities are client, Network Access Server (NAS), Authorization Server and Resource Manager. Client sends an authorization request to NAS. Secondly, NAS forwards the request to Authorization Server. Then, the Authorization server passes the request to resource manager. After the configuration procedure is completed, the authorization reply is sent backwards. In Pull sequence, there is a direct transaction between user and service equipment. Lastly, in Push sequence, an authorization certificate is provided by the AAA server to the user. Every time, this certificate is presented by user, automatically is gained an authorization reply [11-13].

In this phase of AAA, the Accounting Server collects the necessary accounting metrics (usage information of network resources) in order to forward them to the billing server. Also, some of accounting metrics are saved locally in a data base running on Accounting Server for future use. This process is called “Trend Analysis”, [5].

Accounting Protocols should be able to have some requirements [7], [14] such as secrecy of accounting policies, accounting data, and protection of integrity of accounting policies/records. Accounting policies and records must be protected from unauthorized entities ability, of reading or modifying. Also, it must be ensured, that there is no modification in the way from sender to the receiver. For this statement data authentication can also be protected by digital signatures [5].

Client sends its authentication credentials to the Network Access Server. Secondly, NAS forwards the credentials to the Authentication Server in order to gain access to the network services. Authenticator checks the credentials to allow or deny the access. When the authentication process is finished, the conversation between authenticator and Authorization server starts, in order for the authority level of the client to be checked. Lastly, and when the authority level is known, client is ready to start using the network resources, always interacting with the accounting server, which collects metrics for Trend Analysis. Afterwards, Accounting Server passes the accounting metrics to the billing server, in order to be used in billing processes [7].

In traditional network security models, the basic functions to ensure the data transport is privacy. Data that can be read and understood without any special transformations is called plain text or clear text. The method of changing plaintext, using specific algorithms, in such a way as to hide its substance is called encryption. Encrypting plaintext results in unreadable condition called ciphertext. We could use encryption to ensure that information is hidden from anyone for whom it is not intended such us MITM (Men-In-The-Middle), even those who can see the encrypted data. The process of reverting cipher text to its original plaintext is called decryption [5].

III. PROPOSED SYSTEM ARCHITECTURE

In the proposed system design, the authentication architecture is based on EAP (Extended Authentication Protocol) and also uses MD5 hash function and SQL for building data bases saving credentials to be used in future transactions between server and client. EAP is one of the most famous and useful protocols, which can be used for authentication in mobile networks. This protocol is not only used in AAA [8], [10], but in a number of network protocols and applications today.

In the part of cryptography, as one of the most important issues in networks’ architecture, MD5 is selected as the defined hash function. MD5 is portable and safe for that kind of interactions, on the phase of authentication. It takes as input one variable length of characters, and when it runs, produces a symbol-row of 128-bit output [8]. The basic issue of this hash function is that each input has a unique output, and there is no way using the output to find the input, after any reverse process.

In the proposed system, authentication credentials are two entities; username and password. Using MD5 [17], authentication credentials can be encrypted and ensured for their transportation in unsafe network connections. Afterwards a multiplexer will encapsulate username and password in a single set of 128-bit. That set of credential will be transported from client side to AAA server, as it is presented in the next Figure 3.

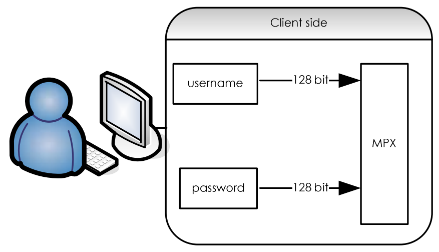

Fig.3 – Client Side Authentication.

When AAA server receives the authentication credentials authentication procedure begins. In this phase, authenticator checks the authenticity of credentials using the same hash function and if calculates the same result, saves the credentials to the SQL database and replies a “success” message to the client. AAA server side architecture is presented on Figure 4.

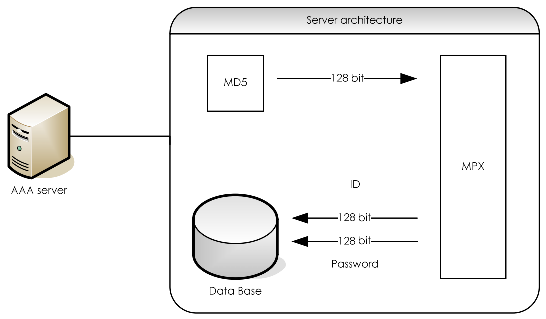

Fig.4 – Server Side Authentication.

As we referred in a previous section, NAS forwards the requests and replies between client and AAA server. In the part of authorization, the agent sequence is provided, where is used a Network Access Server (AAA client) as an agent of user. In this sequence of authorization messaging, user sends its service request, which can be seen as authorization request, towards the AAA server.

First of all, user sends the authorization request and a personal certificate, provided by AAA server, to NAS which, forwards the certificate to the AAA server. In the AAA server side, the authorization procedure begins. AAA server checks the certificate and produces a result for each case (how many and what kind of network resources user is authorized to access). Certificates are also saved locally for a little period of time.

Then AAA server, (Figure 5), forwards the authorization result to resource manager, where states are set up for providing the network services. When the configuration procedure is completed, AAA server provides the authorized access to network resources and equipment, which are necessary for services. Intra-domain accounting events are typically routed to the local billing server, while inter-domain accounting events will be routed to accounting servers, operating within other administrative domains.

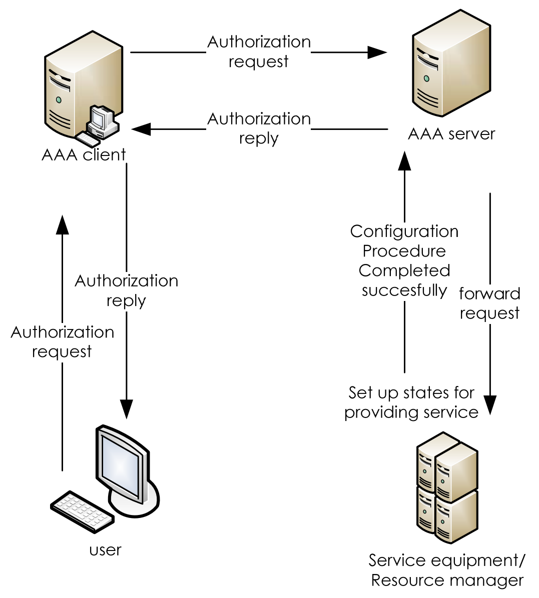

Fig.5 – Agent Sequence Authorization Model.

While it is not required that session record formats are used in inter- and intra- domain accounting as the same. This is desirable, since it eliminates translations that would otherwise be required. Where a proxy forwarder is employed, domain-based access controls may be employed by the proxy forwarder, rather than by the devices themselves. The network device will typically speak an accounting protocol to the proxy forwarder, which may then either convert the accounting packets to session records, or forward the accounting packets to another domain. In either case, domain separation is typically achieved by having the proxy forwarder sort the session records or accounting messages by destination.

Where the accounting proxy is not trusted, it may be difficult to verify that the proxy is issuing correct session records based on the accounting messages it receives. This is due to the fact that the original accounting messages typically are not forwarded along with the session records.

Therefore where trust is an issue, the proxy typically forwards the accounting packets themselves. Assuming that the accounting protocol supports data object security, this allows the end-points to verify that the proxy has not modified the data in transit or snooped on the packet contents [4].

As it is shown in the following Figure 6, accounting procedure is a part of two sub procedures. Thus of Accounting and Trend Analysis. Accounting is the act of collecting accounting metrics for further use. Trend Analysis is the act of collecting statistics for the recently used resources by a client. In trend analysis the accounting metrics are saved back in a database, in order for the system to make a forecast for the trends. That could be a great point for a better use of the network, in order to avoid unnecessary traffic.

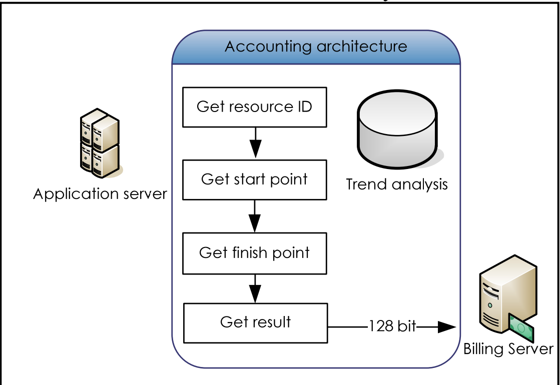

Fig.6 – Accounting Model.

IV. SYSTEM APPLICATION SCENARIO

In this section, we will present our scenario based on a web application using programming languages such as XHTML, PHP and Java, connected with a web data base (MySQL). That stands for a practice on what we referred previously. For the GUI environment we used a content management system. In this case we selected Joomla (Open Source).

First of all, when a user tries to gain access in the portal, after entering the parent domain name he will be redirected in a login interface. After entering the credentials, the user will be able to view the authenticated user's control panel, where he has three specific choices. The first is a link with the home page of the control panel, the second is a link with the main page, where he can browse the content of the web-site and the third is a log-out button, which finishes the login session. As we referred previously, all the credentials are safely protected using MD5 hash function [10].

By following a link labeled "AAA in practice", the user is forwarded to the main content page and here the authentication process is finished successfully. The application interface for authenticated users is presented in Figure 7.

In left side panel there are three menus. The first is the main menu where we could find the basic links, such us "home page", "contact us" and "search". The second menu has only one link labeled "administrator", from where the administrators can gain access, redirecting in a high level authorization check (in this case the CMS authorization form). The third menu is the main login form, where the main different level authorized members, can gain access in each case.

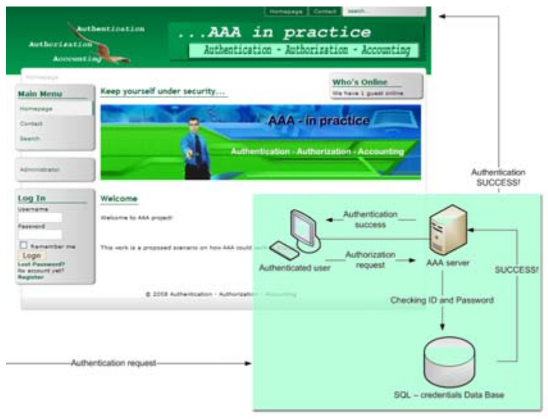

Fig.7 – Authentication Interface.

There are also two more choices; one for the "forgot password" statement and one for the registration main panel. When an authorized user has be checked by the Authorization Server, in the main page will be appeared one more menu in the left-side panel. There would be placed more options. Those links are depended on each authorization level, where low level users will have specific permissions (or services provided) and high level users will have more advantages (or more services).

The new choices for authorized users are different for each level of authorization. The categorization of authority levels are specified in this proposed system application scenario, as registered users, authors, editors, publishers, administrators, super administrators. Registered users can only read (or watch) more articles (or videos in each case of service). Editors can subscribe an article or a link (or another provided services), but it will be shown "live", only if the publisher accepts the publish request of the editor. Publishers are authorized to publish documents, or to accept publish requests. Administrators can modify the site representation, but has not access to the global configuration and super administrator can globally configure the whole application.

The options that could be selected are "Elements", "Add News", "Add Links", "Log Out" and "Administrator". In the section "Elements" a user can see his personal set of credentials as name, username, password etc. The other three following menus act as they are called (to add news or links and to end the login session). The last menu element is shown only to the administrators or super administrators, and redirects to the authorization check for the protected area of administrative tools control panel.

The accounting process could be made in three several ways for each case. If the produced service was audio or video streaming, there would be some counters used to report the start time and the finish time for each session. Otherwise, if the produced service was the usage of some network resources, accounting could be done by getting the IP of each resource and the duration of usage (here would also be the process of trend analysis) . Finally, if the produced service was downloading files from a file server, accounting would take place by counting the number of files or the size of them. Afterwards all data would make a report to the billing server, in order to start each billing process.

In the proposed system application scenario, the accounting is made by counting the session's duration for each user and forwarding the results to the billing server. That is being using Java Script, on the client side, forwarding the accounting metrics to the Accounting server. This function could also be produced in server side, using PHP.

The accounting process acts collecting a counter value, and afterwards saving it in a hidden field. After the end of each session (e.g. by leaving the authorized area, or losing the server connection.) the updated value will be saved in a SQL database in accounting server. The value will be forwarded lastly to the billing server, and saved for trend analysis.

V. PROPOSED SYSTEM: AUTHENTICATION SCHEME

In our scenario authentication credentials are 2 entities (Figure 8); username and password. Using MD5 [17], authentication credentials can be encrypted and ensured for their transportation in unsafe network connections. Afterwards a multiplexer will encapsulate username and password in a single set of 128-bit. The authentication procedure can be produced via two specified methods. The first method is client-based, which is not completely secured and for this reason is not recommended. This method can be applied using Java and an authN component which is used for the authentication comparisons. It is not secured because the used component could not be encrypted. The second one is our proposed method, which is based on the client-server model. In this proposed scenario client (user or device) interacts with the Network Access Server (NAS – an entity of AAA architecture which acts as an agent between client and AAA server) and then NAS forwards the client requests to the server. Firstly, the client presents a set of credentials to the NAS. The set of credentials for each client is different and hardly encrypted, using the MD5 hash function. When the NAS receives the set, forwards the authentication request to the AAA server. The AAA server checks the authenticity of credentials for each client, using PHP scripts, which communicate with the SQL database where the usernames and passwords are stored. If the authentication procedure has a result of success, AAA server sends a reply to the NAS, and the client gains access to the network resources for a specific time limit (accordingly to the time limit of each session).

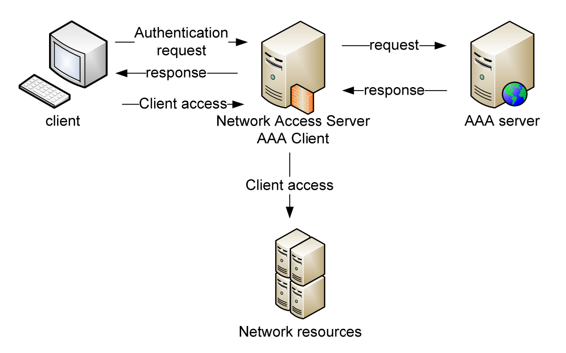

Fig.8 – Authentication Scheme.

VI. PROPOSED SYSTEM: AUTHORIZATION SCHEME

The authorization procedure (Figure 9) is totally connected to the authentication procedure. That means that when a client gets registered, the system collects the data of each authority level. The privileges of each client are stored in a SQL database and there is maden the categorization of users.

After the authentication procedure is succeeding, authentication server sends a request to the authorization server, in order to check the authorization level of the specific client.

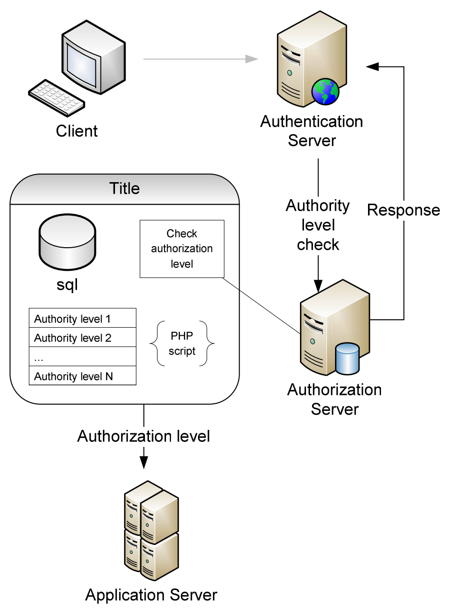

Fig.9 – Authorization Scheme.

A PHP script is running and checks the authority level for the specific client (the owner of the authentication credentials) using an SQL database, where the authority groups and each users of them, are stored in separate columns. After the end of the authorization procedure, the results are forwarded to the Application Server (resource manager) in order to know in which level will transact with each user and which resources will make available for it.

VII. PROPOSED SYSTEM: ACCOUNTING SCHEME

The accounting process (Figure 10) could be made in three several ways for each case. If the produced service was audio or video streaming, there would be some counters used to report the start time and the finish time for each session. Otherwise, if the produced service was the usage of some network resources, accounting could be done by getting the IP of each resource and the duration of usage (here would also be the process of trend analysis) . Finally, if the produced service was downloading files from a file server, accounting would take place by counting the number of files or the size of them. Afterwards all data would make a report to the billing server, in order to start each billing process.

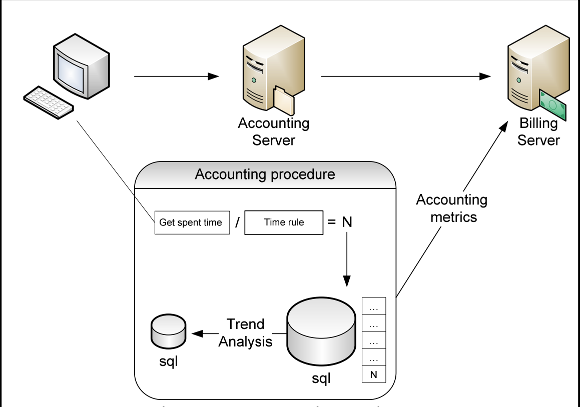

Fig.10 – Accounting Scheme.

In the proposed system application scenario, the accounting is made by counting the session's duration for each user and forwarding the results to the billing server. An "onload" Java script counts the time of each session. When the authorized page is loaded, the counter starts and a read only field appears on page. The authorized user can be informed each 30 seconds for the current session duration. On unload, an alert message informs the user for the counted duration in minutes and a PHP script saves the accounting metric to the accounting server in a SQL data base. Each metric should be saved in a new field of the database, and not as a result of an addition with the previous session results, for a more specific Trend Analysis process to be done. After the user closes the web browser, or finishes the browsing of authorized session, the client side counter stops and starts the trend analysis in AAA server side.

VIII. IMPLEMEMTATION PLATFORM

The implementation platform used for the proposed system application is a mix of PHP, SQL, HTML and Java Script.

PHP is a powerful server-side scripting language for creating dynamic and interactive websites. PHP is the widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. PHP is perfectly suited for Web development and can be embedded directly into the HTML code. The PHP syntax is very similar to Perl and C. PHP is often used together with Apache (web server) on various operating systems. It also supports ISAPI and can be used with Microsoft's IIS on Windows [10].

JavaScript is used in millions of web applications to add functionality, validate forms, detect browsers, and much more. In this case JavaScript used for the accounting process [17].

SQL stands for Structured Query Language. SQL is a standard language for accessing and manipulating databases [16].

HTML is the programming language used to create documents for display on the World Wide Web. HTML means HyperText Markup Language, which is the predominant markup language for Web pages. It provides a means to describe the structure of text-based information in a document — by denoting certain text as links, headings, paragraphs, lists, and so on — and to supplement that text with interactive forms, embedded images, and other objects. HTML is written in the form of tags, surrounded by angle brackets. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code (such as JavaScript) which can affect the behavior of Web browsers and other HTML processors [18].

IX. COCLUSIONS & OUTLOOK

AAA is the most portable protocol in network security for mobile access. The set of credentials, encapsulated in 128-bit packets, is trustful in any transaction between the network resources. The choice of MD5 as hash function of the authentication procedure, ensures security of a good level strength (using 128-bit message digest). Authentication, Authorization and Accounting as security schemes in a network architecture, should interact properly each other, in order to provide security of high level strength. In this work, a web based application scenario using the AAA protocol is proposed, with the server-side developed in PHP, SQL and Java, implementation platform.

REFERENCES

[1]. P. Kitsos, N. Sklavos, O. Koufopavlou, "UMTS Security: System Architecture and Hardware Implementation", Wireless Communications and Mobile Computing Journal, John Wiley Editions, Vol. 7, Issue: 4, pp. 483-494, 2007.

[2]. N. Sklavos, X. Zhang, *Wireless Security & Cryptography: Specifications and Implementations*, CRC-Press, A Taylor and Francis Group, ISBN: 084938771X, 2007.

[3]. N. Li, D. Lee, "Virtual Authentication Ring for Securing Network Operations", proceedings of NTMS' 2007 Conference, 2007.

[4]. B. Aboba, J. Arkko, D. Harrington, "Introduction to Accounting Management", RFC 2975, IETF, October 2000.

[5]. Madjid Nakhjiri, Mahsa Nakhjiri, *AAA and Network Security for Mobile Access – Radius, Diameter, EAP, PKI and IP Mobility,* John Wiley Editions, 2005.

[6]. L. Blunk, J. Vollbrecht, *PPP Extensible Authentication Protocol,* RFC 2284, IETF, March 1998.

[7]. B. Aboba, D. Simons, *PPP EAP TLS Authentication Protocol,* RFC 2716, IETF, October 1999.

[8]. R. Rivest, *The MD5 Message Digest Algorithm,* RFC 1321, IETF, April 1992.

[9]. NIST, *Secure HASH Standards,* FIPS PUB 180-1, April 1995.

[10]. C. Laat, *Generic AAA Architecture,* RFC 2903, IETF, August 2000.

[11]. J. Vollbrecht, *AAA Authorization Framework,* RFC 2904, IETF, August 2000.

[12]. J. Vollbrecht, *AAA Authorization Application Examples,* RFC 2905, August 2000.

[13]. S. Farrell, *AAA Authorization Requirements,* RFC 2906, IETF, August 2000.

[14]. B. Aboba, *Introduction to Accounting Management,* RFC 2975, IETF, October 2000.

[15]. P. Calhoun, *RADIUS Accounting Interim Accounting Record Extension,* Draft, IETF, January 1998.

[16]. T. Zseby, *Policy Based Accounting,* RFC 3334, IETF, October 2002.

[17]. J. Deepakumara, H.M. Heys, R. Venkatesan, *FPGA Implementation of MD5 Hash Algorithm*, in Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2001), Toronto, Ontario, May 2001.

[18]. L. Welling, L. Thomson, *PHP and MySQL Web Development*, Third edition,Sams Publishing, 2005.

[19]. T. Berners Lee, D. Connolly, *Hypertext Markup Language – 2.0*, RFC 1866, IETF, November 1995.

[20]. Meta Group, *Java Standards, The Mandate for Interoperability and Compatibility*, February 2005, Meta Group.

Nikolaos Papatheodoulou is a post graduate student with the Informatics & MM Department, Technological Educational Institute of Patras, Greece. His research interests include security, wireless networks, and computers architecture. Contact him at: npapatheodoulou@yahoo.gr.

Dr. Nicolas Sklavos received the Ph.D. Degree in Electrical & Computer Engineering, and the Diploma in Electrical & Computer Engineering, in 2004 and in 2000 respectively, both from the Electrical & Computer Engineering Dept., University of Patras, Greece. Currently he is an Assistant Professor in Informatics & MM Dept, Technological Educational Institute of Patras, Greece. He is also adjunct faculty in the grade of Assistant Professor in Computer Engineering & Informatics Dept., University of Patras, Greece. His research interests include System on Chip Design, Computers Architecture, VLSI Design, Security of Computers and Networks. He holds an award for his PhD thesis on “VLSI Designs of Wireless Communications Security Systems”, from IFIP VLSI SOC 2003. He was the General Co-Chair of MobiMedia 2007, (ACM). He serves as an Associate Editor for both Computers & Electrical Engineering Journal, Elsevier Press, and Information Security Journal: A Global Perspective, Taylor & Francis Group. He has been the Guest Editor of Special Issues for Elsevier & Springer-Verlag publishers. He has participated to the organization of up to 50 conferences by IEEE, ACM, IFIP, as Publicity, Publication Chair, Program and Technical Committee member. He is the Council’s Chair of IEEE Greece GOLD Affinity Group. N. Sklavos is a member of the IEEE, the Technical Chamber of Greece, and the Greek Electrical Engineering Society. He has authored or co-authored up to 100 scientific articles, books chapters, tutorials and reports, in the areas of his research. Contact him at: nsklavos@ieee.org.