Server-Based Access Verification

Francesco Bergadano[2], Davide Cavagnino[2], Pasquale Andrea Nesta[2]
[2]Dipartimento di Informatica – Università degli Studi di Torino
Corso Svizzera 185, 10149 Torino, Italy

Keywords. Log file, certification, web server, software module.

Abstract. In many practical contexts, it is necessary to certify that the content of a web server log file is correct with respect to real client traffic. This certification should be carried out by an independent third party, which we will call a certification agency, that is trusted by the web server owner and by the log file user. The certification agency should use adequate technology to perform the requested certification. The used technology must ensure that the log file was not altered or, in case of modifications to the log file, it should detect individual items that were added or removed. In this paper a novel technique for web server access certification based on software is presented, and its reliability and performance is discussed. A case study and experimental data from a web site with significant traffic is also presented.

1 Introduction

Web site usage data is stored in a log file recorded by the web server. This log file may be modified, but there are contexts where access data should be truthful. For example, some web sites must show to other parties how many visits they have in a certain period of time: this may be useful in case of selling advertisement space on a web site. The necessity of proving the truthfulness of the log file means that all the web requests recorded in the log file must be proven real. The technology[1] we present in this paper may be used to verify that in a log line the machine corresponding to the IP address really requested the corresponding URL (that is, IP addresses are not modified, nor is the resource requested, and no faked requests are added to the log file). To be impartial and reliable, this certification should not be done by the web site owner. A third party certification is needed from an entity that is not involved neither with the web site owner nor with the user of the certified data.

Log file modification can be performed in many ways. For example, a log file editor may be used to add/delete/modify lines in the log file, and this modification may be performed manually or using an automatic tool. Another way to create a log file with false entries (i.e. entries not corresponding to real client requests) is to use a machine located on the web server's LAN that makes requests to the web server using false IP addresses (i.e. addresses not belonging to the machine) asking for resources on the web server. This technique is called IP spoofing and may cause the web server

[1] Patented. Italian patent no. 1320716

Cavagnino D., Bergadano F. and Andrea Nesta P. (2004).
Server-Based Access Verification.
In *Proceedings of the 1st International Workshop on Electronic Government and Commerce: Design, Modeling, Analysis and Security*, pages 91-103
DOI: 10.5220/0001402400910103

to log requests as if they were coming from the spoofed machine (the machine owning the IP address used by the spoofing machine) [4][7]. There are cases in which spoofing may work; below we present a discussion on this issue. In general, with IP spoofing a machine sends IP packets containing a source address different from the IP address belonging to the machine. In the web server case, a client may perform an HTTP [11][12] request using a false IP address. HTTP uses TCP to create a connection to the destination machine (in our case, the web server). TCP performs an handshake to make the connection, and this handshake prevents the creation of the connection because the network (i.e. Internet) routes the web server response packets towards the spoofed machine, which does not recognize them and may simply discard them. Thus, the spoofing machine is not able to maintain the TCP connection needed by HTTP, and no line is recorded in the log file. An exception to this is when the spoofing machine is able to intercept the network traffic from the web server; for example, if the spoofing machine is on the same web server's Local Area Network, then it is able to see the content of the response, and thus answer correctly to the TCP handshake (obviously with a software aimed to this purpose). Performing a correct handshake, the spoofer is able to send the HTTP request to the unaware web server, that will log the request as correct, real, and as if it was coming from the spoofed IP address.

In any case, log files should not be altered, and in this paper we present a method for certifying the content of a log file.

From the previous considerations, in some situations it is possible to see the necessity of an independent third party that certifies that the content of a web server log file is truthful: we call this entity a *certification agency*. Moreover, the certification agency should use appropriate means to perform the activity of certifying a log file content.

The technology we present can be used to control the operations of a web server and to certify the content of its log file. The system we developed allows to check the IP of a machine that connects to a web server, and also allows to verify that the logged requested resource is the one really requested. We implemented this procedure in a software module we called Certimeter Express (see [8]). This module can be added to an Apache web server [2], but the technology we developed can be added to other web servers as well. The main requirement is that the web server structure should allow for adding a new component for the processing of web requests (e.g. modules, ISAPI filters).

Our technology is strictly related to the measurement of the popularity of a web site. In fact, the certification of the log file has as a consequence that all the statistics derived from it are also certified, giving a measure of the number of visits and visitors a web site has received in a period of time. Many methods may be used to measure web sites' popularity. In section 2 we give an overview of other technologies and methods. These technologies may be divided into two families, census-based and panel-based. Census based methods perform the analysis of web site accesses by directly controlling the web server, adding for example a hardware device to the web server, or connecting a device to the web server's LAN, or, like our system, adding a software module to the web server. Panel-based methods perform some statistical analysis using a panel of users that answer questions about the web sites they visited, or use modified browsers that collect visit information, recording meaningful data during the user browsing. Both methods are useful because they can collect different kinds of data that may be used in conjunction to get general information on web usage. Section 3 and section 4 present the system we developed. Section 5 discusses a

running experimentation of the system, and section 6 presents the future developments we are planning. In the final section we draw some conclusions.

2 Related Work

The problem of counting and verifying the accesses to a web site has been addressed in the literature with various approaches.

Some methods are based on a work on secret sharing [18], in which a secret is distributed into N parts having the property that only K<N parts are sufficient to reconstruct the original secret. [15] presents a solution to the problem of counting the clients; this method distributes parts of a secret to N clients. When a client visits a web site that is controlled, then it also sends to the web site its part of the secret. When a web server collects K parts then it is able to reconstruct a secret that proves it has got K visits. [16] is a patent application on secure accounting and auditing.

[6][13][14] present variations and improvements to the method of secret sharing, introducing the concepts of multilevel thresholds and of pricing. One feature of these solutions is the possibility to have different proofs of visits, depending on the number of these visits.

[9] and the patent [10] describe a method for counting unique client visitors and number of visits. It is based on sending a computation to perform to the client that connected to the server; the result of this computation is collected by the server as a proof of a client visit. The computation is chosen to be time consuming, making it difficult for the server's owner to use computing time to create false proofs of visits.

[3] presents a methodology for certifying in an efficient way a web server log file, with the use of a software module and a tamper evident hardware box. The software module is used to modify the behavior of the web server, that for every request asks to the hardware box if the client needs to be checked (that is, the request being logged is verified). In that context, the tamper evident hardware box is used to avoid the possibility of controlling and foreseeing which request will be verified. Using the box, it is possible to control a very small number of requests, making the system efficient and secure. But the disadvantage of that method is that a hardware box must be installed on the web server's LAN. The technology we present in this paper does not use any hardware module, making the method lighter to install and to configure on the controlled web site.

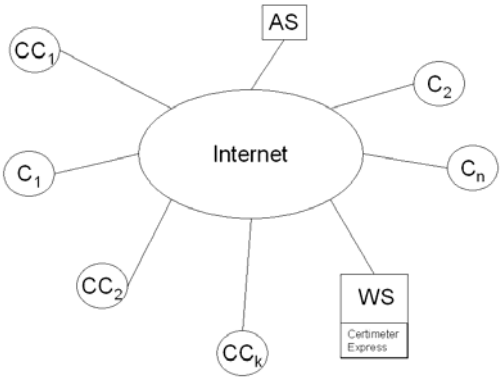

Fig. 1. The components of the system in the network context.

Another approach is called page tagging. Hit counting is performed by inserting an object (like an image) on all the pages that are to be counted. This object is contained on a server operated by the certification agency and performing as an auditing entity; thus every time a client downloads a page from the site that needs a log file certification, then it also requests the object to the server of the certification agency, allowing for an independent hit count.

The academic literature has also dealt with a problem similar to the one of counting web accesses, namely the counting of click-throughs. Click-through means accessing a web page clicking on a link contained into another web page (the referring web page). There may be situations (see for example [1]) in which it is important to resolve with a secure method possible disputes between two sites, the referring and the referred, on the number of accesses and clicks they independently counted. [17] presents such a method.

3 The Architecture

In this section we present the architecture of the solution we propose, named Certimeter Express, introducing all the components that interact through the network, during a request to the web server that needs a certification of the log file. The interaction between all these components will be clarified in the next section.

In Figure 1 all the components involved are shown, and we give below an explanation for each one:

- WS: is the controlled web server, i.e. the web server that needs a certification of its log file. This server contains the specific software module we developed, Certimeter Express. The addition of this component forces every request to be treated in a particular manner, allowing for a possible verification of the client request.

- AS: is the audit server, operated by the certification agency. It is a web server (that uses the HTTP protocol) located on a network owned by the certification agency. Its job is to verify that the client IP made a request to WS.
- C_w: a generic web client that supports the HTTP protocol [11][12]. All the clients that access the controlled web server WS are verified with the technique explained in the following section.
- CC_i: is part of a set of controlling clients; these clients are operated by the certification agency, and from the point of view of WS they are undistinguishable from any C_w. The aim of the use of these clients will be discussed in the following section, where all the data exchanges between the system components will be explained, showing how the desired certification may be achieved.

4 How the system works

The objective of the certification is to verify that all the lines of the web server log file that contain an IP and a requested URI are correct. This means that the certification procedure should guarantee that the machine having the logged IP really requested the corresponding URI. Certimeter Express performs some tests in real time, verifying

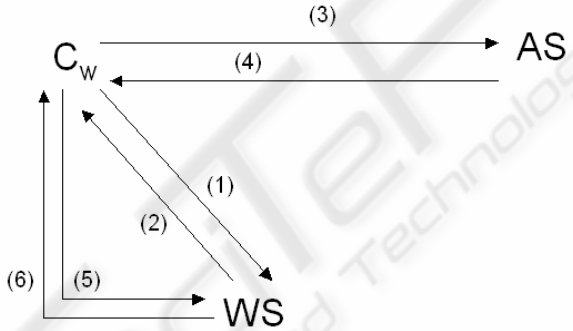

Fig. 2. How web requests are served when the verification component is added to the web

some requests when the web server serves them. The Certimeter Express component is added to the controlled web server, as shown in Figure 1.

To describe in a few words what happens in a normal request to the web server, let us say that a client sends an HTTP request to the web server, and the server responds with the data requested, or eventually an error.

In case a certification is performed using our system, the way the web server replies to the client requests is modified with respect to the normal operation. This modified behavior (shown in Figure 2) is obtained by adding the component Certimeter Express (see Figure 1 and previous section 3) to the standard set of components of WS.

When a client C_w that never connected to WS asks for a resource to WS (1), then the Certimeter Express component changes the normal operation of WS, and redirects (sending an HTTP code of Moved Temporarily) the client (2) to the audit server AS (3). This redirection contains also, as a parameter, the originally requested resource,

to allow the AS for another redirection. Moreover, Certimeter Express keeps track of the IP of this client in a local list. The job of the AS is to log the request and to redirect (4) the client C_w to the originally requested resource. The audit server may accomplish this by means of a CGI script that simply extracts the parameter containing the original resource and redirects to it.

When a client that previously issued a request to WS asks for a resource to WS, then the request is served normally (Certimeter Express avoids a new redirection checking in its local list for the presence of the IP). This allows WS to serve normally (5, 6) the client C_w for the present request and all the subsequent ones. In this way, only when a never seen client IP connects to the web server it is controlled with a redirection to the AS, but successive requests are served normally. The consequence of this fact is that the impact of the verification on the client navigation in the web site is minimal. Furthermore, the existence of the client machine navigating the web site WS is verified through the connection of the client to the Audit Server AS.

This mode of operation allows to immediately know the number of different IP addresses that connected to the controlled web server: in fact, all the different IP addresses are collected in the local list of the Certimeter Express module, and a counter may keep track of their number.

Another consideration is that the redirection performed by AS is very fast, having a small load also on the machine operating the AS.

Now the question is: why we do verify only the first request for each IP, and not all the subsequent ones? One reason is that we want this method to be efficient: in this sense, we decided to require the client to connect to the audit server only once.

At a first sight, this could expose the technique to an attack (in this context, by attack we intend that it is possible for some entity to alter the log file, without the certification agency noticing this behavior). In fact, knowing the methodology, when one sees an IP address in the WS log file that has already been verified, then he could add any faked request with that IP, because he knows that it would not have been verified anymore with a redirection to the audit server. Here come in play the Controlling clients. It is important that they are unknown to WS, and they must appear as any other client on the Internet. These clients perform requests, but are controlled by the certification agency. Thus their accesses to WS are perfectly known, and any discrepancy with the log file content (in terms of number of requests made by that IP and/or requested URI) may be easily detected during the certification phase.

The next question to pose is how many of these controlling clients distributed on the Internet are needed. It may be shown that a very small number of these clients is necessary. This is essential for a real application of this technology, because it may be impossible, for the certification agency, to control a large number of clients.

4.1 How many controlling clients?

To answer the question, we may consider sampling the IP addresses contained in a log file, the same way as quality tests do. When performing sampling on a set of objects, the objective is to look for objects that do not satisfy some constraints, without having to test all the objects in the set being tested. Our task here is to check that none of the IP addresses in the log file had some of the requested URIs modified or added. To obtain this, we use the controlling clients as samples of the IP addresses (in fact, from

the controlled web server point of view, the controlling clients are random; moreover, they will never be known to this server). The result of the sampling will be probabilistic, in the sense that after having checked that all the sampled IPs satisfy the constraints, then the probability that some of the remaining IPs had their entries altered (at this point, without noticing by the certification agency) will be below a predefined threshold. A discussion on the certification of large amount of data using sampling techniques may be found in [5]. We present here a discussion that shows the probability of failing to detect that an IP in the log file had its log line entries modified. A similar application is discussed in [5].

Assume that a log file contains N different IP addresses, and of these IP addresses let's suppose that n had their log entries modified (or added). For the moment, let m be the number of IP addresses whose machines are controlled by the certification agency. The probability that none of the controlled machines has one of its log entries modified is:

$$P = \left(1-\frac{n}{N}\right)\left(1-\frac{n}{N-1}\right)\left(1-\frac{n}{N-2}\right)\cdots\left(1-\frac{n}{N-(m-1)}\right)$$

Note that all the factors of the product are less than or equal to:

$$\left(1-\frac{n}{N}\right)$$

This yields:

$$P \le \left(1-\frac{n}{N}\right)^m$$

which may be maximized by:

$$P \le e^{-\frac{nm}{N}} = P_F$$

This equation indicates that the probability of failing to detect the modification of some accesses depends on:

- the number of different machines used as controlling clients;
- the fraction of IP addresses that had some of their entries modified/added, with respect to the total number of IP addresses that contacted the controlled web server.

We report in Table 1 some values of P_F for different values of the number of controlling clients and different ratios of modified IP addresses.

Table 1. P_F w.r.t. some parameter values (N.B. values are rounded in the last digit).

n/N	*m=10*	*m=20*	*m=40*	*m=60*
10 %	0.3679	0.1353	0.0183	0.0025
20 %	0.1353	0.0183	0.0003	0.000006
30 %	0.0498	0.0025	0.000006	$1.523*10^{-8}$

The certification agency is able, according to the probabilities in the previous table (or computing other probabilities using the latter equation) to estimate the risk of not being able to detect log file modification. An important parameter that the certification agency must decide is the minimum fraction of IP addresses (i.e. n/N) that may be modified by the owner of the controlled web server. Using that parameter, it is possible to fix the number of controlling clients and compute the probability of failing to detect modifications in the log file. If this probability is not

acceptable, then the number of controlling clients should be increased. From a practical point of view, if we assume that 20%, or more, of IP addresses have their accesses modified, then using 40 controlled clients gives a probability of 0.9996 to discover log file modification.

According to the operations of the system described so far, when a machine connects to the controlled web server WS, it is checked with the audit server once, and after that it should be no more verified. Given that access statistics on a web site are produced according to time intervals, for example daily, then our method foresees a periodic reset of the local data of the Certimeter Express component. The reset may be programmed to take place, for example, every twelve hours, or daily, or weekly. The effect of a reset is to clean all the IP addresses of which the controlled server component is keeping track. Thus, after a reset, every machine that contacts the web server will be redirected to the audit server (given that it is an IP never seen before). As previously said, the redirection will take place only in the first request of every new IP address.

5 Running Experimentation

The system we described is being experimented on a set of production web servers. The web site that is testing this technology is Finanzaonline (www.finanzaonline.com, www.finanzaonline.it). This site operates a set of web servers having millions of hits per day. We found very useful a test on a site having a large number of requests, because the technology we have presented may be faced with many different possible situations every day.

We developed the ideas presented in this paper implementing an Apache module (see [2]) that modifies the Apache server normal operations to allow for the redirection of new IP addresses. Thus, the module was installed on all the web site's servers. The AS being used is located on the network of the University of Turin. At the present stage, only the functionalities of the module, written in C language, are under testing. We found no perceptible performance degradation of the web servers in serving the client requests. This is due to the fact that only the first request of a never seen IP address is redirected, while the following requests are served almost normally: in fact, the module performs only a check to see if the IP address requesting the web resource has already been redirected. The check is made through a search in a hash table kept in memory and maintained by the Certimeter Express module.

5.1 Module characteristics

We report here some characteristics of the implemented module that we think useful for working in a real environment, and that the experimentation helped us to tune. First, the IP addresses already redirected are stored in memory in an efficient data structure. For efficiency we mean that data search and insertion should have a low computational complexity. The reason is that for every hit (i.e. a request coming from a client), the IP address of the machine requesting a resource must be searched, and

eventually inserted for future reference. Our choice was to store, as previously said, the IP addresses in a hash table in memory.

Second, more than one audit server may be used. This implies that the module may choose among a set of available audit servers which one to use. The reason for this will become clear in the following discussion. Third, the module periodically contacts the audit servers, to see which are active. This is done because there may be situations in which some of the audit servers are unavailable. In that case, the redirection should take place towards an available audit server. In case no audit servers are available, then no redirection is made (because the client would be unable to contact the audit server). Instead, the module keeps track that a particular client was not redirected, leaving the redirection for an eventual subsequent request when some audit server will be again available.

Fourth, the module is configurable in real time; this means that audit servers may be added or removed from the set of the audit servers to use. Note that the use of many audit servers implies only that the log of the redirections is distributed among a set of servers, instead of being on only one.

Fifth, the module periodically (or at the web server shutdown) writes a backup on disk of the redirected IP addresses. The reason for this is that in case the controlled web server is stopped then the content of the hash table in memory is cleaned. When the server is restarted, the IP addresses that were already verified should not be redirected again to an audit server. To deal with this possible situation, the module loads the backup, if it exists, at the server start up, filling the hash table with the saved IP addresses.

6 Future developments

In this section we present the future developments of the technology discussed so far.

At the present stage, the system shown in Figure 1 is running and is being experimented as discussed in section 5. The verification procedure, that will produce a certification of the log file content, is under development. We illustrate here the steps that are involved in the check of the log file performed by the certification agency.

The first step is to download the log files from the audit server(s) and to download the lists of IP addresses from the controlled web server(s) (the latter operation must be done using a secure connection with the module on the web server). Here comes the first consideration. Some web sites have more than one server (like in a server farm), and requests are distributed to the various servers taking into account considerations of load balancing and machine availability. Then the Certimeter Express module should be installed on each server and should maintain its list of IP addresses. It may happen that an IP address is replicated in some lists, because each server has redirected and served the same IP address: in this case the IP address would have contacted the audit server(s) more than once (i.e. a number of times as the number of different web servers its requests have been sent for load balancing). Thus, the second step of the verification must check that all the clients in the module lists have contacted the audit server(s) a correct number of times. If all the checks are positive,

at this stage it is already possible a certification of web site accesses, namely the number of different IP addresses that contacted the web site.

To avoid multiple redirections for the same IP address due to the presence of many servers in the server farm, a solution could be for the Certimeter Express modules to exchange among them the data of every IP address as soon as they answer a redirection. We did not made this choice for the following reasons:

- this would have added network traffic between the servers;
- redirections are not heavy for clients, so a limited number would not be perceptible by the user navigating the controlled web site.

The third step of the certification involves testing the accesses of the controlling clients. That is, the certification agency must download the log files from the controlled web servers, and get from the controlling clients all the resources they requested during the period of time whose accesses have been logged. At this point, the log files may be verified with the following constraints:

- every request in the log files coming from a controlling client must correspond to a real request of that client;
- every IP address having made a request recorded in a log file must appear in the list of the Certimeter Express module of the corresponding controlled web server.

If all the controls made in the various steps satisfy the constraints, then a certification of truthfulness of the log files may be produced by the certification agency. This certification may simply consist of the digital signature of the log file made by the certification agency.

A situation that may arise is when a client does not follows the redirect. That is, even if the controlled web server sends an HTTP response of Moved Temporarily, the client does not contact the audit server. To our knowledge, this behavior may happen for one of the following reasons:

a. the client may be programmed to do not accept redirects;
b. the user browsing the web site may stop the navigation before the client contacts the AS.

In both cases it is not responsibility of the controlled web site owner the lacking of an IP address in the audit server log file. But the situations a. and b. are indistinguishable from the spoofing of that IP address. To deal with this situation, the certification agency should accept a certain percentage of IP addresses that do not appear on the audit server(s) log files but have contacted the controlled web server. This percentage threshold may be determined empirically observing the behavior of clients on a set of trusted sites running the Certimeter Express module, evaluating the average and the variance of the proportion of clients that do not contact the audit server(s).

6.1 Counting users

This subsection presents a discussion on a method we are planning to insert in our module with the aim of counting the different users.

As described in section 4, the Certimeter Express module is able to count the different IP addresses that connect to the controlled web server. There are contexts in which the same IP address may be assigned to different users. For example, an

Internet Service Provider may assign to different users at different times, the same IP address; or a set of users may be hidden behind a proxy server, that connects to the web server on behalf of the clients. In these cases our technology correctly counts the different IP addresses, but does not count all the users. A method we plan to use for counting different users are the server cookies. Simply stated, a cookie is a piece of information that a server sends to a client, and the client *may* send this information back to the server in every successive connection. This cookie may be used to keep a context in the interaction between the server and a specific client.

Our idea is to send a cookie to a client if this client does not send a cookie in the request. In all the subsequent requests the client will send back the cookie to the server, and the server will identify that client with the pair <IP address, cookie>, keeping this pair in a local list. The cookie expiry date and time should be set to the time of the next module list reset (see section 4.1). Thus, according to the pair, the Certimeter Express module will decide to redirect or not the client to the audit server. In principle, when a client performs the first request to the controlled web server, it will not send any cookie; thus, the module will answer back a cookie along with a redirection to an audit server (if any available). This redirection will contain the cookie as a parameter, in such a way the audit server will log the cookie. This allows the identification of the client in the verification procedure. If a request comes from a client that also sends the cookie, then the pair <IP address, cookie> is searched in the module local list, and inserted in this list if not present. In case we are certifying log files of a server farm, this client will not be redirected because we assume that this is not the first request of the client (it has the cookie!), and that another web server of the farm previously redirected this client. In case the certification involves a single server, then a client should not send a cookie in the first request (it should not have one), but in that case it will be redirected to the audit server anyway. If the same user contacts the web server using two different IP addresses, he will be counted twice.

We think cookies are a possible tool that may help in identifying a client. But, in case the presented method is extended using cookies, one must be aware that cookies are managed on the client side, thus they may be disabled, or may remain undeleted after the expiry date.

7 Conclusions

In this paper we presented a novel technology for the certification of the content of web server access log files. We discussed the involved parties, how they interact with each other according to the HTTP protocol and our method. At the present time we have developed a software module for the Apache web server that implements all the functionality of Certimeter Express, and this method is under testing in a production web site having more than one web server. Preliminary tests have shown that our added module does not influence the web server performance, but deeper tests should be done. We are confident that our technology has a very low impact on the users' browsing: in fact the client is redirected to the audit server only once (i.e. the first time it connects to the controlled web site), and all the subsequent requests are not verified.

We illustrated how we plan to build the verification procedure, based upon the data collected by the module and by the audit server(s). Also, a further development has been presented, that plans the use of cookies for the counting of individual users that access a site.

Acknowledgments

The authors thank Dr. Sergio Rabellino for the help in testing this technology at an early stage on a Computer Science Department web server, and Mr. Natalino Picone and the Finanzaonline management for the collaboration in verifying this technology on production web servers.

References

1. Anupam, V., Mayer, A., Nissim, K., Pinkas, B., and Reiter, M. K., 1999. On the security of pay-per-click and other web advertising schemes. In *Proc. of the 8th International World Wide Web Conference*.
2. Apache, http://www.apache.org
3. Bergadano, F., and Cavagnino, D., 2000. Certificazione di Accessi a Server Web. In *Proc. of AICA 2000*, Taormina, Italy.
4. Bergadano, F., and Galvan, F., 2001. Facile e utile "gonfiare" le statistiche web. *Available at* http://www.i-dome.com.
5. Bergadano, F., Cavagnino, D., and Egidi, L., 2002. Partially sighted signatures on large documents. In *Proc. of Int. Network Conference 2002*, pp. 373-380.
6. Blundo, C., De Bonis, A., Masucci, B., and Stinson, D. R., 2000. Dynamic Multi-Threshold Metering Scheme. In *Proc. of Selected Areas in Cryptography 2000*, D. R. Stinson and S. Tavares, eds., *LNCS 2012*, Springer-Verlag, pp. 130-144.
7. Capozzi, F., 1998. Sicurezza della rete: IP spoofing ... ed il MAC ???. *Available at http://www.linuxvalley.com.*
8. Certimeter, http://www.certimeter.com
9. Franklin, M. K., and Malkhi, D., 1997. Auditable Metering with Lightweight Security. In *Proceedings of the Financial Cryptography Workshop*.
10. Franklin, M. K., and Malkhi, D., 1998. Auditable Metering with Lightweight Security. WIPO WO9826571.
11. Berners-Lee, T., Fielding, R., and Frystyk, H., 1996. Hypertext Transfer Protocol -- HTTP/1.0. *Internet Engineering Task Force RFC 1945*, May 1996.
12. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T., 1999. Hypertext Transfer Protocol -- HTTP/1.1. *Internet Engineering Task Force RFC 2616*, June 1999.
13. Masucci, B., and Stinson, D. R., 2000. Metering Schemes for General Access Structures. In *Proc. of 6th European Symposium on Research in Computer Security – ESORICS 2000*, Tolouse. In F. Cuppens, Y. Deswarte, D. Gollmann and M.Waidner, eds., *Lecture Notes in Computer Science*, no. 1895, Springer-Verlag, pp. 72-87.
14. Masucci, B., and Stinson, D. R., 2001. Efficient Metering Schemes with Pricing. *IEEE Transactions on Information Theory*, Vol. 47, No. 7, November 2001, pp. 2835-2844.
15. Naor, M., and Pinkas, B., 1998. Secure Efficient Metering. *Proc. of Eurocrypt 1998, Lecture Notes in Computer Science*, no. 1403, Springer-Verlag, pp. 576-590.

16. Naor, M., and Pinkas, B., 2000. Method for secure accounting and auditing on a communications network. US Patent #6,055,508, April 25th, 2000.
17. Reiter, M. K., Anupam, V., and Mayer, A., 1998. Detecting Hit Shaving in Click-Through Payment Schemes. In *Proceedings of the Third USENIX Workshop on Electronic Commerce*, pp. 155-166.
18. Shamir, A., 1979. How to share a secret. *Communications of the ACM*, Vol. 22, pp. 612-613.