An Access Control System with Time-constraint Using Support Vector Machines

Chin-Chen Chang[1], Iuon-Chang Lin[2], and Chia-Te Liao[3]
(Corresponding author: Iuon-Chang Lin)

Department of Information Engineering and Computer Science Feng Chia University[1],
Taichung, Taiwan, R.O.C. (Email: ccc@cs.ccu.edu.tw)
Department of Management Information Systems, National Chung Hsing University[2],
250 Kuo Kuang Rd., Taichung, Taiwan 402, R.O.C. (Email: iclin@nchu.edu.tw)
Department of Computer Science and Information Engineering, National Chung Cheng University[3],
160 San-Hsing, Min-Hsiung, Chiayi 621 Taiwan, R.O.C.

(Received Sep. 21, 2005; revised and accepted Oct. 4, 2005)

Abstract

Access control is an important issue in information security. It is a necessary mechanism for protecting data in a computer system. In this paper, we apply support vector machines to construct an access control system. The security management framework is a pattern classification system using well trained support vector machines. The proposed system can not only be applied to broad access policies, but the access rights of users can also be associated with a time period. The proposed scheme is useful to make predictions on real-time data and it is better adapted to complex computer systems.

Keywords: Access control, information protection, neural network, security, support vector machines

1 Introduction

Resource sharing is one of the most important benefits in the client/server architecture or distributed environment. For example, a file server provides a centralized management and shares the storage of remote data among a number of workstations. If distributed workstations want to access some files, they send their requests to the file server via a communication network. The file server facilitates resource sharing among the autonomous workstations and has some obvious benefits, such as easy management, economy, and so on. On the other hand, a database is also a very important element in resource sharing in modern computer usage. It allows sharing of data among many users or applications. Now, almost every business application is supported by some sort of database in order to work, and thus more and more secret data are being stored in databases for easy access.

However, the resource sharing systems are dangerous if they are without information protection mechanisms. Because the amount of secret information, such as business plans, military maps, and so on, is available to illegitimate or unauthorized users, more and more threats to secret data stored in computer systems are increasing exponentially with the growth of network users and technological developments. Therefore, how to achieve privacy and integrity of information is becoming an important issue in modern computer security.

Access control is one of the efficient ways to prevent information from being destroyed, altered, disclosed, or copied by unauthorized users. Access control makes decisions on all users accessing a system. The policies of access control define the rules which permit certain subjects (such as users) to invoke objects (such as programs or files) in a computer system and which give them access right to operations (such as to executions, own, reading, or writing). Typically, the policies of access control can be defined according to five components [12].

1) Users: The policy can identify what a user is allowed to do. The term "users" could refer to the people sitting at a terminal or workstation or the processes which run the computer system.

2) Resources: The policy can specify what resources a user can access. The term "resources" could refer to the programs, services, and data accessed by users.

3) Operations: The policy can specify what operations the user is permitted to invoke from the resource. The term "operations" refer to the actions that can be performed on a resource. For instance, one user may be permitted to write a file, whereas another user may only read the file.

4) Authority: The authority policy is to ensure that the access control is implemented according to the policies of the management of an organization. The term "authority" refers to the legitimate power to make policy decisions.

5) Domain: The policy can specify the boundary of the resources or users. For instance, a manager usually has higher authority than the resources and people in a department.

On the other hand, authorization control also can be divided into three categories [9]. The first category is called discretionary access control (DAC). In this category, the system leaves the specification of access control policies to individual users and controls the access of users by authenticating the identity after proper verification.

The second category is called mandatory access control (MAC). The subjects and objects in MAC are classified into many clearances and classifications. Furthermore, the access right for each subject is defined by a system administrator. Typically, the policies of MAC must satisfy two restrictions, namely, (1) no read up and (2) no write down.

However, the DAC and MAC models are not flexible enough to provide a variety of access control policies. Since the access policies of these models are pre-defined and have been built into the access control mechanisms, they cannot support the dynamic requirements needed in modern application environments.

In order to deal with this problem, the third category of an access control model, called role-based access control (RBAC), is presented. Currently, RBAC is the most popular access control model. RBAC can not only support the policies of DAC and MAC but can also support more complex policies. The roles of RBAC are created for various job functions in an organization, and users are assigned roles based on their responsibilities and qualifications. In this model, users can easily be reassigned from one role to another. This greatly simplifies the management of access policies [14].

So far, many schemes in the three categories have been proposed for controlling access to computer systems [7, 9, 14, 15]. In these schemes, once a user tries to access a protected file, the user needs to provide proof to the system that he/she is the authorized user. Password or key verification is the most popular technique for identifying an authorized user. Once an outside user wants to access certain resources in a system, the user has to submit a secret password or key for verifying the access privilege to the resource.

However, some complex computer systems may need to change the access right with time. Thus, a file can sometimes be accessed and sometimes not. For example, a file may be updated every morning by the system administrator, and we wish that no one can access this file during this time period to avoid any data inconsistency. The property is very flexible and useful in many modern applications. In this paper, we shall propose a novel access control scheme, called time-constraint access control, which can satisfy such application environments. In the time-constraint access control model, the access policies are associated with the time period. In different time periods, a user has different access rights.

Our time-constraint access control scheme is constructed from support vector machines. Support Vector Machines (SVMs) is a useful classification tool based on statistical learning theory [17]. Because of the good generalization ability of SVMs, SVMs have been widely and successfully applied in a number of fields, such as handwritten digital recognition, face detection, particle identification, and text categorization [13]. In this work, we shall construct an access control verification system by using the technology of SVMs and make it applicable to a time constraint access control model. In this system, a user only needs to remember one password and present it to the verification system to prove that he/she has the access right during the defined time period. The input pattern of the verification system is the user's password and the system-assigned time period, and the output is the object domain and the access right during the time period that can be acquired by the user.

Before describing the proposed scheme, we will briefly introduce the related works on access control in Section 2. It is worthwhile to note that the reviewed schemes have already solved the problems of access control in computer systems. However, these schemes are not applicable to time-constraint access control policies. Following this review, an overview of support vector machines with different goals is described, as well as the technologies that will be used in our scheme. Section 4 provides a detailed description of our new time-constraint access control scheme using SVMs. Then, our experimental results and extensive discussions are given in Section 5. Finally, we shall summarize the benefits that our scheme provides in Section 6.

2 Previous Works

In this section, we shall briefly review some related schemes in the area of access control in computer systems. Typically, an access control system involves two entities, namely, (1) identity and (2) identifier. Identity is the user who approaches the access control system and claims to obtain access privilege to the system. Identifier is the authentication mechanism that authenticates whether the identity is a legitimate user. So far, many authentication mechanisms have been proposed to authenticate a user's access privileges. In 1972, Graham and Denning [8] first introduced an abstract protection model. In their model, the security state of an access control system is defined by an access control matrix, where rows correspond to users, columns correspond to the protected files, and the entries correspond to the lists of users with access rights to the protected files. A simple access control matrix for a file system is shown in Figure 1. The entry a_{ij} stands for the access right of the user U_i with respect to the file F_j. If one user has no access right to a file, the corresponding entity a_{ij} in the matrix is assigned a "null" value. For example, the user U_1 has access rights to execute the files F_1 and F_3 and to write the file F_4. When a user tries to access a file, the system will check the matrix to verify the user's access right. The access control model is simple and easy to implement. In general, however, there is a huge amount of data stored in a file system. Therefore, the size of the access control matrix would be very large, and most of the entities in the matrix are nulls because each user is usually only allowed access to a subject of the files. If we store the whole access control matrix, the memory needed for the whole matrix becomes impractically large, and the utilization of storage is very low. On the other hand, the system must maintain and protect the matrix from being modified by an intruder.

a_{ij}	F_1	F_2	F_3	F_4
U_1	2	Null	1	4
U_2	Null	1	Null	2
U_3	2	Null	1	4
U_4	1	Null	Null	2
U_5	4	1	3	2

Null: No access
1: Execute
2: Read
3: Write
4: Own

Figure 1: Access control matrix

To overcome the problems faced by the above scheme, Wu and Hwang [18] proposed a single-key-lock (SKL) model for implementing the access control system. In their model, each user U_i is associated with a key vector K_i, and each file F_j is associated with a lock vector L_j. Therefore, the access right a_{ij} for U_i to F_j can be formulated as $a_{ij} = f(K_i, L_j)$, where $f()$ is a predefined function. Since then, several methods have been developed for a single-key-lock access control model. Chang [1, 2, 3] proposed three of them based on the Chinese remainder theorem, Euler's theorem, and prime factorization. Laih et al. [11] proposed a single-key-lock model based on Newton's interpolating polynomials. The previous schemes work well, but they are impractical for complex computer systems. A drawback to this kind of model is that it is not flexible for dynamically controlling the access rights of users to protected files.

Role-base access control provides a way to efficiently protect information in complex systems with many users and resources. In the RBAC model, roles are created for various job functions in an organization and users are assigned roles based on their responsibilities and qualifications [7, 9, 14, 15]. Hence, RBAC allows a wide range of policies to be implemented. Furthermore, in some complex application environments, the access rights of users to the protected files should be changed according to a defined time period. Therefore, the access policies for all users to the resources should be associated with time. For this reason, time-constraint role-based access control is more useful and practical than traditional access control models in complex computer systems. As far as us aware, no one has ever proposed a time-constraint access control model. However, in this paper, we apply support vector machines to construct a practical time-constraint access control scheme.

3 The Review of Support Vector Machines

We have seen that more and more researchers are currently engaging in studies of SVMs. In addition, in recent years, there have been many interesting applications with the good abilities of SVMs to meet some desired requirements. It is also that most of them have attained very successful results in fields such as text categorization, hand-written pattern recognition, bioinformatics, as well as others. Basically, SVMs are systems of linear learning machines with good generalization ability that makes use of the optimization theory for efficient training in kernel-induced space. At the same time, the generalization theory provides insights to control the complexity and keep the consistency of the hypotheses [10, 16].

To make it easy to understand, we can simply describe an SVM problem as follows: Suppose that we have a given set of instances that satisfy the requirement $(x_i, y_i) \in S$, where (x_i, y_i) denotes an instance in S. Here, $S = \{(x_1, y_1),\ (x_2, y_2),\ \cdots (x_i, y_i)\} \subseteq (X \times Y)^l$ is the training set of examples for $i = 1, 2, \cdots, l$. The notation l is the number of training instances in S. For each $x_i \in X$ and $y_i \in Y$, $X \subseteq R^d$ is the input space while Y is the output domain. Note that we will use d to refer to the dimension of the input space X. Therefore, for cases of binary classification, the output domain we have is $Y = \{1, -1\}$; for cases of m-class classification, the output domain we have is $Y = \{1, 2, \cdots, m\}$; and for cases of regression, the output domain we have is $Y \subseteq R$. SVMs act as the learning methodology used to find the hyperplanes which divide all the examples so that all the instances with the same labels stand on the same side when it comes to classification, or it is used to get a linear function that best interpolates the set of examples in regression cases. The basic concept of SVMs with the simplest binary classification is shown in Figure 2 [6]. In this section, we will briefly review the various conditions of SVMs in the following subsections.

3.1 Support Vector Classification Cases

Let us start with the simplest conditions for classification. Assume that we want to separate the set with $Y = \{1, -1\}$ for the binary classification cases. In other

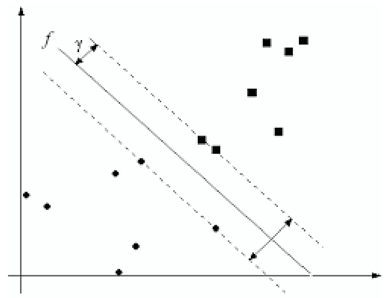

Figure 2: The optimal separating hyperplan (OSH) and its margin

words, our job is to find the hyperplane f of w and b, subject to

$$y_i(w \bullet x_i + b) > 0, \tag{1}$$

for $i = 1, 2, \cdots, l$, where the notations w and b are the weight and bias, respectively. The notation l denotes the number of training instances. If there is a hyperplane f which satisfies Equation (1), then the set of examples S is said to be separable. Besides, the hyperplane f of w and b is always able to be scaled such that

$$y_i(w \bullet x_i + b) \geq 1,$$

for $i = 1, 2, \cdots, l$. In Figure 2, each point is an example instance in S, and the notation $\gamma = \frac{1}{\|w\|}$ denotes the distance from an arbitrary point closest to f. The quantity $\gamma = \frac{2}{\|w\|}$ in Figure 2 is called the margin, and it can be treated as a benchmark of the generalization ability of hyperplanes [17]. By minimizing the quantity of $\|w\|^2$, we can obtain the Optimal Separating Hyperplane (OSH) with the largest margin, i.e. the best generalization ability. It is an optimization problem. Assume that for the training set S, a solution of OSH f^* in the form (w^*, b^*) has been derived. Then the decision function $h(x)$ will be:

$$h(x) = sign(w^* \bullet x + b^*).$$

With this decision function, therefore, we are able to classify an unclassified data instance $z \in X$ by employing $h(z)$ to determine which class it belongs to. However, sometimes in real-life situations, the instances are not so easily classified. To handle that, just as we select the architecture for a neural network, we often adopt kernel representations in order to increase the computational power of linear SVMs when we design our scheme [5]. Kernel representations will project data onto a more meaningful and higher dimensional space to provide non-linear approximation abilities. This is extremely useful when we handle problems in real life that are not so easy to solve. The sketch map of kernel representations is shown in Figure 3.

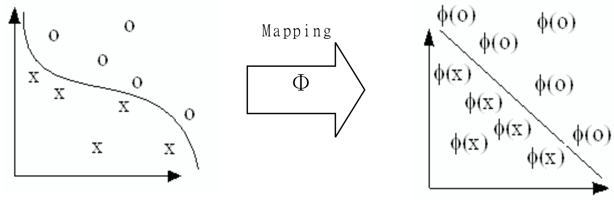

Figure 3: Simplifying the classification task by feature mapping

3.2 Support Vector Regression Cases

The goal of regression problems in SVMs is to find a function f that best interpolates a given set of examples. That is, we can apply support vector methods such that it learns a non-linear function in the kernel-induced feature space. To get better generalization abilities, we will ignore the errors within a certain distance ε of a true value. The ε-insensitive loss function $L_\varepsilon(x, y, f)$is thus defined as:

$$L_\varepsilon(x, y, f) = \max(0, |y - f(x)| - \varepsilon).$$

The notation f expresses a real number function in the input space X. Note that $x \in X$ and $y \in R$, which is the same as we defined previously. In brief, a simple regression problem now can be formulated as:

$$\begin{aligned} &\textit{minimize} && \| w \|^2 + C \sum_{i=1}^{l} (\xi_i^2 + \hat{\xi}_i^2), \\ &\textit{subject to} && ((w \bullet x_i) + b - y_i \leq \varepsilon + \xi_i, \textit{ and} \\ & && y_i - ((w \bullet x_i) + b) \leq \varepsilon + \hat{\xi}_i, \end{aligned}$$

where ξ_i, $\hat{\xi}_i \geq 0$ for $i = 1, 2, \cdots, l$. The parameter C used here is to exhibit the penalty, and it will perform the function when f makes errors. Just as Figure 4 depicts, corresponding to each instance x_i the slack variable $\hat{\xi}_i$ is for that which exceeds the target value by more than ε, and the other slack variable ξ_i is for that which is below the target with a distance more than ε. We can convert the optimization problem into another form with the Lagrange multipliers induced: maximize

$$W(\alpha) = \sum_{i=1}^{l} y_i \alpha_i - \varepsilon \sum_{i=1}^{l} |\alpha_i| - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j K(x_i, x_j) \tag{2}$$

subject to $\sum_{i=1}^{l} \alpha_i = 0$, $-C \leq \alpha_i \leq C$, for $i = 1, 2, \cdots, l$. Note that the notation $K(a, b)$ in Equation (2) is to represent a projection of two vectors a and b into the desired meaningful space, that is, the kernel representation was induced here to make the prediction task more easily.

Assume now that we have found a solution α^* satisfying Equation (2). Then the regression function best interpolating S will be:

$$f(x) = \sum_{i=1}^{l} \alpha_i^* K(x_i, x) + b^*.$$

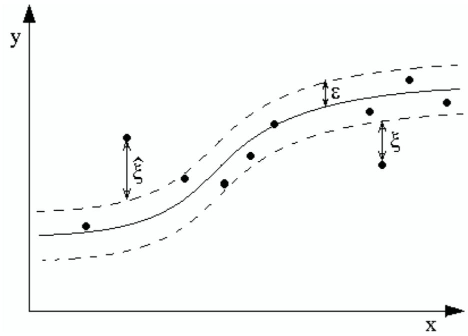

Figure 4: The insensitive band of a non-linear regression function

And, finally, we are able to apply the found regression function f to further predict an unknown instance which has the same similarities as the training examples.

4 The Proposed Scheme

Let us start to describe the proposed scheme with a simple example. Assume that we have an enterprise organized into several different departments. All this company's secret business information is stored in a server or a database, and employees can directly access these secret files according to their positions in the company. However, due to security reasons, the privileges of system users must be restricted according to their identities; for example, it is only natural that a sales manager can not access technical reports which belong to the department of engineering. Moreover, if it is not the time that an employee is supposed to work, he/she, of course, is not allowed to open these secret files. So, here, we give the proposed scheme for access control that satisfies all the security requirements mentioned above. The excellent prediction abilities of SVMs will be exploited to classify system users into different levels of security according to the input passwords. And we also include the time factors to guarantee that users can not overdo except during the office hours. This means that the users' security levels should be changed according to the users' login times. Briefly speaking, first, all system users will be separated into different groups (i.e. departments). Then, to vary the access rights within a group of people in different job positions, various passwords will be given corresponding to the groups and their different security levels. For instance, a manager should have a higher security level than his/her assistant, but he/she still can not access restricted files from other departments. Finally, after the training procedure is used to feed the expected output values into SVMs, we are able to apply SVMs to classify users into their appropriate groups and security levels.

As Figure 5 depicts, the proposed access control system can be roughly divided into three portions: (1) the input pattern transforming phase, (2) the training phase for SVMs, and (3) the authority decision phase. We will give a detailed explanation of the three parts in the flowing three subsections.

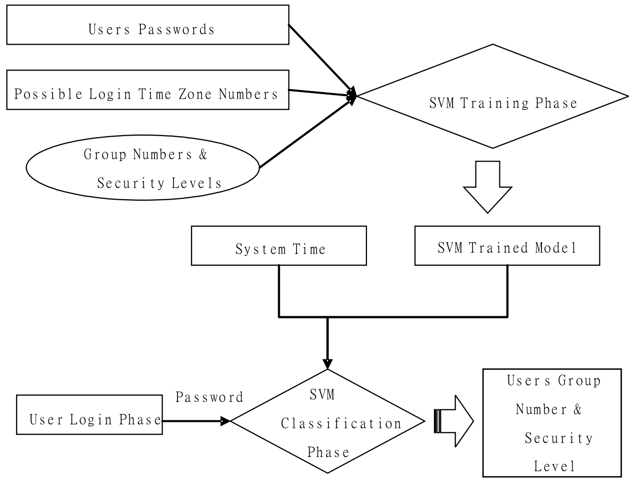

Figure 5: Block diagram of the proposed access control system

4.1 Input Pattern Transforming Phase

In this subsection, we shall describe how to make the used input patterns in the proper form for SVMs. For convenience, it is usual that users' passwords are composed of English words and/or numerical characters. To make these textual data suitable for SVM problems, first, we have to convert these passwords from characters into numbers so that we can further train out an appropriate model of SVMs. To do so, a mapping of characters to numbers should be constructed in advance so that every textual input instance can be transformed into a numerical form before being fed into SVMs. In the proposed scheme, we directly adopted the most commonly used ASCII code as the relationship to build this mapping, which is shown in Table 1.

For example, if a user's password is "kMXoeUZR," then the converted input pattern will be $(t, 107, 77, 88, 111, 101, 85, 90, 82)$, which is a 9-dimensional vector. Here, the additional element t denotes the number used to represent this user's login time section. We will use it with an explanation in the flowing subsection.

4.2 The Training Phase for SVMs

In this subsection, all the input patterns of users will be taken into the training procedure in order to build the SVM models. Suppose that we have a set of collected passwords for users in advance. Before the transforming procedure, all the input patterns have to be converted into the desired numerical formats, as we showed in Subsection 4.1. Then, considering the group numbers and security levels of the users' identities, we assemble these training examples at first in order to further train the

Table 1: The mapping table from characters to numbers

character	a	b	c	d	e	f	g	h	i	j	k	l	m
mapping	97	98	99	100	101	102	103	104	105	106	107	108	109
character	n	o	p	q	r	s	t	u	v	w	x	y	z
mapping	110	111	112	113	114	115	116	117	118	119	120	121	122
character	A	B	C	D	E	F	G	H	I	J	K	L	M
mapping	65	66	67	68	69	70	71	72	73	74	75	76	77
character	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
mapping	78	79	80	81	82	83	84	85	86	87	88	89	90
character	0	1	2	3	4	5	6	7	8	9			
mapping	48	49	50	51	52	53	54	55	56	57			

needed SVM models. Every training example here is composed of a 9-dimensional vector with the features of x_i, together with one single bit from the desired target label. Being the same as the simple example we just gave, these 9-dimensional feature vectors are composed of one element to represent the users' login times and 8 elements of the transformed passwords. Every desired target label here will be treated as a series of bits and processed separately in order to reduce the complexities of the SVM training. Note that in the proposed scheme, an unbroken target label can be viewed as two various parts to express the number of user groups and its corresponding security levels.

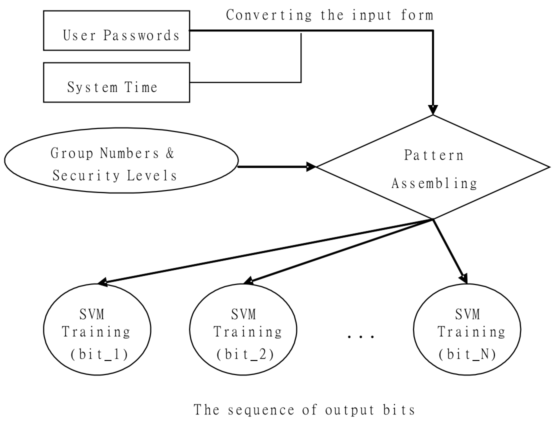

Figure 6: The sketch of the pattern training procedure

As we show in Figure 6, in the implemented architecture, we use one support vector machine to predict one bit of the desired target values. In other words, there are more than one SVMs used in the proposed scheme. Actually, there will be a total of $\log(mn)$ SVMs adopted, where the notation m is the number representing a group and n stands for the total number of different security levels. Formally, what we mentioned above can be formulated into an SVM problem as:

$$S = ((x_1, y_1), (x_2, y_2), \cdots, (x_l, y_l)) \subseteq (X \times Y)^l,$$

where S is the set of given training examples, and l denotes the number of instances in S. Here, $X \subseteq Z^9$ is the input space and Y is the output domain. Notice that in the implementation, we will take one support vector machine which will try to predict one single bit of the target label y_i so that y_i must be detached into a series of bits being predicted separately. For each i from 1 to l, where $x_i \in X$ and $y_i \in Y$, getting the linear functions for each bit of y_i that best interpolates the examples is the eventual goal. In short, we are going to apply the support vector regression function in each SVM node and try to predict a value close to 0 or 1 according to the given targets. Finally, after the training procedure, we can obtain the models used for the prediction of SVMs.

4.3 Authority Decision Phase

After the training procedure, now with the trained models, we can classify the users into their groups and give them corresponding security levels using their passwords. First, the system administrators (SA) have to finish the training procedure in advance to obtain the models used by the SVMs. Then, in this phase, every user will be given a password by the SA to use as the key directly related to his/her access rights. Moreover, in the proposed scheme, another important thing is that by the use of the precise classification abilities of the SVMs, the access rights of each user is able to be changed according to the login time. For example, a company's SA can give specific employees permission to modify some files only in the morning, but they do not have permission to write when evening comes. Therefore, with the proposed access control system, the SA can easily build his/her supervision policy using SVMs; the system users will be precisely classified into their own groups and given their security levels according to the login time. Therefore, in the proposed scheme, we can give the convenience that successfully makes use of the excellent prediction abilities of SVMs to achieve a fluctuant access control policy with the time factors involved.

5 Experimental Results

We will practically evaluate the performance of the proposed scheme in this section. As described above, the security levels of system users need to be varied according to the different zones of login time. So, to meet this requirement, in the experiments, we divided the hours in a day into 8 distinct time zones for simulation; that is, by assigning a distinct number from 0 7 to each time period consisting of 3 hours, we can easily tell when the users log in. Besides, in order to test the correctness of the proposed access control system using SVMs, the testing patterns (composed of passwords and the numbers representing the time zones) in the simulation procedure were entirely the same as those we took in the training stage. Note that for every security level within a group, there can be more than one password. By feeding these patterns into the SVMs with the models trained previously, therefore, we were able to see if the resulting outputs equal to the expected values.

Library of Support Vector Machines (LIBSVM), which are available in [4], is an efficient and easy-to-use software implementing support vector learning. It not only solves problems with ease by using SVMs, but it also provides several selective kernels in order to fit the data with different types. We made use of LIBSVM as the tool for our SVM simulating process. In order to obtain the most suitable kernel type of SVMs, we have tested several kinds of kernels in the experiments which are linear kernel $K(x, y) = (x \bullet y)$, polynomial kernel $K(x, y) = (\gamma x \bullet y + c)^d$ and RBF (Radial Basis Function) kernel $K(x, y) = e^{-\rho\|x-y\|^2}$, respectively. Actually, in the experiments, we used the RBF kernel representation for our data type because of its better performance in training speed. Note that the type of kernel used has to be the same as those in both the training and predicting procedures. We took a total of 8 groups for the simulation. Moreover, there were 8 different security levels within a group in the simulation. In other words, along with the 8 time zones in a day, there were a total of 512 testing instances in the evaluation. Table 2 illustrates the organization of the used instances in our experiments. They are composed of the number of login time zones, passwords, group numbers, and the expected values of security level. Note that for each password, it is capable to give the different kinds of security levels for the variant login time zones. As shown in Table 2, the first three bits of the expected output values are meant to express 8 different groups in the experiments, and the last three bits stand for the values of security level corresponding to the 8 different time zones.

In Table 3, we have shown the resulting values derived from 6 units of SVMs which performed the function of support vector regression. We see that the predicted errors are bounded very tightly (< 0.01) compared to the expected values; that is, these resulting values can be transformed extremely easily the desired bits of security levels. Therefore, by simply quantizing these output values to the desired bits of group numbers and security levels, we can perfectly achieve the purpose of access control. Note here that we can arbitrarily scale the range of the errors made by the prediction of the SVMs; this is only a simple task of tuning the parameters. However, the smaller the range of the errors, the more time will be taken when the SVMs try to converge in the training phase. Actually, after the procedure of quantizing these output values with a threshold, a perfect correctness percentage is achieved by using the all 512 testing instances.

6 Conclusions

In this paper, we proposed a time-constraint access control scheme using support vector machines. In this scheme, the access control system does not store or maintain an access control matrix. Only the trained support vector machines need to be stored in the system. Using support vector machines, the system can authenticate the access rights of users to the protected files. The main advantage of this scheme is that it is applicable to broad access policies, such as the access rights of users that can be associated with the time. The system user only holds one password and then he/she can access various files with various access rights at various times. As the experimental results show, the system indeed can authenticate the access rights of users by using all the 512 testing instances. Therefore, our scheme is practical such that it can be implemented in more flexible applications.

Acknowledgement

The authors are grateful to Chih-Chung Chang and Chih-Jen Lin for making LIBSVM software available on the website for researchers.

References

[1] C. C. Chang, "On the design of a key-lock-pair mechanism in information protection system," *BIT*, vol. 26, pp. 410-417, 1986.

[2] C. C. Chang, "An information protection scheme based upon number theory," *Computer Journal*, vol. 30, no. 3, pp. 249-253, 1987.

[3] C. C. Chang and D. C. Lou, "A binary access control method using prime factorization," *Information Science*, vol. 96, no.1-2, pp. 15-26, 1997.

[4] C. C. Chang and C. J. Lin. (2001). LIBSVM: a library for support vector machines. [Online]. Available WWW: http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[5] N. Cristianini and J. S. Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, 2000.

[6] C. Cortes and V. Vapnik, "Support-vector network," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995,

[7] D. Ferraiolo, J. Barkley, and R. Kuhn, "A role based access control model and reference implementation within a corporate," *ACM Transactions on Information and System Security*, vol.2, no.1, pp. 34-64, Feb. 1999.

Table 2: The set of training instances and their expected outputs

	Input		The Expected Output					
	Time Zones	Passwords	Group Numbers			Security Levels		
GROUP_0 POLICY_0	0	kMXoeUZR	0	0	0	0	0	1
	1	kMXoeUZR	0	0	0	1	1	1
	2	kMXoeUZR	0	0	0	0	1	1
	3	kMXoeUZR	0	0	0	1	1	1
	4	kMXoeUZR	0	0	0	1	0	1
	5	kMXoeUZR	0	0	0	1	1	0
	6	kMXoeUZR	0	0	0	0	0	0
	7	kMXoeUZR	0	0	0	1	1	0
GROUP_0 POLICY_1	0	vb6oDuJN	0	0	0	1	0	0
	1	vb6oDuJN	0	0	0	0	1	0
	2	vb6oDuJN	0	0	0	0	0	1
	3	vb6oDuJN	0	0	0	1	0	0
	4	vb6oDuJN	0	0	0	1	0	1
	5	vb6oDuJN	0	0	0	0	1	1
	6	vb6oDuJN	0	0	0	1	1	1
	7	vb6oDuJN	0	0	0	1	1	1
GROUP_0 POLICY_2	0	pcLWHFoK	0	0	0	0	0	1
	1	pcLWHFoK	0	0	0	0	1	0
. . .								
GROUP_7 POLICY_6	0	Rks3Rnx	1	1	1	0	1	1
	1	Rks3Rnx	1	1	1	1	1	0
	2	Rks3Rnx	1	1	1	1	1	1
	3	Rks3Rnx	1	1	1	1	1	0
	4	Rks3Rnx	1	1	1	0	1	1
	5	Rks3Rnx	1	1	1	1	0	1
	6	Rks3Rnx	1	1	1	0	0	1
	7	Rks3Rnx	1	1	1	0	1	0
GROUP_7 POLICY_7	0	3y285xp	1	1	1	1	1	0
	1	3y285xp	1	1	1	0	0	0
	2	3y285xp	1	1	1	0	1	1
	3	3y285xp	1	1	1	0	1	1
	4	3y285xp	1	1	1	0	0	1
	5	3y285xp	1	1	1	1	1	1
	6	3y285xp	1	1	1	1	1	0
	7	3y285xp	1	1	1	0	1	0

Table 3: The classification results of group numbers and security levels

	Time Zones	The Output Results					
		Group Numbers			Security Levels		
GROUP_0 POLICY_0	0	0.01000	0.01000	0.01000	0.00984	0.00983	0.99004
	1	-0.01000	-0.01000	-0.01000	0.99028	0.99033	1.00990
	2	-0.00047	-0.00047	-0.00047	0.00953	1.00943	0.99016
	3	0.01000	0.01000	0.01000	0.99059	0.99073	1.00978
	4	0.00999	0.00999	0.00999	1.00942	0.00926	0.99022
	5	-0.00047	-0.00047	-0.00047	0.99045	0.99058	0.00982
	6	-0.00999	-0.00999	-0.00999	0.00973	0.00964	-0.00989
	7	0.00999	0.00999	0.00999	0.99013	0.99019	0.00994
GROUP_0 POLICY_1	0	0.01000	0.01000	0.01000	0.99001	0.00989	-0.00982
	1	-0.01000	-0.01000	-0.01000	0.00994	0.99019	0.00967
	2	-0.00047	-0.00047	-0.00047	-0.00989	0.00967	0.99052
	3	0.01000	0.01000	0.01000	1.00985	-0.00959	0.00934
	4	0.00999	0.00999	0.00999	0.99016	0.00960	0.99064
	5	-0.00047	-0.00047	-0.00047	0.00986	0.99030	1.00951
	6	-0.00999	-0.00999	-0.00999	0.99009	1.00982	0.99028
	7	0.00999	0.00999	0.00999	1.00994	0.99009	1.00987
GROUP_0 POLICY_2	0	0.01000	0.01000	0.01000	-0.00989	-0.00995	0.99023
	1	-0.01000	-0.01000	-0.01000	0.00978	1.00986	0.00956
⋮							
GROUP_7 POLICY_6	0	0.98999	0.98999	0.98999	0.00984	0.99003	0.99019
	1	1.01000	1.01000	1.01000	0.99031	1.00992	0.00963
	2	1.00047	1.00047	1.00047	1.00946	0.99013	0.99059
	3	0.98999	0.98999	0.98999	0.99068	1.00983	0.00925
	4	0.99000	0.99000	0.99000	0.00930	0.99017	0.99073
	5	1.00047	1.00047	1.00047	0.990549	0.00985	1.00944
	6	1.01000	1.01000	1.01000	0.00966	-0.00990	0.99032
	7	0.99000	0.99000	0.99000	-0.00982	1.00996	0.00984
GROUP_7 POLICY_7	0	0.98999	0.98999	0.98999	0.98999	1.00995	-0.00998
	1	1.01000	1.01000	1.01000	-0.00993	-0.00985	0.00998
	2	1.00047	1.00047	1.00047	0.00990	1.00976	0.99001
	3	0.98999	0.98999	0.98999	-0.00987	0.99031	1.01000
	4	0.99000	0.99000	0.99000	0.00986	0.00966	1.00999
	5	1.00047	1.00047	1.00047	0.99010	0.99027	0.99001
	6	1.01000	1.01000	1.01000	1.00993	1.00983	0.00998
	7	0.99000	0.99000	0.99000	-0.00996	0.99009	-0.00998

[8] G. S. Graham and P. L. Denning, "Protection-principles and practices," *Proceedings of Spring Jt. Computer Conference*, vol. 40, pp. 417-429, 1972.

[9] M. Hitchens and V. Varadharajan, "Design and specification of role based access control policies," *IEE Proceedings of Software*, vol. 147, no. 4, pp. 117-129, 2000.

[10] C. W. Hsu and C. J. Lin, "A comparison of methods for multi-class support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415-425, 2002.

[11] C. S. Laih, L. Harn, and J. Y. Lee, "On the design of a single-key-lock mechanism based on Newton's interpolating polynomial," *IEEE Transactions on Software Engineering*, vol. 15, no. 9, pp. 1135-1137, 1989.

[12] J. D. Moffett and M. S. Sloman, "The source of authority for commercial access control," *IEEE Computer Journal*, vol. 21, no. 2, pp. 59-69, 1988.

[13] S. Raudys, "How good are support vector machines?," *Neural Networks*, vol. 13, no. 1, pp. 17-19.

[14] R. Sandhu, V. Bhamidipati, and Q. Munawer, "The ARBAC97 model for role-based administration of roles," *ACM Transactions on Information and System Security*, vol. 2, no. 1, pp. 105-135, Feb. 1999.

[15] R. Sandhu, E. J. Coyne, and H. L. Feinstein, "Role based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38-47, Feb. 1996.

[16] S. K. Shevade, S. S. Keerthi, C. Bhattaacharyya, and K. R. K. Murthy, "Improvements to the SMO algorithm for SVM regression," *IEEE Transactions on Neural Networks*, vol. 11, no. 5, pp.1188-1193, 2000.

[17] V. N. Vapnik, Statistical Learning Theory, Wiley-Interscience Publication, 1998.

[18] M. L. Wu and T. Y. Hwang, "Access control with single-key-lock," *IEEE Transactions on Software Engineering*, vol. 10, no. 2, pp. 185-191, 1984.

Chin-Chen Chang was born in Taichung, Taiwan on Nov. 12th, 1954. He obtained his Ph.D. degree in computer engineering from National Chiao Tung University. He's first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in computer and decision sciences. Both were awarded in National Tsing Hua University. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005.

Prior to joining Feng Chia University, Professor Chang was an associate professor in Chiao Tung University, professor in National Chung Hsing University, chair professor in National Chung Cheng University. He had also been Visiting Researcher and Visiting Scientist to Tokyo University and Kyoto University, Japan. During his service in Chung Cheng, Professor Chang served as Chairman of the Institute of Computer Science and Information Engineering, Dean of College of Engineering, Provost and then Acting President of Chung Cheng University and Director of Advisory Office in Ministry of Education, Taiwan.

Professor Chang's specialties include, but not limited to, data engineering, database systems, computer cryptography and information security. A researcher of acclaimed and distinguished services and contributions to his country and advancing human knowledge in the field of information science, Professor Chang has won many research awards and honorary positions by and in prestigious organizations both nationally and internationally. He is currently a Fellow of IEEE and a Fellow of IEE, UK. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. He also published over 850 papers in Information Sciences. In the meantime, he participates actively in international academic organizations and performs advisory work to government agencies and academic organizations.

Iuon-Chang Lin received the B.S. in Computer and Information Sciences from Tung Hai University, Taichung, Taiwan, Republic of China, in 1998; the M.S. in Information Management from Chaoyang University of Technology, Taiwan, in 2000. He received his Ph.D. in Computer Science and Information Engineering in March 2004 from National Chung Cheng University, Chiayi, Taiwan. He is currently an assistant professor of the Department of Management Information Systems, National Chung Hsing University, Taichung, Taiwan, ROC. His current research interests include electronic commerce, information security, cryptography, and mobile communications.

Chia-Te Liao received the M.S. in Computer Science and Information Engineering from National Chung Cheng University, Chiayi, Taiwan, Republic of China. His current research interests include information security and cryptography.