Controlling the Dependency of User Access Control Mechanisms on Correctness of User Identification

G. O'SHEA*
99 Russell Road, Buckhurst Hill, Essex, 1G9 5QF

The effectiveness of User Access Control mechanisms is largely dependent upon correctly establishing user identity, and a diversity of techniques with varying degrees of accuracy, reliability cost and convenience to the user may be employed in verification of identity. This paper describes an approach allowing this dependency on the accuracy and reliability of identity verification techniques to be considered and controlled within programmed User Access Control mechanisms.

Received March 1987

1. INTRODUCTION

Shared computing services often require protection against unauthorised activity by individuals or processes using the system, where authorised activities are defined in accordance with a desired security policy.

Logical access control systems enforce predefined security policies by directing all access requests for the protected object type through an implementation of a reference monitor,[19] which mediates in all accesses to the set of resources under its control.

The majority of reference monitors exist within the controlled environment of a single computing system, and relate only to other entities existing within that same environment. In such cases, the identity of the originator of an access request and the identity of the target object are usually fixed with the system and trusted by other objects belonging to the system.

User Access Control (UAC) mechanisms are a special case in that they are required to mediate in access requests that originate from human users operating beyond the system boundary, such that the identity of the requester is no longer reliably known. Thus UAC systems are particularly vulnerable to deceptions involving falsified or counterfeit requester identity.

In order to understand and control this vulnerability, criteria for comparing and controlling the effectiveness of identity verification techniques are required.

Measurement of the information delivered in an authentication check offers a general metric for comparing the relative strengths of different authentication mechanisms, or for measuring different authentication parameters used in a particular type of authentication mechanism.

This alone does not provide sufficient criteria for measuring the certainty of correct identity verification, since many authentication mechanisms are also vulnerable to compromise. The extent to which a particular mechanism is at risk of compromise is not necessarily related to the quantity of information produced by the mechanism, and may be dependent upon the nature of the mechanism, and on the conditions under which it has been used. This concept may also be expressed using measurements of information quantity.

Authentication strength is a quantity measuring the certainty with which an identity is authenticated, being derived from some combination of the above quantities for an authentication mechanism in a particular instance.

The primary functions of a User Access Control mechanism[5] may be defined as:

1. Identification of the user
2. Authentication (verification) of the user
3. Authorisation of access requests.

Thus UAC mechanisms differ from reference monitors generally in respect of the first and second functions.

Personal identity verification may be achieved on the basis of one or more of the following:[1,11-15,20]

1. Something known to the individual, such as a password or an algorithm.
2. Something possessed by the individual, such as a card key.
3. Some distinguishing characteristic of the individual, essentially physiological or morphological measurements.

2. A METRIC FOR COMPARISON OF AUTHENTICATION MECHANISMS

The techniques being introduced will be illustrated using examples to include each of the above.

All authentication mechanisms deliver some type of information concerning the individual concerned, be this in relation to something known, possessed or characteristic of the user. While the nature of authentication parameters is extremely diverse, ranging from encryption keys through biometrics to passwords, a common property by which they may all be compared is their information content.

Information theory[18] calculates information in terms of the probability of a particular message or symbol in the finite set of all possible messages of the same type. Information is then measured in 'bits', taken as the (positive) logarithm to base 2 of the message or symbol probability,

$$-\log_2 p(i)$$

where p is the probability of message i.

It is often more appropriate to measure the average or expected amount of information for a message from a given message set. This measure is termed entropy, and is calculated as the weighted average of the information in all messages in the set,

$$H = -\sum_i p(i) \log_2 p(i).$$

[*] Now at Department of Computer Science, Birkbeck College, University of London, Malet Street, London, WC1E 7HX.

For any message set, H is maximal when all messages are equiprobable (and therefore carry equal information), reducing as the probabilities become more skewed.

Many message classes display statistical properties that indicate a relationship between successive messages or symbols within the set, such that the probability of y is increased following arrival of x.

Such characteristics are measurable in terms of the conditional ($p_i(j)$) probability or joint probability ($p(i,j)$) of the i,j pair. Since it is inevitable that a measurable amount of redundancy exists in a message system exhibiting statistical patterns, conditional entropy is measured as:

$$H_i(j) = -\sum_{i}\sum_{j} p(i,j)\log_2 p_i(j).$$

Information theory may be applied to communication channels where a degree of noise is present. Noise reduces the information rate of the channel by introducing the possibility of symbols being corrupted or misinterpreted as a result of the noise.

This uncertainty as to the correctness of received symbols is represented as the conditional probability of a given received symbol r being a corruption or misinterpretation of a different transmitted symbol s_i. The information delivered via a noisy channel is given by

$$I(S_iR_i) = \log_2 \frac{P_{ri}(s_i)}{P(s_i)}.$$

The destructive effect of noise in a channel is given by

$$H_r(s) = -\sum_{s}\sum_{r} p(s,r)\log_2 p_r(s).$$

The information transferred across a noisy channel is given by

$$I(sr) = H(s) - H_r(s)$$

where $H(s)$ is the information source entropy, that is the information transferred prior to the introduction of ambiguity through noise.

3. PASSWORD TECHNIQUES

Many commentators agree that the low cost and high availability of password-based mechanisms will result in the continued predominance of such systems for some time.[1,2,5,6,11] This, together with the unreliability of many contemporary password-based systems, requires that consideration be given to such systems.

There is considerable disagreement over the best technique for selecting password values. User-selected passwords tend to offer better human factors,[2] but are often highly predictable[23]. System selection of random character sequences for use of passwords tends to dramatically increase the key space of a given length password, since user selection tends to choose values from natural language. Natural-language selections are limited by the strong statistical characteristics and structure exhibited by language. Table 1 illustrates the difference between the range of randomly selected characters and natural-language words obtained by search of an English dictionary.[22]

For a natural language such as English, structural dependencies may exist at a variety of levels:[16]

0. Zero order, letters (grams) from the language alphabet occur randomly.
1. First-order, letters (grams) have the frequency expected for the language.
2. Second-order (diagram structure), each letter occurs with the expected probability given its immediate predecessor.
3. Third-order (trigram structure), each letter occurs with the expected frequency given the immediately preceding diagram.
4. Fourth order, words occur at the expected frequency.
5. Fifth order, words occur at the expected probability given the preceding word.

For English, the entropy per letter has been measured at approximately 2.3 bits per letter when 8 letters are known, down to approximately one bit per letter when 100 letters are known.[16] Given that the maximum entropy for a 26-letter alphabet is $\log_2 26 = 4.7$ bits per letter, it is clear that natural language is highly redundant, compared to a message set using the same alphabet but not restricted by the same statistical nature.

The controversy surrounding system versus user selection of passwords is resolved by the observation that the pertinent important property of a password is not necessarily its length and how it was chosen, but rather its information content. If the password value is a selection from natural language, as is likely for a user-selected password, then it is likely to be a combination of words, or at least to exhibit the expected diagram or individual symbol frequencies.

Taking an eight-character password as an example, a user-selected value may be expected to exhibit third-order (or above) characteristics, and to have a corresponding entropy in the order of

$$2.3 \times 8 = 18.4 \text{ bits},$$

while a randomly selected value would exhibit zero-order characteristics with a corresponding entropy in the order of

$$4.7 \times 8 = 37.6 \text{ bits}.$$

Additionally, the common password values often chosen by users have a high probability within the set of possible fourth-order values, and therefore a correspondingly low information content.

Alternatively, however, user-selected passwords may

Table 1. Expected number of words by word length in characters

	1	2	3	4	5	6	7	8	> 8
Dictionary	2	26	722	2166	3610	4765	4260	3861	8447
Random	26	676	17.5K	457K	11.8M	309M	8B	200B	—

where K = 1000, M = 1,000,000, B = 1,000,000,000 be pseudo-random, or exhibit reduced natural-language characteristics if carefully selected.[2]

Phonetic system-selected passwords have no meaning, but in selecting only pronounceable combinations of characters, they will tend to exhibit many second- and third-order characteristics of the language. Consequently, the entropy will lie between those of completely random values and values selected from language.

Further, passphrases[12] are appropriately measured by the same metric, the only significant difference being an increased tendency to exhibit characteristics of fifth order and above.

Given frequency tables for the various orders of characteristics expected in the English language, a given password may be analysed by scanning for occurrences of high order grams through low order grams, and accumulating an information count as grams are recognised.

Given the relatively low probability of randomly generating high-order grams from grams of a lower order (although a selection of n-grams may display $2n$-gram similarities), it is appropriate to search for high-order grams before searching for lower-order grams.

However, this approach is complicated by lack of knowledge concerning both the precise alphabet in use, and the gram-order characteristics displayed by the chosen password.

This approach can provide only an approximation to the true information content, since it would be impossible to be precise without knowledge of the symbol set from which the password was chosen. Further, higher-order probabilities may vary significantly depending on the language source studied, as exhibited by a comparison of the language of legal documents with that of the popular press.

For example, consider the case where a conscientious user attempts to select a 'random' password value, and password information content is being calculated using statistics obtained from natural language.

For a four-character random password (selected from the English alphabet), the probability of a certain letter being a given character is

$$1/26 = 0.0385$$

The probability for a given password is then

$$0.0385^4 = 2.197 \times 10^{-6}$$

In analysing this password, however, we do not know that it is supposed to be random and treat the password on the basis of the natural-language characteristics it displays. If this treatment includes expected character frequencies[22] then, for example, probabilities for selected 'random' passwords would be

$$\text{'NSRH'} = 0.071{*}0.0655{*}0.0613{*}0.0542 = 1.545{*}10^{-5}$$

$$\text{'VKJQ'} = 0.01{*}0.0066{*}0.0016{*}0.0011 = 1.1616{*}10^{-10}.$$

To avoid this, lower-order gram frequencies may be treated as if single-character selections were random, that is, zero-order frequencies used where first-, second- and third order frequencies would otherwise have been used.

Finally, it is necessary to consider the size of the alphabet from which the password has been selected, and usually this will be some subset of the characters supported by standard keyboards. For simplicity this may be considered as four sets of characters, comprised of upper-case alphabetic, lower-case alphabetic, numeric and all remaining non-alphanumeric characters (such as '!' or ESC). A given password may then be considered as having been chosen from the union of all sets that have a member character in the password.

The definition of character subsets above is useful in that the subsets have different characteristics:

(i) the alphabetic subsets may have been used to form natural-language words, in which case high-order gram characteristics may be recognised;
(ii) the numeric subset may be used to form numbers;
(iii) the remaining subset may be considered to include all non-alphanumeric characters, such that its size is not precisely defined, but may be taken as approximately for average keyboard character sets.

Characters from the password are used to accumulate information counts based upon the probability of the character within the character subset to which it belongs. Thus, alphabetic character sequences may be analysed for high order grams, while numeric digits and non-alphabetic may be treated as random. When the entire password has been analysed, the values accumulated may be weighted by considering the range of other character subsets that have also been used. Thus, the probability per character for passwords consisting entirely of alphabetic or numeric characters is 1/26 and 1/10 respectively. For a password including both alphabetic and numeric characters however, the probability per character would be treated as 1/36 as a result of the final weightings applied to characters for both subsets.

The above approach provides a convenient solution to the problem posed by mixed-case alphabetic characters. These are converted to a single-case representation for most operations within the algorithm, thus simplifying such operations as searches through tables of high-order grams. The final weighting stage redresses the effects of the conversion by treating upper and lower case as different character sets.

4. PERSONAL CRYPTOGRAPHIC AUTHENTICATORS

Personal authentication devices resembling pocket calculators have been produced by commercial concerns, and achieve authentication based on something possessed (i.e. the device) and something known (a Personal Identification Number or PIN).

These devices typically are primed with a key associated with a given user and known also to the host computer performing authentication. Further a PIN is known to the host, and must be correctly entered into the device each time the device is used.

The device calculates a value that is dependent upon:

(i) the device key
(ii) the users PIN
(iii) either the time or a pseudo-random number broadcast by the host and entered into the device by the user.

The function used to generate the authenticating value is usually an encipherment, so that the device key and PIN are computationally infeasible to derive given examples of the authenticating dialogue (although a dictionary attack may be possible over a prolonged period in the case of a pseudo-random number challenge). The same function is performed by the host and the result tested for equality with the value returned by the user.

These types of device are capable of generating large authentication values, using encryption algorithms such as DES. However, since the user is required to enter the generated value on to a terminal keyboard (and a challenge value into the device in some cases), the range of values used is typically of the order of a five- or six-digit decimal number.

The functions used to generate the authentication parameter tend to make the values generated appear random, so that the values are effectively one-time passwords and are difficult to predict by an attacker. This characteristic makes calculation of the authentication parameter strength straightforward, since the probability of each possible value may be treated as equal.

If the authentication parameter is a numeric quantity, then the power of the parameter is the entropy for randomly chosen values of the same order, e.g.

$$\text{for 5 decimal digits, } 5 \times 3.32 = 16.6 \text{ bits}$$
$$\text{for 6 decimal digits, } 6 \times 3.32 = 19.92 \text{ bits}$$

5. BIOMETRIC AUTHENTICATION TECHNIQUES

Biometric authentication techniques operate by measuring some physiological or morphological characteristic of the user. The characteristic being measured will generally be such that wide interpersonal variations exist, and include such characteristics as hand geometry, fingerprints and signatures.

Authentication is performed by requiring the individual to submit to the test involved, and comparing the result of the test against a previously recorded profile for that individual. The profile may either be stored centrally, or alternatively stored on a security token presented at the time of authentication.

Biometric authentication techniques must cater for the variations that naturally occur in the characteristics of every individual. In the same way that body weight is changeable as a result of such eventualities as exercise, illness and eating, most other measurable attributes will also tend to vary. Further, some mechanisms experience practical difficulties in achieving accurate measurements, for example finger print images may be deformed due to tissue pliability. Consequently, biometric techniques are required to operate within tolerances, and hence an element of uncertainty is often introduced into the authentication procedure.

The effectiveness of a biometric technique in authenticating identity for a known user population may be determined by conducting a series of trials.[20] In some cases, it may be that a certain number of authentication checks will fail, either accepting an impersonator or denying the authentic individual. The percentage of such failures are termed the Imposter Pass Rate (IPR) and False Alarm Rate (FAR) respectively. While it may be possible to adjust the tolerances of the biometric mechanism to reduce or eradicate one type of failure, this will tend to increase the incidence of the other type of failure, and a compromise tolerance may often be chosen.

In considering the application of information theory to biometric authentication, several observations are made.

Firstly, the biometric mechanism may require tuning to achieve the desired (or compromise) IPR and FAR. The need for tuning and the success of tuning may be determined from performance statistics for the mechanism.

Secondly, information is delivered by the measurement process, irrespective of how that information is encoded. To quantify the amount of information delivered, it is necessary to obtain statistics covering the defined user population in respect of the characteristic being measured. These statistics determine the probability of the measured characteristic falling within defined ranges and hence the information delivered by a measurement within that range. Thus a relatively probable measurement does not constitute a particularly distinguishing characteristics. Intuitively, this is what we expect, considering the usefulness of such personal descriptions as 'of average height and medium build'.

Thirdly, many physiological and morphological characteristics may exhibit relationships between each other, so that small and slender will generally weigh less than small and well-built. Thus, conditional probabilities may exist between different characteristics, and clearly this effects the information delivered by techniques measuring more than one characteristic.

Finally, since a biometric mechanism may produce a range of values for a given individual, an additional level of uncertainty is introduced. A series of test results may be taken to determine the range and probability of measurements expected for each individual user. It may be necessary to limit the range of acceptable values for a user in order to limit the IPR of the biometric mechanism, but the FAR for the individual will clearly increase as a result (indeed, the sum of the probabilities of the excluded measurements and the observed FAR for any individual user should correlate). The element of uncertainty for any given measurement may be treated as noise effecting the original quantity of information delivered by a particular measurement.

Thus, the amount of information produced by a particular measurement is initially based upon the probability of that measurement for the entire user population, but this quantity is then considered to be reduced by a level of noise based on the probability of the measurement for the individual user in question.

6. VULNERABILITY OF AUTHENTICATION MECHANISMS

The risk of compromise to an authentication mechanism is dependent upon the nature of mechanism and the precise conditions under which it is used. Since it is not reasonable to generalise on the conditions of use, exposure must be derived by some method that recognises local considerations.

Vulnerability may be assessed using a risk-analysis methodology.[21] However, the objective of the risk-analysis exercise is not to calculate the risk of compromise in terms of fiscal or any other units, although such evaluation may be used to identify the criticality of computing services and hence the need for reliable authentication of users. Rather, the objective is to determine the frequency with which the authentication mechanism may be expected to suffer compromise, or to become unacceptably susceptible to compromise. This will often be dependent upon such criteria as data-storage facilities and frequency of use.

Certain authentication mechanisms are extremely vulnerable to compromise through disclosure of some secret value. Notably, password systems may be compromised by exposure of the password through user carelessness, illicit storage at point of entry, during transmission or in the host computer storage. Techniques such as link encryption and storage of the password as a one-way encrypted value are available to reduce the vulnerability of these mechanisms.

Personal authentication devices, as introduced earlier, may provide an information quantity lower than that of a password, but are typically vastly superior in terms of their reduced vulnerability. Typically, such devices do not expose all critical data at time of entry, since the value transmitted is a value dependent on secret quantities that are not exposed and which are computationally difficult to determine. Some of these devices are exposed to a dictionary attack, notably those where a random challenge value is input to a transformation producing the same value on each occasion. Further, some mechanisms are well protected against compromise of physical security at the host location by using a tamper-resistant peripheral module for storing all critical data and performing authentication checks.

Biometric techniques are also vulnerable to compromise. For example, fingerprint scanning mechanism may be deceived by presenting a moulding of the correct fingerprint to the scanning device, and voice recognition techniques deceived by mimicry or by replaying a recording of the correct voice.

The assessment of vulnerability requires consideration of the specific local conditions in question. Risk-analysis methadologies provide appropriate means of achieving such assessment, but the results produced may often be somewhat subjective. For example, many contemporary password mechanisms are highly vulnerable to compromise through exposure of pasword values at point of entry, during transmission or at the host system. Although a strictly qualitative approach may discriminate heavily against such systems, a less stringent approach may often be found acceptable.

Clearly the risk of compromise of an authentication mechanism introduces uncertainty as to the correctness of user identities verified using the mechanism. Since the authentication mechanism in question may be characterised by the quantity of information it produces, the uncertainty introduced by the vulnerability of the mechanism to compromise may be expressed as the effect of noise on the original quantity of information produced.

Since the vulnerability of the mechanism may increase with both time and use of the mechanism, the quantity of noise required to represent vulnerability may also increase with time and use. This quantity will be referred to as the 'exposure' of the mechanism or authentication data.

7. EXPLOITATION IN USER ACCESS CONTROL MECHANISMS

Having identified the necessary techniques for quantifying the results of authentication mechanisms and the risks of compromise to such mechanisms, it is possible to consider the advantages of exploiting these techniques in User Access Control mechanisms.

Many User Access Control mechanisms depend upon human administration functions and user conscientiousness in order to retain reliability. Further, the requirement for certainty of correct identification may vary with the sensitivity and privilege of the service being accessed, and the risks of compromise may vary with such conditions as the telecommunication services used on a particular occasion.

The quantity of information delivered by an authentication technique, henceforth referred to as 'strength', and exposure values may be evaluated by a programmed system. Further, strength and exposure values may be stored and retrieved as necessary, such that exposure values may be incremented with time or usage as desired.

The points at which evaluation of strength and exposure may occur may be, for example, any of the following:

(i) When an access request is received, to calculate a minimum strength value to be achieved for the request to be authorised.

(ii) In selecting authentication methods, to calculate a strength value for that method. This algorithm may be evaluated for each of several alternative authentication methods available, and may consider, for example, the characteristics of the authentication parameter, and the exposures of the authentication parameter and communication channel.

(iii) Following a successful authentication check to calculate the exposure to the authentication parameter. This value may then be added to the total exposure to date.

(iv) Following an unsuccessful authentication check, to calculate the exposure to the authentication parameter. This value may then be added to the total exposure to date.

The fundamental concept introduced is that a programmed system may use the strength and exposure values to control the vulnerability of the authentication mechanisms it exploits, and may include mandates concerning authentication strength and exposure in access authorisation rules.

For example, consider connection to a service with a classification of between one and nine, in decreasing order of sensitivity.

Available strength might be calculated as:

$$\text{available strength} = (S-(S^*(1/C)))-(E/C)$$

where S = strength, $S>0$, C = classification, $0<C<10$, E = exposure, $0 \leqslant E \leqslant S$.

This excludes the technique in question from use for access to the highest classification, with selection always dependent upon classification and in certain circumstances upon previous exposure also.

Supposing that a risk analysis exercise indicates that a password should not be used more than n times. This may be enforced by the following available strength evaluation,

$$\text{available strength} = \text{strength} - \text{exposure}$$

plus the following after each use of the authentication mechanism,

$$\text{exposure} = \text{exposure} + (\text{strength}/n).$$

Finally, notice that calculation of strength offers a suitable criteria for managing an authentication mechanism than has conventionally been applied. By comparison, forcing password change after a set time period will generally be a highly arbitrary approach.

8. AN EXTENDED SYSTEM MODEL FOR USER ACCESS CONTROL SYSTEM

This model is concerned with the access control function up to the point at which a user or service is authorised for access to a specific service. The model incorporates the concept that access request authorisation be subject to the quality of identity authentication, and this accounts for the essential difference between this and other models. Requirements of the model are that it considers:

1. The representation of objects, specifically protection domains (system users) and computing services.
2. The expression of access rights to a service object available with a protection domain.
3. The requirement for adequately identifying and authenticating the system user.

Generalised models of access control systems have been described elsewhere, as summarised below.

Access control matrix models represent a protection system by a triple (S, O, P), where

O is the set of protected objects

$S \subseteq O$, the set of subjects requiring access to objects

P is an access matrix.

Additionally, there is the set R of access rights, $r \in R$, and a set of operations for defining and modifying the matrix.

The entry $P[s_i,o_i]$ contains the access rights $P[s_i,o_i] \subseteq R$ of subjects s_i (or to any subject operating in the protection domain s_i) to object o_i.

Refinements of the model allow provisions to be made for:

(a) copy flags, denoted as r^*, signifying that the right r may be copied to other domains[10]
(b) ownership, signifying that an object is owned by a given domain, expressed as a right within the matrix entry[8]
(c) control, signifying that a domain has control over another domain (domains being represented as objects)[8]

In the context of this model, a monitor is defined as the protection mechanism responsible for authorising access (S,r,O) if r is in $P[S,O]$.[8]

However, the general matrix model is not entirely suitable for modelling a sophisticated UAC system for the following reasons.

1. The matrix model stipulates that each subject must have a unique, unforgeable identity,[10] or that identity be established at least 'to within any reasonable doubt'.[8] In a UAC mechanism identity is indeed subject to doubt, since authentication mechanisms are fallible, and this consideration should be represented in the system model.
2. Since the objects to be represented in a UAC system model are an arbitrary set of computing services, each providing an arbitrary set of facilities, it is unreasonable to assume that access rights or objects should be generic.[7] While it may be possible to model arbitrary access rights using sets of generic rights, this would lead to overcomplication of the model to the extent that its use would become burdensome.

An alternative to the matrix model is the formulary model.[9] This model claims flexibility beyond that available in matrix models, since the authorisation process and the definition of object types is performed by programs, or 'formularies', at run time. The model does not limit the number of formularies that may exist, and although certain functions and relationships of individual components of a formulary are defined, the model permits the definition of arbitrarily complex decision rules and object types.

The formulary model is essentially concerned with mediation in database access requests. Consequently it provides many functions, such as name binding, that are not required in a pure access control system. Moreover, it does not include authentication mechanisms, which must presumably be providing the 'control' procedure within each formulary.

The principal component of the extended model is an access matrix P, such that the state of the system is the conventional (S,O,P) triple.

Additionally, there exists a set F of service access functions, $f \in F$, and a set A_s of authentication functions for each user $(s \in S)$ of the system.

Service access functions are similar to access control procedures[10] or formularies, and provide contextual mapping for a given service object $o \in O$, and indeed may modify the access request in enforcement of specialised security policies. Additionally, each f interprets a user's access request and returns a value indicating the minimum level of authentication required for access to be authorised. Thus there exists a one to one mapping of F onto O.

Each function $a \in A$ performs user authentication, returning a value dependent on

(a) the authentication technique employed,
(b) the properties of the authentication parameters used (including its value, previous use and risk of compromise).

These functions model suspension of a user id for some period following unauthorised access requests or failed authentication checks.

The access rights represented in the model are:

1. $a \in P[s,o]$. The right to invoke the service access function for service o.
2. $c \in P[s,o]$. The right to insert or delete access right 'a' in to entry $P[s,o]$, for the o in $P[s,o]$ and all $s \in S$, granting other users the right to invoke the service access function for that o. This is preferred to copy flags since issues of authorisation of rights transfer are simplified.
3. $x \in P[s,m]$. The right to insert or delete access rights 'c' and x from $P[s,o]$, for all s and all o, representing the system administrators ability to devolve respon-

sibility to the principal for a particular service $o \in O$. This allows centralisation of certain administrative functions.

Notice that a is not implied by c, and that c is not implied by x. The right to administrate the system does not imply the right to use the protected resource, although the latter right may be granted by exercising the former right (it would be possible to prevent this by insisting that $c \in P[s,o]$ when checking for $a \in P[s,o]$.

The mechanics of the commands for changing the state of the matrix are essentially the same as in conventional matrix models[7,8] but are dependent upon the rights 'c' and 'x' as above. They are not repeated here.

Access authorisation rules recognising dependency on identity verification may then be derived from the following example, generally by including further stipulations concerning the presence of certain rights in an access matrix entry.

Let

$S = \{\text{subjects}\}$, $s \in S$, the system users

$O = \{\text{objects}\}$, $o \in O$, the services being protected

$R = \{a,c,x\}$ = access rights, $r \in R$

V = supplied connection parameters for a particular access request

fo = a function associated with $o \in O$, returning a value $n, 1 < n < m+1$, for all o

A_s = a set of authentication functions associated with $s \in S$, where each $a \in A_s$ returns a value n such that $O < n < m$

P = an access control matrix (S,O,P) = *a state of the UAC system*

An access (s,o,V) is authorised if, and only if,

1. $A'_s = A_s \cup \{a\}$ for all $a \in A^A_s$, if $a \in P[s,o]$ and
2. $\sum_{i=1}^{i < n\{A'_s\}} a_i(s) \geqslant f_o(s,V)$

Absolute denial of access is a capability of all a_i and all f_i. Observe that no a_i can satisfy $M+1 = f_o(s,V)$, further that $a_i = 0$ can never satisfy $f_o(s,V)$.

9. CONCLUSION

The techniques that have been described may be incorporated in programmed logical User Access Control Systems.

The techniques are general, being applicable to a wide range of authentication mechanisms, and allowing an area of considerable vulnerability in contemporary User Access Control systems to be expressed and thus controlled in an automated fashion.

REFERENCES

1. A. Berman, What to know about effective password protection. *Canadian Datasystems*, pp. 60–63 (Nov. 1984).
2. B. F. Barton and M. S. Barton, User-friendly password methods for computer-mediated information systems. *Computers and Security*, pp. 186–195. (1984).
3. J. Koehring, Automatic identity verification. *Information Age* **6** (2), pp. 103–110. (April 1984).
4. C. C. Wood, Effective information system security with password controls. *Computers and Security* **2**, pp. 5–10. (1983).
5. J. A. Schweitzer, Computer security: make your passwords more effective *EDPACS*, pp. 6–11. (Feb. 1983).
6. J. A. Haskett, Pass algorithms: a user validation scheme based on knowledge of secret algorithms. *Communications of the ACM* **27** (8), 777–781. (Aug. 1984).
7. M. A. Harrison, W. L. Ruzzo and J. D. Ullman, Protection in Operating systems. *Communications of the ACM* **19** (8), 461–471. (Aug. 1976).
8. G. S. Graham and P. J. Denning, Protection – Principles and Practice. Spring Joint Computer Conference, **40**, AFIPS Press, Montvale, NJ (1972).
9. L. J. Hoffman, The Formulary Model for Flexible Privacy and Access Controls. Joint Computer Conference, **39**, AFIPS Press, Montvale, NJ (1971).
10. B. W. Lampson, *Protection*. ACM Oper. Syst. Rev., **8** (1), 18–24. (Jan 1974).
11. Helen M. Wood, The use of passwords for controlling access to remote computer systems and services. National Computer Conference pp. 27–33. (1977).
12. Sigmund N. Porter, A password Extension for Improved Human Factors. *Computers and Security* **1**, 54–56. (1982).
13. Michael B. Wood, *Computer Access Control*. NCC Publications (1985).
14. H. M. Deitel, *An Introduction to Operating Systems*. Addison-Wesley, Reading, MA (1984).
15. E. B. Fernandez, R. C. Summers, and C. Wood, *Database Security and Integrity*. Addison-Wesley, Reading MA, (1981).
16. C. E. Shannon, Prediction and Entropy of Printed English. *Bell System Technical Journal* **30**, 50–64. (Jan. 1951).
17. L. Lamport, Password Authentication with Insecure Communication. *Communications of the ACM* **24**, (11), 770–772. (Nov. 1981).
18. C. E. Shannon, A Mathematical Theory of Communication, *Bell System Technical Journal* **27**, (3), pp. 379–423. (July 1948).
19. United States Department of Defense, *Department of Defense Trusted Computer System Evaluation Criteria*. Technical Report, CSC-STD-001-83. Computer Security Centre, Fort George G. Meade, Maryland, U.S.A.
20. Federal Information Processing Standards Publication 48, National Bureau of Standards. *Guidelines on Evaluation of Techniques for Automated Personal Identification*.
21. Federal Information Processing Standards Publication 65, National Bureau of Standards. *Guideline for Automatic Data Processing Risk Analysis*.
22. C. C. Foster, *Cryptanalysis for Microcomputers*. Hayden, Rochelle Park, NJ (1982).
23. R. Morris and K. Thomson, Password Security: A Case History. *Communications of the ACM* **22** (11), pp. 594–597. (November 1979).