4th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2010)

Two Level Efficient User Authentication Scheme

Nikolaos G. Bardis, Member IEEE and
Nikolaos Doukas
Dept of Mathematics and Engineering Science
Hellenic Army Academy
Vari - 16673, Greece
bardis@ieee.org, nikolaos@doukas.net.gr

Oleksandr P. Markovskyi
Department of Computer Sciences National Technical University of Ukraine, (Polytechnic Inst. of Kiev), Peremohy pr., Kiev 252056, KPI 2003, Ukraine
markovskyy@mail.ru

Athanasios Drigas
NCSR 'Demokritos', Institute of Informatics and Telecommunications, Net Media Lab
Agia Paraskevi, 153 10, Athens, Greece
dr@imm.demokritos.gr

Abstract— User authentication procedures are a computer system's first line of defense against malicious attacks that could potentially jeopardize its availability and reliability. Computational complexity considerations may prevent system designers from implementing efficient authentication procedures. This paper presents a robust two level authentication procedure is presented that is highly efficient and suitable for multi-user systems. Authentication is attained in only a single cycle of exchanges between the user and the authentication system. Further advantages arise from the fact that with the proposed technique, it is not necessary to store lists of passwords and therefore the overhead caused in systems with large numbers of registered users is minimal. The proposed architecture is shown to be reliable while creating significant economies in the amount of information that needs to be stored. These facts therefore render the architecture highly suitable for hardware implementations.

Keywords- Authentication, Multi user systems, Secure non – volatile memory

I. INTRODUCTION

Current trends show an ever expanding utilization of information systems in security sensitive applications, such as military, public safety, banking, electronic commerce fiscal etc. This expansion produces worries about the risks arising from illegal accesses to the resources of such information systems. The difficulties of designing and implementing efficient access control procedures are accentuated by the fact that current systems usually need to support large numbers of users.

Given that attacks to such systems are very common and increasingly intelligent, significant work has taken place towards the design of secure authentication procedures [1] that are based on stored passwords, on smart cards or on zero knowledge techniques. The rewards that a successful illegal access may entail for the intruder are a driving force for potential cyber-criminals to devote more effort into mounting their attacks. Consequently, the same force drives the need for improving the means of deterring unauthorized access to critical systems.

The problem becomes even more acute with the use of embedded systems that are used for controlling equipment or performing other important tasks and do not possess the computational resources required to implement extremely complex algorithms. Preserving computational resources is a requirement even for bigger systems given the large numbers of users they need to support and the computational strain they are under anyway in order to serve their purpose.

II. ANALYSIS OF EFFECTIVENESS PROBLEMS FOR USER AUTHENTICATION

Subscriber authentication is a task which is of paramount importance to the overall computational system integrity and is simultaneously a task to which one would not be prepared to devote a significant amount of computing resources. The reason for this is that authentication is a side activity that can be considered as overhead, since it has no immediate benefits. There are two factors that will determine the efficiency of an authentication scheme. The first factor is the volume of illegal access attempts traffic (the amount of computing resources devoted to dealing with such attempts). The second factor is the processing power necessary in order to service the authentication of legitimate users. More advanced and more reliable authentication schemes will usually involve the use of more complicated algorithms and will absorb a larger amount of system resources that will have to be drawn away from other tasks.

Subscriber or user authentication involves the prospective user responding successfully to a challenge set by the computing system to which they require access. Therefore, one may recognize two distinct stages in any authentication procedure. Firstly the user registers, acquiring authentication information and hence strict authentication occurs, during which phase the legitimacy of the access request is controlled. There exist a variety of authentication procedures, based on many different algorithms that are currently in use [2], [3]. Most of these protocols are based on one of two principles: authentication based on passwords or on a "zero knowledge" approach. Authentication based on zero knowledge is considered effective; however, existing implementations demand significant computational resources [2]. For this reason, authentication based on passwords is widely used in a variety of different systems. Smarter password based techniques have been proposed that provide a high level of reliability.

Attacks to authentication systems may include observation by third parties of the information exchanged during a

legitimate authentication, as well as interception and replacement of such information. This is facilitated by the fact that authentication exchanges inevitably happen over unprotected, public channels. Attacks may go as far as to attempt to acquire information about the system by closely monitoring the activity of the system itself during a legitimate authentication [2]. Such attacks may be materialized via a variety of methods, e.g. via the use of computer viruses, via the successful accidental authentication of an unauthorized user or when a legitimately authenticated user attempts to widen their own rights or grant access rights for system resources to illegitimate users. Additionally, the security of the system may be jeopardized when personnel responsible for maintaining the system breaches security maliciously or due to negligence.

Authentication is mainly implemented as programs that process all the necessary information that is stored locally or is input by the user. Complex algorithms that have slow implementations, however robust they may be, prohibit repeated authentications during the same user session. Repeated authentications are not feasible in such cases, since they would occupy more processing power than desirable. Similarly, the existence of many registered users means that searches froe their authentication data are slow and hence impractical to repeat regularly. It should be noted though that if it were possible to re-authenticate the legitimate user at regular intervals during a session, the probability of such a user to manage to extend their rights and gain access to restricted system resources would be minimized.

The most reliable method for the minimization or even elimination of the ability of illicit access to system resources and information is the authentication based on the use of an additional memory access action that achieves security via the device level storage of information. VLSI memories are currently bulk manufactured by a large number of suppliers. The secure memory circuits are capable not only of storing information, but also of executing procedures belonging to authentication protocols so as to eliminate temporary storage and execution in computer system nodes.

The main drawback of existing VLSI secure memory devices is the relatively small volume of information that they can store, compared to the rapid increase of the authentication information required as the number of legitimate users increases. The study presented in [4] proposes an authentication architecture that requires just one code for the authentication of all users. However, the use of this architecture does not offer any protection from the possibility of legitimate users accessing unauthorized resources of the computer system. The principle advantage of this architecture is the variability of the authentication message between successive system accesses.

The above analysis demonstrates that the development authentication procedures that do not require the storage of large amounts of per user information, represents a very urgent necessity. The development of such procedures could contribute significantly to the development of authentication protocols that are extremely more efficient than existing ones in averting the danger of illegal system accesses. Several issues regarding efficiency of authentication methods in multi-user distributed systems are given in [5], [6] and [7]. In a recent paper, the idea of computational complexity reduction in user authentication via a two level architecture was presented by the authors [8]. In the following sections, the full authentication algorithm and an associated protocol will be presented that require minimal amount of stored information and thus are extremely efficient while still being very robust.

III. ARCHITECTURE OF A SECURE MEMORY DEVICE

The principal aim of a device that implements a secure memory (SM) system is the storage of the authentication information and the enforcement of access control on this information. This memory must be energy independent so as to eliminate the danger of unauthorized users gaining access to the stored information during a write cycle.

For the design of devices that implement secure memories, the following principles must be taken into account:

1. The SM that is implemented at device level needs to have an open connection interface for the standardized system buses and especially the PCI bus. The parameterization of the SM and its identification within the computer's address space must be compatible to the Plug and Play principle.

2. The SM stores the information that is the object of the security protocol. Such information should not be written to the SM or read from it in open form.

3. The SM needs to implement the cryptographic processing functions that exploit the secret information, apart from the storage functions they implement anyway. This implies that the effectiveness of the SM immediately depends on its specialization: the more the initial processing functions operating on the raw data that the SM implements, the more efficient the implementation of the security protocol becomes.

The architecture of the SM that is oriented for use in a user authentication system is presented in Figure 1.

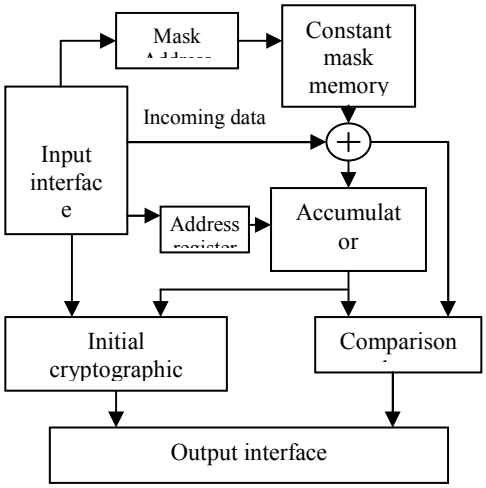

Figure 1. Block diagram of the proposed architecture

During the write cycle of the information to the accumulator, the address arrives at the input interface and becomes valid in the address register. The actual data that needs to be stored in the secure memory are masked with a

code that needs to be known only to the principal administrator of the computer system. For the masking, the bitwise logical XOR is used.

All the masking codes, that are secret and only known to the system administrator, are stored in the non – volatile memory of the SM. During the write cycle of the secret information, it is necessary to define the address of the mask to be used. This address is defined in the Mask Address Register. Consequently, the secret masking code is read from the non-volatile memory. In the logical XOR unit, the mask is subtracted from the input raw authentication code and the result is written in the memory. This sequence of actions that compose the write cycle, eliminate the possibility that secret information be captured during their storage into a SM.

Using a similar sequence, the mask is used for storing the data that are compared during the authentication procedure with the code that are stored within the SM. The code that is necessary to be compared to the secret code that is stored inside the SM is masked by the administrator and delivered to the input of the SM. The masking is subtracted within the SM.

Simultaneously, the address of the secret code to be compared is placed in the input interface. This secret code is read from the memory and compared to the code that has been given. The comparison is performed at device level within the SM. The result of the comparison is returned to the host computer via the output interface. Apart from the SM device comparison operation, the primary processing of the secret information during the application of cryptographic algorithms is also implemented within the SM device.

One of the basic problems of the practical use of the SM device is the fact that existing security protocols do not take into account the particular characteristics of such memories. In particular, they do not take into account its limited capacity. For this reason, it is necessary to develop authentication protocols that are specially oriented to the device level use of SMs

IV. TWO LEVEL AUTHENTICATION ARCHITECTURE FOR MULTI USER COMPUTER SYSTEMS

High level of efficiency in remote user authentication in multi user systems is achieved via the compromise between contradicting needs. The complexity of the problem is accentuated by the inability to devise a sufficient action protocol for the communication of the different participant of the authentication procedure. To a first approach, the authentication procedure must satisfy the following requirements:

1. Distribution of the data storage between the user and the system space. Each communicating part will hence not be able to access system resource services on its own.

2. The password must be user supplied, may not be stored in memory, must be input for the establishment of every session and must not be sufficient on its own for gaining access to system resources.

3. Minimization of the use of the transmission lines, which is the part of the system that is most vulnerable from the point of view of illegal intrusion to system resources.

4. Change of the authentication exchanges for the user information for each system access session.

5. The volume of the secret information that is stored in the system and is necessary for the user authentication procedure must be minimized.

6. The authentication information exchanged on the transmission line must be encrypted with keys that are the same for all users.

7. The identification of legitimate users and the determination of their access rights must be implemented by different security mechanisms.

Satisfying all the above requirements is extremely complex within a one-level authentication architecture. It is hence necessary to distribute the realizations of the user authentication level and the system resources access rights establishment level, into two distinct authentication subsystems. This distribution is implemented within the context of the development of a multi-level user authentication architecture for multi-user systems.

The two-level architecture for user authentication separates the two functions described above. The first level establishes the legitimacy of the user without using the system information relevant to that user. The second level uses the user information stored in the system. This principle of the use of authenticated information makes possible an acceleration of the filtering process that distinguishes legitimate access requests from illegal attempts to access system resources that are made from unauthorized users.

In the proposed architecture for the user authentication, both symmetric R=SCT(D,K) and non – symmetric R=NSCT(D,KD) cryptographic transformations are used. In the first case the Rijndael algorithm could be used, while for the second case the RSA is a suitable choice. The variable D represents the unencrypted data packet, R is the encrypted data packet and K the transformation key. The inverse transformations are defined as:

$$D=\mathrm{SCT}^{-1}(R,K) \text{ και } D=\mathrm{NSCT}^{-1}(R,K_R),\ K_D \neq K_R \quad (1)$$

Apart from the transformations mentioned above, the proposed architecture presupposes the use of a function P(X) that permutes the bits of the code X (P-1 represents the inverse transformation such that X=P-1(P(X))). During the authentication, the hash transformation is used that forms the hash signature X. For this function any hash algorithm could be used, such as the SHA.

V. USER REGISTRATION PROCEDURE

During the registration of user A, a random password P_A is created that is inserted by the subscriber with every session of their access to the system. With the registration this password is transferred to the system, where it is scrambled with the

constant secret code *W*: $K_A=P(P_A, W)$. The block diagram of the user registration process is presented in Figure 2.

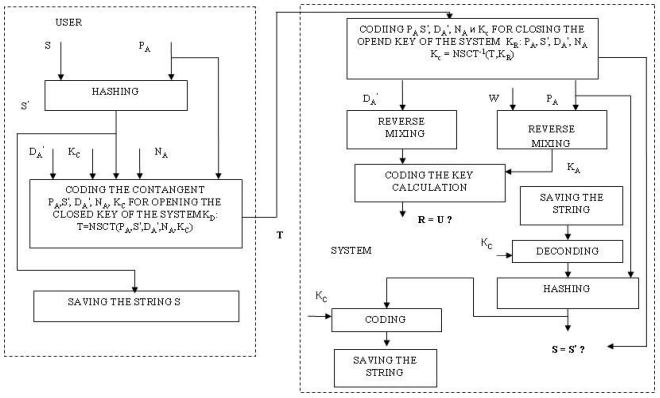

Figure 2. Block diagram of the user registration process

The code K_A produced is used as part of the key for the generation of the universal key for all the subscribers of the system. The subscriber access code D_A is hence obtained as $D_A=SCT(U, K_A)$. Subsequently, the scrambled version of the access code D_A is calculated as $D_A'=P(D_A)$ and is returned to the subscriber subsystem together with its serial number N_A. In the system memory, a memory area is allocated where the code N_A is assigned an address. A random sequence S is produced that is stored in the memory segment of subscriber A. This sequence together with D_A and N_A are returned to the subscriber A.

The exchange of the registered information is performed in encrypted form. For this purpose, the subscriber uses the system open key K_D to encrypt the password P_A that they need to store. This is done via a randomly selected session key that the subscriber supplies K_C: $T_1=NSCT((P_A, K_C), K_D)$. Within the system, using the unlock key K_R the codes P_A, K_C are recovered. Next, using the recovered session key K_C the system encrypts part D_A of the user password, the number N_A and the stored sequence *S*, $T_1=NSCT((D_A, N_A, S), K_D)$.

After storing in memory the access code P_A, the subscriber stores on their local computer system the partial password D_A and the sequence S that they have obtained from the system. The block diagram of the subscriber authentication, including the system access is presented schematically in Figure 3.

During the user access to the system, the authentication cycle follows the sequence of steps listed below.

1. Subscriber A inputs the password P_A that they have stored on which the sequence S is appended. The result is calculated via the hashing function *H* as $S'=H(P_A|S)$. The sequence *S'* is stored in memory in the place of sequence *S*.
2. On the subscriber computer system the calculation of the contingent is performed that is based on the locally stored password P_A, the second part of the access code D_A', the number N_A and the sequence *S'*. The result of the calculation is encrypted using the system lock key K_D: T=NSCT(P_A,D_A',N_A,S'). The obtained code *T* is sent to the system.
3. The multi-user system receives the authentication codes *T*, the request of subscriber A and using the unlock key K_R, the decryption of the code is performed: P_A,D_A',N_A,S'=NSCT^{-1}(*T*).

The inverse transformation of the system bits recovers the incoming code $D_A=P^{-1}(D_A')$ and calculates the code of key K_A via scrambling with the constant secret code: *W*: $K_A = P(P_A,W)$.

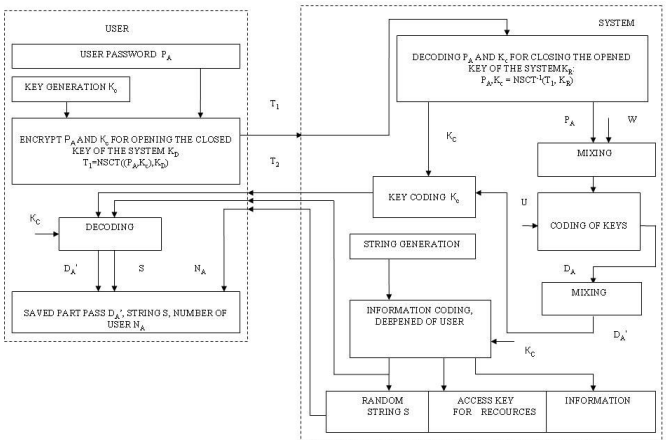

Figure 3. Block diagram of the subscriber authentication process, including the system access

4. Using the code K_A the inverse cryptographic transformation is calculated for the code D_A: R=SCT^{-1}(D_A,K_A). If the result is equal to the code *U*, i.e. $R=U$, then a decision can be made that A has submitted a legitimate request. On any other case, access is denied.
5. Once it has been established that subscriber A has been successfully authenticated, then the access rights of subscriber A are also established for the resources that are available to the subscriber. For this purpose, the sequence S_A is read from the memory indicated by N_A. The sequence S_A is then appended to the user stored code P_A. On this result, the hash function is applied and a new sequence is obtained S_A'=H(P_A | S_A). The result of the hashing function S_A' is compared with the sequence S' obtained from the subscriber. If these sequences coincide, i.e. S_A' =S', then the subscriber is allowed to make use of the system resources that are determined in memory N_A. In this case the sequence *S'* is stored in the place of the previous sequence S_A' in memory N_A. If $S_A' \neq S'$, then the instance is recorded as a legitimate access to the system, but access of certain of the resources is denied. Hence the access is not allowed and the sequence in memory is not replaced.

Consequently, the proposed procedure implements a two level subscriber authentication in multi-user systems. On the first level and from the system side, just three secret codes are used: the unlock key K_R and the codes *W* and *U* that are chosen at random at system setup and are the same for all subscribers.

The first level establishes the fact that the subscriber is legitimate and corresponds to steps 4-5 of the transformation with the shared secret code U, but not with the data of the authentication information directory and are implemented in the known remote user authentication procedure. This increases the speed of the authentication for remote users and this authentication speed does not depend on the numbers of these users. In this case, the volume of the secret information that is stored in the system is reduced by several orders of magnitude. This fact significantly simplifies the actual implementation of the secret memory.

On the second level the authentication is achieved via the comparison of the sequences produced by the user and the system. This ensures the random change of the sequence of authentication exchanges for each system access. The analysis of failed authentication attempts allows effective identification of repudiated illegal access attempts for system resources. The overall security of the systems may hence be significantly enhanced.

The proposed authentication procedure employs a single authentication information exchange cycle for each system access. This reduces the danger of illegal penetration to the system by injection of data during the transfer of this authentication information. Additionally, the load imposed upon the transfer channel – a resource that is important for multi-user, mass access systems – is significantly reduced.

The importance of the proposed method also lies in the fact that within this framework a diversification of the information used for the authentication is achieved: the user selected password need not be stored in the memory of the local computer. This fact reduces the danger of a third party obtaining access to this code, while additionally, an attack based on trying to guess this code is pointless since this code alone does not allow access to system resources.

In the memory of the subscriber computer system, the additional part of the access code D_A is stored only in a transformed form D_A'. The shared code W is stored in the closed memory of the system. This prevents the user that does not have access to the codes D_A and W, as well as the permutation function P, to gain access to the code U. For gaining access to the system, an illegal user needs to know the password P_A, the code D_A that corresponds to it, the number N_A and the code of sequence S. The sizes of D_A and P_A are 256 (given the symmetric Rijndael algorithm and the size of S is longer than 256. These numbers make the probability of a successful illegal authentication on both levels extremely small.

For a legal user to acquire access to restricted system resources, it is necessary that they can successfully reproduce that connection between the one-way transformations and the codes P_A, D_A, N_A and S. In the case of access to the shared memory, it is possible that the sequence S and the numbers in the memories may be obtained, but this information does not guarantee access. For access, it is necessary that the codes P_A and D_A are obtained.

VI. Results

The proposed two level organization of subscriber authentication in multi-user systems reduces the risks of illegal access to system resources both by illegal and legal users, by separating the authentication in two levels. The first level establishes the legitimacy of the user and the second level verifies the access rights of these users. In contrast to know authentication schemes, the determination of the legitimacy of the user is produced without access to the memory associated to that user. This renders feasible the reduction of the volume of secret information to just three codes: K_R, W and U, that are common for all subscribers and thus simplify the implementation of the specialized protected memory. A complete security analysis is currently under way and will be presented in a future publication

VII. Conclusion

A novel procedure for a two level scheme for the authentication of remote users was presented. The proposed method achieves authentication during a single cycle of information exchange between users and the system and hence reduces the overhead of the authentication process while minimizing the danger of malicious third parties intervening during this process. The proposed scheme eliminates the need for stored lists of passwords or lookup procedures. Increases in the number of registered users therefore causes minimal burden onto the overall performance. The volume of stored information was shown to be small enough for an economical hardware implementation of the overall scheme in a device with an independent power supply and secure memory.

References

[1] Bengio S., Brassard G., Desmedt Y.G. Goutier C.,Quisquater J.J. “Secure implementation of identification system”, Journal of Cryptology, v.4, n.3, 1991, pp.186-192.

[2] Rhee, H., Kwon, J. and Lee, D. H. A remote user authentication scheme without using smart cards. Computer Standards & Interfaces 31 (2009) pp 6–13.

[3] Menezer A.J., Van Oorschot P.C., Vanstone S.A. Handbook of Applied Cryptography. CRC-Press.- 780 p. ,1997.

[4] Nikos G. Bardis, Alex Polymenopoulos., Evgenios G. Bardis, Alexander P. Markovskyy, (2004) “Methods for Increasing the Efficiency of the Remote User Authentication in Integrated Systems”, TRENDS IN COMPUTER SCIENCE, Volume 12 No.1, ISBN 1-59454-065-9, Nova Science Publishers, Inc, New York, pp.99-107, 2003.

[5] Braz, C and Robert, J.M. “Security and usability: the case of the user authentication methods”. Proceedings of the 18th International Conference of the Association Francophone d'Interaction Homme-Machine. 2006, pp 199-203.

[6] Wang, H., Sheng, B, Tan, C and Qun, L. “Comparing Symmetric-key and Public-key Based Security Schemes in Sensor Networks: A Case Study of User Access Control”. Proceedings of the 28th International Conference on Distributed Computing Systems, 2008, pp 11-18.

[7] Jia-Lun Tsai “Efficient multi-server authentication scheme based on one-way hash function without verification table”, Elsevier Computers & Security, Volume 27, Issues 3-4, 2008, pp 115-121

[8] P.Stavroulakis, O.Markovskyi, N.G. Bardis, N. Doukas. “User Authentication in Multi User Systems Based on Secure Memory”. To appear in IASTED Information and Communication Technology (ACIT-ICT 2010)