On the Use of Negative Selection in an Artificial Immune System

Marc Ebner, Hans-Georg Breunig and Jürgen Albert
Universität Würzburg, Lehrstuhl für Informatik II,
Am Hubland, 97074 Würzburg, Germany
ebner@informatik.uni-wuerzburg.de, Tel. (+49)931/888-6612
`http://www2.informatik.uni-wuerzburg.de/staff/ebner/welcome.html`

Abstract

The natural immune system is very effective at protecting the body from diseases. Several researchers have analyzed the natural system and created artificial systems which copy mechanisms of the natural system in order to improve computer security. We suggest that the negative selection algorithm, which is at work in the natural system, might have been copied too closely. We argue against the use of negative selection if space is finite and self comprises only a small fraction of the available space or if space is infinite. We illustrate this on the problem of user authentication using keystroke analysis.

1 MOTIVATION

The natural immune systems' task is to detect molecules which don't belong to the organism. This ability led several researchers to look closely at the workings of the natural immune system. Inspired by the natural system they have tried to copy mechanisms which are at work in the natural system more or less closely for use in the area of computer security (D'haeseleer et al. 1996; Forrest et al. 1997; Forrest et al. 1996; Forrest et al. 1994; Hofmeyr and Forrest 1999a; Hofmeyr and Forrest 1999b; Kephart 1994; Kim and Bentley 2001a; Kim and Bentley 2001b; Somayaji et al. 1998). After having a closer look on how the natural immune system works, we briefly review some of the artificial immune systems and analyze the advantages and disadvantages of using the mechanism of negative selection in an artificial immune system.

2 THE NATURAL IMMUNE SYSTEM

Our discussion of the natural immune system is based on Alberts et al. (1994). A substance causing an immune reaction is called an antigen. The immune system is capable of distinguishing between highly similar antigens. Even proteins which differ by only a single amino acid can be distinguished. The cells which are responsible for the immune specificity are called lymphocytes. They belong to the class of white blood cells. The human body has approximately $2 \cdot 10^{12}$ lymphocytes. Two classes of lymphocytes exist: B-cells and T-cells. B-cells develop in the adult bone marrow or the fetal liver. They produce antibodies. T-cells develop in the thymus and are responsible for the so called cell-mediated immune response.

The immune system is based on a mechanism which is called clonal selection. Each lymphocyte is equipped with a receptor which can be used to bind an antigen. The term clonal selection comes form the fact that a large variety of receptors exist which can be grouped into families, or clones, of cell. Each receptor has a specific shape and can only react with a certain antigen. The receptors are generated at random and are thought to cover the whole space of possible antigens. If a lymphocyte binds an antigen, then the cell becomes activated. The cell proliferates, matures, and finally secretes antibodies. The antibodies have the same shape as the receptor of the cell which secreted it.

The antibody response includes the production of antibodies which circulate through the blood and other body fluids. The antibodies consist of a Y-shaped molecule which can bind an antigen at two locations. An abstract representation of this Y-shaped molecule and a close-up of the antigen-binding site of an antibody molecule is shown in Figure 1. The antibodies

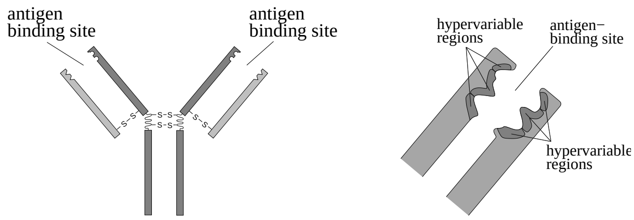


Figure 1: Antibody (left). Close-up of the antigen-binding site of an antibody molecule (right). Redrawn from Alberts et al. (1994).

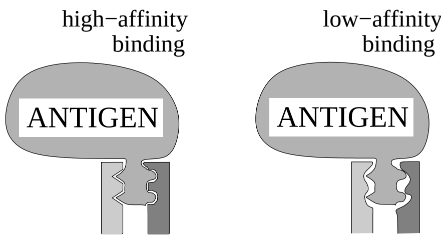


Figure 2: Binding of an antigen. Redrawn from Alberts et al. (1994).

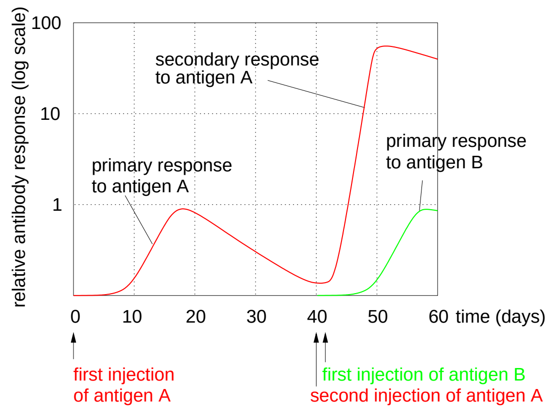


Figure 3: Response to antibodies. The response to the second exposure to antigen A happens much faster than the response to the first exposure. In addition the response is also stronger. Redrawn from Alberts et al. (1994).

bind antigens which fit into the receptors. Through this binding process a virus may be inactivated. Antigens coated with antibodies may also be digested or killed by special cells.

In the course of an immune response B-cells increase the affinity of the antibodies they produce. This process is called affinity maturation. Changes to the shape of the receptors are caused by mutations. This process is referred to as somatic hypermutation. The mutations happen with a frequency which is approximately one million times higher than the mutations which happen to the other genes. Cells which have a high affinity binding reproduce better because they can more easily dock on an antigen (Figure 2). This results in a selection of those cells which closely match the given antigen. Thus, an evolutionary process is embedded in the immune system which produces highly specific antibodies to any possible antigen.

The cell-mediated immune response consists of the production of specialized cells, called T-cells. These cells are used to detect cells which have been infected by a virus. Peptide fragments of a foreign molecule are brought to the cells surface by specialized molecules. Inside the cell those molecules are invisible to the immune system. Once these fragments show up on the cells surface they can be detected by the T-cells. We have two kinds of T-cells: cytotoxic T-cells and helper T-cells. Cytotoxic T-cells kill infected cell directly while helper T-cells activate other cells who then kill the infected cells. In addition, the helper T-cells are necessary for the activation of B-cells. Helper T-cells stimulate themselves as well as other helper T-cells to reproduce once they are activated. Only those helper T-cells become activated which have detected an antigen.

In addition to the immune responses the immune system also has a memory. If an antigen is detected for the first time then the immune response only happens after a certain delay. This is called the primary immune response. The immune response on a known antigen, called the secondary immune response, happens quicker and more strongly in comparison to the primary immune response. The difference between the primary and the secondary immune response is shown in Figure 3. This behavior of the immune system is realized through different stages of the T- and B-cells. There are at least three different stages: virgin cells, activated cells and memory cells. Activated cells die after a few days. However, memory cells can live for several months or even years.

The main task of the natural immune system is to distinguish between own molecules and molecules belonging to a foreign organism. Detecting foreign molecules is mainly the task of the T-cells. The T-cells develop in the thymus. Cells which bind to own peptide are eliminated during development. This process is called negative selection. Only T-cells which have a low affinity to the organisms own molecules remain. B-cells need helper T-cells to react to foreign antigen. Therefore, helper T-cells also ensure that self-active B-cells are harmless.

Following this discussion of the human immune system we now have a look at how the workings of the natural system have been mapped to an artificial immune system.

3 PROPOSALS FOR AN ARTIFICIAL IMMUNE SYSTEM

In the area of computer security one needs to distinguish original data from manipulated data, authorized users from intruders and normal behavior from abnormal behavior. This is exactly the problem the natural system solves, namely to detect self from non-self. A number of properties of the natural system would also be useful for an artificial system: (a) distributed detection, (b) multi-layered, (c) diversity, i.e. every individual has its own immune system, (d) disposability, no single component is essential (e) the immune system can work autonomously, (f) is adaptive and (g) does not depend on secrets (Somayaji et al. 1998).

Kephart (1994) developed a biologically inspired immune system to protect a computer system from previously unencountered viruses or worms. The analogies between the natural system and the artificial system are rather loose. Integrity monitors in conjunction with activity monitors are used to determine if a virus or worm has entered the system. The integrity monitors check if files have been changed or added. Activity monitors check for dynamic behavior which is typical of viruses. They also look at the type of change to see if the change may have been caused by a virus. If it is determined that a virus has entered the system a scan is made to find any known viruses. In case the virus is known, it is removed. Otherwise, decoy programs are used in order to get a sample of the virus. This has been likened to the ingestion of antigen by macrophages or B cells in the natural immune system.

Forrest et al. (1994) developed an artificial immune system for change detection in executables. The system learns to distinguish the original version of a program from a program which has possibly been infected by a virus. Forrest et al. generate a set of random detectors (bit strings) in analogy to the workings of the thymus of the natural immune system. The negative selection algorithm is used to remove those detectors which would detect the original programs. The feasibility of generating detectors was analyzed by D'haeseleer et al. (1996) who also proposed a more efficient algorithm for generating detectors.

Hofmeyr and Forrest (1999a, 1999b) developed an artificial immune system for intrusion detection. This system has a closer analogy to the workings of the natural immune system. The system's task is to distinguish normal from abnormal connections between two computers in a local area network. The system basically consists of a set of detectors which are used to detect non-self, i.e. abnormal behavior. Initially the detectors are generated at random. During an initial maturation period, it is checked if a detector matches any part of the system which is to be protected. If a match occurs then the detector is deleted. If a detector has survived this process for a specified number of steps then the detector matures and is now ready to detect non-self. The set of mature detectors continually monitor the data stream for non-self. If a detector is not activated for some time then the detector is deleted and replaced with a new mature detector leaving the negative selection algorithm. Detectors are memorized if a detector receives a special type of co-stimulation. Detectors which have been memorized previously can immediately detect non-self.

Forrest et al. (1996) have developed an artificial immune system which monitors dynamic behavior of processes. Sequences of system calls are used to define normal behavior for standard unix programs. Deviations from this normal behavior are detected by comparing short sequences of system calls with normal sequences stored in a database. Plans to extend this work include the addition of the negative selection algorithm and using on-line learning.

Kim and Bentley (2001b) modeled clonal selection for use in an artificial immune system for network intrusion detection. Basically, Kim and Bentley evolve detectors which detect non-self antigens. A negative selection operator is embedded in this process. Detectors which match self antigens are deleted. Other authors have used principles from artificial immune systems for diversity maintenance in multi objective optimization (Cui et al. 2001), to improve adaptability in the context of time dependent optimization (Gaspar and Collard 1999) or to recognize spectra for chemical analysis (Dasgupta et al. 1999).

Many of the artificial systems stay very close to the workings of the natural system, with all its advantages and disadvantages. We now have a closer look at the efficiency of the negative selection algorithm.

4 ON THE EFFICIENCY OF THE NEGATIVE SELECTION ALGORITHM

Recently, Kim and Bentley (2001a) analyzed negative selection in an artificial immune system for intrusion detection. Their main result is that as the task be-

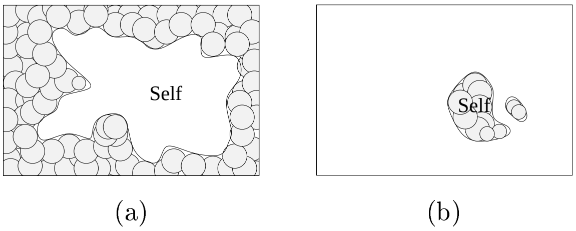


Figure 4: (a) Covering non-self with detectors works best if self is large and non-self occupies only a small fraction of space. (b) On the other hand, covering self with detectors works best if self is small and non-self occupies a large fraction of total space.

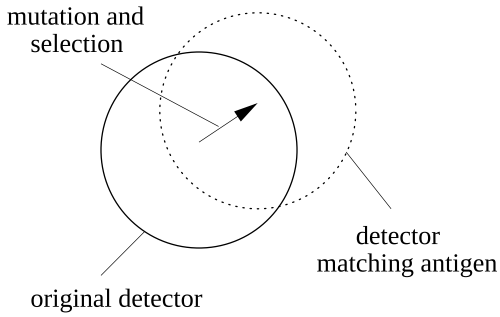


Figure 5: Somatic hypermutation and clonal selection increases the affinity between detector and antigen. This corresponds to a movement in shape space.

comes more complex, the number of detectors has to be unacceptably large and the time needed to generate a sufficient number of detectors is impractical. In comparison to this critique of the negative selection algorithm our critique is much more general.

For this analysis we assume that we have n-dimensional real valued vectors which represent antigens and detectors. We also assume, that we have some mechanism which is able to detect if a match occurred between a detector and an antigen. This can be some arbitrary measure such as correlation coefficient or distance between the two vectors. Now we need a definition of self. We define self as a subset of points in the n-dimensional space. The set of points describing self does not have to be static but can vary over time. A threshold is usually used to determine if a detector matches either an antigen or any point of the self. This fact is modeled by placing a hyper-sphere around each point which belong to the self. The radius of the hyper-sphere determines the sensitivity of the detector. An antigen is detected if it lies inside the hyper-sphere of a detector.

The negative selection algorithm distributes detectors randomly over this space. Detectors overlapping any points of self are removed (Figure 4a). This algorithm works fine if space is finite and self occupies a large fraction of the total space. But what if space is finite and self comprises only a small fraction of the available space or what if space is infinite? In this case it makes more sense to describe only the self and then check if an antigen falls outside of this area (Figure 4b).

Note that if the number of detectors is finite then learning a concept or learning its negation are not equivalent. If self can be covered using a smaller number of detectors in comparison to non-self then it makes more sense to detect self instead of non-self. If space is infinite and one tries to cover all points belonging to non-self, only a finite number of points will be covered. The sensitivity of the detectors need to be reduced if one wants to be able to detect all possible antigens. Reducing the sensitivity means enlarging the radius of the hyper-spheres. However this also means that during random generation of a detector it is very likely that this detector will cover some part of self and therefore will be removed from the set of detectors. If the size of self is small and the number of detectors is limited then self can be much better approximated by placing the detectors inside of self.

Suppose we want to determine if a point is located outside of a square. All we need is to check if the point is *not* located *inside* the square. This negation of a test can be done in a computer system quite easily but is very difficult to realize in a natural system. For the natural immune system it is simply not possibly to check if a given molecule is equivalent to one of its own molecules. The natural system would have to store samples of the molecules which occur in the human body in some part of the body and then check if the intruder falls *outside* this class of molecules. That is, the natural system would have to check that the given molecule is unlike any of the stored molecules. Because this is infeasible, the natural system is dependent on the use of negative selection to detect molecules which don't belong to its own organism. This works because local shape space is only finite (Kauffman 1993).

The natural system generates detectors which match other molecules rather crudely. The match is then refined using somatic hypermutation. Detectors which have a better match produce more offspring. What analog would there be in an artificial immune system for the process of somatic hypermutation? In our model, this would correspond to moving the detector in the direction of the position of the antigen (Figure 5). But is this really necessary? If it is established that some antigen is present in the system then it would be

character	duration	delay	character	duration	delay	character

Figure 6: Molecule generated from the user's keystrokes. Each molecule is composed of 7 fields.

the best to store the position of the antigen as a sample of an unknown intruder. The more samples are gained the better our knowledge of the intruder becomes.

So far, our discussion was rather general with no particular problem in mind. We now have a look at solving the problem of user authentication with an artificial immune system.

5 A PRACTICAL EXAMPLE

Biometrics such as a fingerprint or iris pattern may be used to determine who is actually using a computer system. For instance, Klosterman and Ganger (2000) have developed a system for continuous user authentication using a face recognition system. Other examples for user authentication include the analysis of keystroke patterns (Bleha et al. 1990; Brown and Rogers 1993; Furnell et al. 1996; Furnell et al. 1995; Joyce and Gupta 1990; Leggett and Williams 1988; Monrose et al. 1999; Monrose and Rubin 1997; Obaidat and Sadoun 1997; Robinson et al. 1998; Shepherd 1995; Song et al. 1997; Umphress and Williams 1985). As a practical example, we have chosen to analyze typing patterns. We want to make sure that the same user is continually sitting in front of the keyboard. For instance, if a person goes to lunch and forgets to lock the screen, then somebody not authorized could use the terminal. Keystroke analysis could also be used to make sure that even if the password is known to an intruder it can only be used to gain access to the system if the speed of typing corresponds to the authorized user. That is, we want to develop an artificial immune system for user authentication.

The system tries to determine if the same or a different user is using the system. This information is derived by analyzing the user's keystrokes. The duration of the key presses and the delay between key presses is used to determine who is typing on the keyboard. A stream of molecules is generated from the timestamps which are recorded whenever a key is either pressed or released. Each molecule contains data from three successive key release events. The data is stored in seven fields as shown in Figure 6. The first field contains the first character which was released. The second field contains the duration of the key press. The third field contains the delay between the release time and the time of the next key press. The fourth field contains the second character. The fifth field contains the duration of the second key press. The sixth field contains the delay between the release time and the time of the third key press and finally the last field contains the third character pressed.

First we need a notion of self. Self is defined as the normal typing behavior of a user. That is, all points in our 7 dimensional space which describe the typing behavior of the user belong to the set of self. All others belong to the set of non-self. In order to detect a different user we could randomly generate detectors and then check if the detectors match any of the molecules. If a detector matches any part of self then we delete this detector. The problem with this approach is how can we cover an infinite space? To solve this problem we have chosen to store samples of the normal typing behavior of the user and to check the stream of molecules against this sample. Thus, we do not use the negative selection algorithm.

For our experiments we have used a pool of 2000 detectors. Each molecule is stored in the pool of detectors after a delay of 2000 iterations of our algorithm. An activity level models the clonal reproduction of the artificial immune system. The stream of molecules is checked against this pool of detectors. If a match is made between a molecule and a detector, i.e. the molecule is sufficiently similar to one of the detectors, we decrease the activity level by 1% otherwise we increase the activity level by 1%. The activity level is set to 1.0 at the start of our algorithm. The activity can reach a maximum of 2.0. If the activity level reaches 1.5 (half-way between maximum and minimum values) then we assume that a different user has gained unauthorized access to the keyboard.

For a match between a molecule and a detector to be made we require that fields one, four and seven are equivalent. If these three fields are equivalent, then we look at fields two, three, five and six to determine how similar these fields are. For each of these four fields we calculate the following similarity measure

$$\text{similarity} = \sum_{i\in\{2,3,5,6\}} e^{-\frac{(a_i-d_i)^2}{\sigma_i^2}}$$

where a_i and d_i refer to the i-th element of the molecule (a possible antigen) respectively detector and $\sigma_i = d_i/2.1$. A match is made if the similarity measure reaches a value of 3.2 or higher.

We obtained typing characteristics for 5 different users. Each user had to type the first two sections of the seminal paper "Computing machinery and intelligence" by

Table 2: Number of keystrokes until change is detected.

User	1	2	3	4	5
1	-	141	100	139	267
2	39	-	67	459	46
3	53	52	-	71	54
4	123	322	209	-	113
5	120	101	131	119	-

Turing (1950). The data obtained from the keystroke timestamps was saved and converted into a stream of molecules. The stream of molecules was then analyzed offline. Each stream consisted of between 5168 and 5741 molecules. For each user we randomly selected a reading position from which we start reading molecules. After 2000 molecules we start reading molecules from the second users stream. Each user's stream of molecules was compared against the streams of all other users resulting in a 5×5 matrix of activity graphs. If a user's stream was compared against its own stream then we simply skipped 100 molecules after 2000 molecules have been processed. The results of all experiments are shown in Table 1. Table 2 lists the number of molecules (or keystrokes) until non-self was detected.

6 DISCUSSION

The results show that non-self is detected after a relatively small number of keystrokes. Simply storing samples of the user's typing behavior works well for the task of user authentication. However, this is not the only way to address this problem. Another possibility would be to average data and thereby to establish a model of the person sitting in front of the keyboard. In this case, the task is to obtain a compressed form of the data. For the task described here, this can be achieved by calculating the mean and the variance of the duration of keystrokes and the delay between keystrokes.

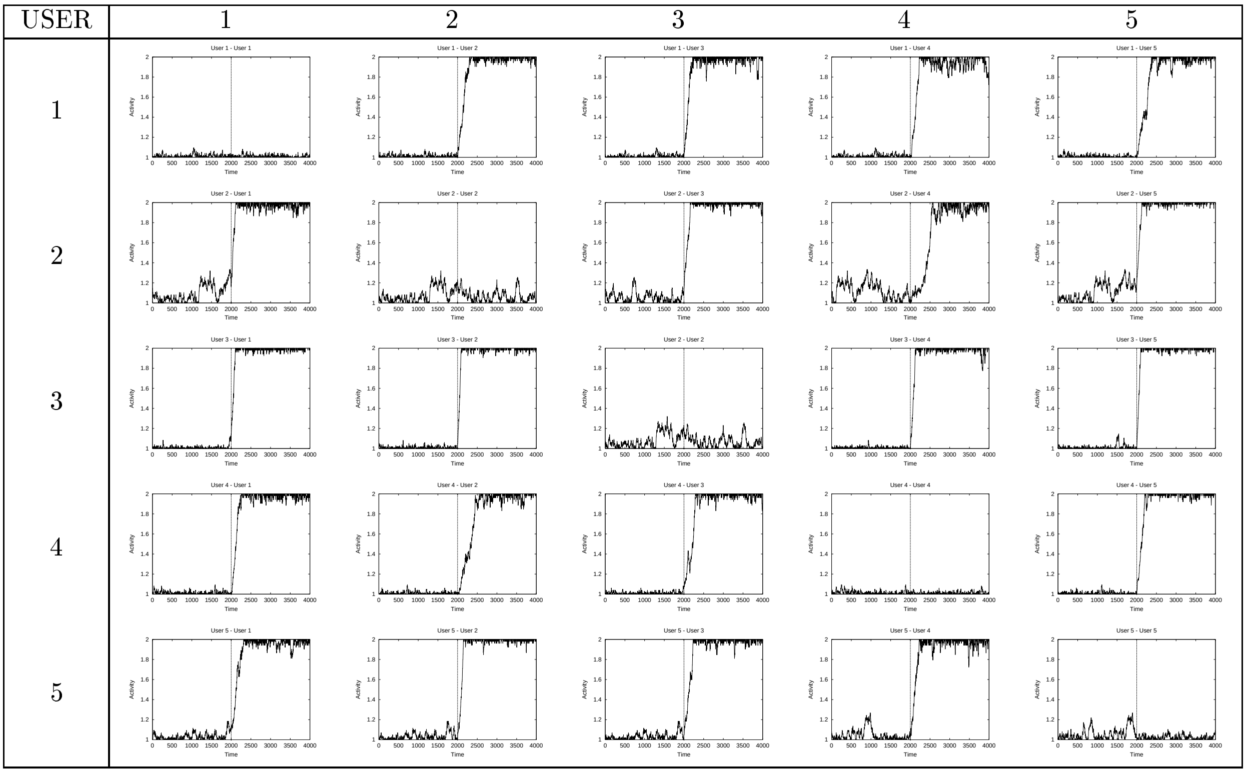
In fact, deriving a model from keystroke characteristics for user authentication was proposed by several researchers (Bleha et al. 1990; Brown and Rogers 1993; Furnell et al. 1996; Furnell et al. 1995; Joyce and Gupta 1990; Leggett and Williams 1988; Monrose and Rubin 1997; Obaidat and Sadoun 1997; Robinson et al. 1998; Shepherd 1995; Song et al. 1997; Umphress and Williams 1985). Song et al. (1997) use the same representation as we do for continuously monitoring the user's keystrokes. In particular, the mean and standard deviation of the duration of a key press and the delay between a key release and another key press are calculated over two or more consecutive press and release events. Next, the probability that the current duration of key presses and the delay between keystrokes belong to the current user is calculated. If the sum of probabilities is very low for a number of time steps then a different user is likely to be sitting in front of the terminal.

The main problem with model based approaches is how to treat outliers. Some researchers e.g. Joyce and Gupta (1990), Leggett and Williams (1988), and Umphress and Williams (1985) remove outliers. However the difficult task is to define what is an outlier. With our approach all user patters are stored. The authorized user will occasionally produce outliers but the large majority of typing patters will be consistent, i.e. not raise the activation value. In contrast, an unauthorized user will almost always produce outliers which keep raising the activation value of the system. Adaptation is achieved by storing molecules in the pool of detectors after a delay of several iterations. Thus our pool of detectors represents an undistorted view of the user's typing pattern. Furnell et al. (1995) and Furnell et al. (1996) achieve a detection rate of 85% within 160 keystrokes or less with a system based on statistical methods. Our system is simpler yet we detect 80% of the intrusion attempts within 160 keystrokes or less.

7 CONCLUSIONS

The natural immune system does a very good job of protecting the body from diseases. Analysis of the natural system can provide many paths to increased computer security. Several successful systems have already been proposed. In our research we focused on the role of negative selection in the immune system. The negative selection operator does a beautiful job in the natural system but is not necessarily useful in an artificial system. The natural system basically has no other way to detect foreign antigens than to remove those cells which produce antibodies which detect its own molecules. However, in a computer system we are able to determine easily if an element is *not* a member of a given set. Thus we don't need to invoke the negative selection algorithm here. As a practical problem to illustrate this case we have chosen user authentication using keystroke analysis. Samples of the user's typing characteristics were stored in a pool of detectors and compared with the current typing behavior. If the typing behavior deviates too much from the normal typing behavior then a different user is likely to be using the keyboard. Experimental results were provided for 5 different users. In each case non-self was

Table 1: Self and non-self detection for 5 users. Each graph shows the activity level over time. The graphs in the diagonal show how the system behaves for a single user when 100 molecules are skipped after 2000 molecules have been processed. Because the typing behavior is still the same, no change is detected. In contrast, the activity level rises sharply whenever a different user's stream is processed. The vertical line denotes the time when 2000 molecules have been processed.



quickly detected after a change to a different user occurred.

Acknowledgement

We thank Rob Shipman from BT Labs, UK, and Richard Watson from Brandeis University, USA, for helpful comments on the draft version of this paper.

References

B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson (1994). *Molecular Biology of the cell* (3rd ed.). New York: Garland Publishing.

S. Bleha, C. Slivinsky, and B. Hussien (1990). Computer-access security systems using keystroke dynamics. *IEEE Trans. on Pattern Analysis and Machine Intelligence 12*(12), 1217–1222.

M. Brown, and S. J. Rogers (1993). User identification via keystroke characteristics of typed names using neural networks. *Int. Journal of Man-Machine Studies 39*, 999–1014.

X. Cui, M. Li, and T. Fang (2001). Study of population diversity of multiobjective evolutionary algorithm based on immune and entropy principles. In *Proc. of the 2001 Congress on Evolutionary Computation, COEX, Seoul, Korea*, pp. 1316–1321. IEEE Press.

D. Dasgupta, Y. Cao, and C. Yang (1999). An immunogenetic approach to spectra recognition. In *Proc. of the 1999 Congress on Evolutionary Computation*, Mayflower Hotel, Washington D.C., USA, pp. 1859–1866. IEEE Press.

P. D'haeseleer, S. Forrest, and P. Helman (1996). An immunological approach to change detection: Algorithms, analysis and implications. In *Proc. of 1996 IEEE Symposium on Computer Security and Privacy*, pp. 110–119. IEEE Computer Society Press.

S. Forrest, S. A. Hofmeyr, and A. Somayaji (1997).

Computer immunology. *Communications of the ACM 40*(10), 88–96.

S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff (1996). A sense of self for unix processes. In *Proc. of 1996 IEEE Symposium on Computer Security and Privacy*, pp. 120–128. IEEE Computer Society Press.

S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri (1994). Self-nonself discrimination in a computer. In *Proc. of the 1994 IEEE Symposium on Research in Security and Privacy*. IEEE Computer Society Press.

S. M. Furnell, J. P. Morrissey, P. W. Sanders, and C. T. Stockel (1996). Applications of keystroke analysis for improved login security and continuous user authentication. In *Proc. of the 12th Int. Conf. on Information Security (IFIP SEC '96), Island of Samos, Greece*, pp. 283–294.

S. M. Furnell, P. W. Sanders, and C. T. Stockel (1995). The use of keystroke analysis for continuous user identity verification and supervision. In *Proc. of MEDIACOMM 95 - Int. Conf. on Multimedia Communications, Southampton, UK*, pp. 189–193.

A. Gaspar and P. Collard (1999). From gas to artificial immune systems: Improving adaptation in time dependent optimization. In *Proc. of the 1999 Congress on Evolutionary Computation*, Mayflower Hotel, Washington D.C., USA, pp. 1859–1866. IEEE Press.

S. A. Hofmeyrd and S. Forrest (1999a). Architecture for an artificial immune system. *Evolutionary Computation 7*(1), 45–68.

S. A. Hofmeyr and S. Forrest (1999b). Immunity by design: An artificial immune system. In *Proc. of the Genetic and Evolutionary Computation Conference*, pp. 1289–1296. Morgan Kaufmann.

R. Joyce and G. Gupta (1990). Identity authentication based on keystroke latencies. *Communications of the ACM 33*(2), 168–176.

S. A. Kauffman (1993). *The Origins of Order. Self-Organization and Selection in Evolution.* Oxford: Oxford University Press.

J. O. Kephart (1994). A biologically inspired immune system for computers. In R. A. Brooks and P. Maes (Eds.), *Artificial Life IV: Proc. of the 4th Int. Workshop on the Synthesis and Simulation of Living Systems*, pp. 130–139. MIT Press.

J. Kim and P. J. Bentley (2001a). An evaluation of negative selection in an artificial immune system for network intrusion detection. In *Proc. of the 2001 Congress on Evolutionary Computation, COEX, Seoul, Korea*, pp. 1244–1252. IEEE Press.

J. Kim and P. J. Bentley (2001b). An evaluation of negative selection in an artificial immune system for network intrusion detection. In *Genetic and Evolutionary Computation Conference 2001, San Francisco*, pp. 1330–1337.

A. J. Klosterman and G. R. Ganger (2000). Secure continuous biometric-enhanced authentication. Technical Report CMU-CS-00-134, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.

J. Leggett and G. Williams (1988). Verifiying identity via keystroke characteristics. *Int. Journal of Man-Machine Studies 28*, 67–76.

F. Monrose, M. K. Reiter, and S. Wetzel (1999). Password hardening based on keystroke dynamics. In *6th ACM Conf. on Computer and Communications Security, Kent Ridge Digital Labs, Singapore*, pp. 73–82. ACM Press.

F. Monrose and A. Rubin (1997). Authentication via keystroke dynamics. In *4th ACM Conf. on Computer and Communications Security*, pp. 48–56.

M. S. Obaidat and B. Sadoun (1997). Verification of computer users using keystroke dynamics. *IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics 27*(2), 261–269.

J. A. Robinson, V. M. Liang, J. A. M. Chambers, and C. L. MacKenzie (1998). Computer user verification using login string keystroke dynamics. *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans 28*(2), 236–241.

S. J. Shepherd (1995). Continuous authentication by analysis of keyboard typing characteristics. In *European Convention on Security and Detection, 16-18 May*, pp. 111–114. IEE.

A. Somayaji, S. A. Hofmeyr, and S. Forrest (1998). Principles of a computer immune system. In *1997 New Security Paradigms Workshop*, pp. 75–82. ACM.

D. Song, P. Venable, and A. Perrig (1997). User recognition by keystroke latency pattern analysis.

A. M. Turing (1950). Computing machinery and intelligence. *Mind 59*, 433–560.

D. Umphress and G. Williams (1985). Indentity verification through keyboard characteristics. *Int. Journal of Man-Machine Studies 23*, 263–273.