Password Security through Negative Authentication

Dipankar Dasgupta and Sudip Saha

Department of Computer Science
Center for Information Assurance
University of Memphis, Memphis, TN 38152
Email: dasgupta@memphis.edu

Abstract:

Authentication systems generally adopt the approach of checking an authentication request in a list of stored profiles of identification and verification information. This stored sensitive information always has the risk of being hacked and exploited by malicious attackers. Negative Authentication is an approach inspired by biological immune mechanisms that mitigates this risk. In particular, this approach exploits a form of complement profiles which resembles the censoring and maturation process of T-cells. The scope and applicability issues of this technique in the context of currently used authentication systems have been addressed in this paper. It has been pointed out where and how the technique can enhance security of authentication systems. Negative authentication mechanism is based on the generic negative selection algorithm found in artificial immune system literature. In experimentation and implementation, the use of a real valued negative selection technique has been examined in this paper. The performance aspects of the technique along with security considerations have been analyzed and feasible configuration settings have been pointed out for practical purpose.

Key Words: positive authentication, passwords, anti-passwords, negative selection, artificial immune systems, 2-layer authentication.

1 Introduction:

Authentication is the process of confirming the claimed identity of a subject. This is a very important process in all kinds of computing systems ranging from earlier timesharing systems to recent time e-commerce sites. Secure authentication is one of the cornerstones of overall computer security. Unfortunately there is no single foolproof authentication technique available. Different systems are constructed depending on the differing type and degree of security requirements. Password authentication is one of them. It is the technique of authenticating by a secret called password which is disclosed by the subject. Due to its effectiveness and feasibility across various computing environments, password authentication is the most widespread authentication method compared to other techniques such as hardware tokens [6], bio-metrics [5], graphical passwords [3], etc. As a result, this technique has been subjected to various forms of attacks. Consequently to mitigate those attacks, many variations of password authentication has been devised and further enhanced over time, as discussed in [21][31].

Authentication, in general, is comprised of two processes: identification and verification [20]. In this particular case of password authentication, identification refers to identifying the access client with its username and verification refers to matching the entered password against the stored profile. In this paper, the word "credential" has been used as a synonym for "username and password pair". The server has to store user credentials, which should be kept very safe and secured. On the other hand, the client has to send his credentials securely over the communication channel. Both of these processes are important from a security point of view, because herein lie crucial threats to secure access. As pointed out in [22], there are three broad ways in which the secret password can be learned by an intruder: (i) Gaining access to secret profiles stored in the system, (ii) eavesdropping to gain information in the communication channel, and (iii) the inadvertent disclosure of secret information by the user. The authentication system should be designed in such a way that these possibilities can be checked. However, the third issue is out of scope from password authentication's point of view, because the same correct password information revealed by two different entities are indistinguishable by the system [22]. But, the first issue, that is, the vulnerability of malicious access to server authentication information, is an important point that researchers have addressed continuously over the years [13] [21][24]. Threats to this vulnerability appear in two forms – online guessing attacks

and offline guessing attacks. When an attacker guesses all probable passwords and matches those guesses by interacting with the system, it is called online guessing attack. Whereas, if the attacker somehow get hold of the password list and matches some guesses against that list offline, it is called an offline guessing attack. Obviously, an offline attack is more useful on the part of the attacker because of no system interaction and quick computation. Various techniques have been proposed to mitigate this threat by introducing expensive hashes like Blowfish [30], Bcrypt [29] etc. Still, studies showed that success in offline guessing attack ranged from 50% to 35% in 90's [31] and up to around 60% in recent times [25]. The focus of this work is mainly to address this problem by a technique called negative authentication.

In contrast to negative authentication, the traditional identification and verification process of password authentication has been termed as positive authentication in [10]. It is called positive in the sense that real credentials are stored in some form, i.e., plain-text or encrypted, and used accordingly to verify an access request. On the other hand, negative authentication is inspired by the mechanisms of the biological immune systems [11]. It incorporates the concept of anti-password. An anti-password or Anti-P refers to an entity which is something other than a valid password. Given a credential profile, a generated set of Anti-Ps represents a reasonable approximation of complement space of valid credentials. This complement space is meant to put an additional barrier to handling bad requests and to safeguard the positive database. The rest of the paper is organized as follows. Section 2 talks about various attacks which necessitate password protection. Section 3 describes the concept of negative authentication. Various authentication systems and the applicability of negative authentication have been discussed in section 4. Section 5 describes the model and its security considerations have been discussed in section 6. Experiments and performance analysis is shown in section 7. Finally, the paper ends with conclusion and future directions in section 8.

2 Authentication Techniques and Attacks:

Authentication techniques have been broadly classified [8] according to their different characteristics or factors they use: (i) something the subject knows, (ii) something the subject has and (ii) something the subject is. The second category of authentication involves token based schemes like smart cards or password calculators etc. This technique is very secure and hard to break; but hardware tokens are expensive and involve the risk of loss and issue of portability. The third category includes biometric authentication techniques that use physiological features like fingerprints, voice etc. This technique is very easy and portable. But its weakness is its inflexibility and high cost [5]. This kind of authentication is subject to sniffing and replay attacks. Password based authentication techniques fall under the first category. Under this broad category many techniques have appeared that have evolved against various attacks.

One of the basic problems of password authentication is the risk of malicious access of password file. Although password file theft is currently not a big threat given the fact that high physical security and server encapsulation is maintained, but still there are ways to access that sensitive file. For example, use of malicious program – Trojan horse, Trojan login or weakly managed audit trails make the password file or a portion of it accessible to the attackers [31]. Weakness in software is also exploited to get a copy of the password file, for example through back door, buffer overrun as in classical 1988 internet worm attack [33]. To guard against this threat, one-way hash function was introduced to convert the passwords and store them in file. Secure hash functions like MD5, DES, SHA-1 etc have the property of being irreversible and uniform. So, a hash cannot simply be reversed to retrieve the password. Various guessing attacks work successfully against hashed passwords. Guessing attack may be either online or offline. Online guessing attacks are relatively easier to guard against. The measure is to lock out a user when several consecutive unsuccessful attempts have been made. But if somehow the hashed password file is accessed offline it becomes easier to crack. Brute-force attacks like dictionary attacks or rainbow attacks perform very well for low entropy passwords in password file, for example "John The Ripper" [39], Oechslin's rainbow attack [27]. Entropy is the measure of the strength of password that indicates how hard it is to guess that password. Feldmeier and Karn [13] pointed out that users should choose high entropy passwords for the sake of long term password security. Still, it was found [24][32] that people in general have the tendency to choose easy-to-guess passwords. Users are generally insisted to follow some strong password-creation rules. But, as Burr et al [7] showed, human-memorable passwords are inherently limited in randomness – only between 18 to 30 bits. Narayanan and Shmatikov [25] showed a sophisticated and smart brute-force technique that exploits this fact to crack even apparently strong passwords. So, compromise of password file is a serious threat to secure authentication.

Another problem of password authentication is sniffing password. In earlier times, keystroke sniffing was used to retrieve password before it is hashed. By use of memory protection of RAM, this problem has been solved. But, other kind of sniffing attacks are prevalent. Physical sniffing like wiretapping captures hashed passwords on transit. Sniffing programs are also available like "Playback", "Keytrap" etc that can access the hashed password. This hashed form can be used in replay attacks to impersonate a legitimate user. This kind of sniffing attack in communication channel is addressed to a great extent by cryptographic authentication techniques. Challenge Handshake Authentication Protocol (CHAP – RFC 1994) is an effective and widely used cryptographic technique. Different challenge is encrypted using the user's password each time authentication is performed. Choice of time frame or counter has synchronization problems; random challenge or nonce is the best and simplest [2]. However different kind of man-in-the-middle attack foils this technique. As a consequence many cryptographic protocols have been devised. Kerberos [26] is the most successful among this kind of techniques. This is a third party authentication system. Both client and server get tokens from a third party authenticator to communicate among themselves.

Social engineering is another kind of attempt to get unexpected access of password. Shoulder surfing, helpful disclosure, bogus password change etc, often work successfully. Password authentication techniques in general have very little to do against this threat. The same password used by a valid and an invalid user is indistinguishable by a machine.

No single authentication system is foolproof; every technique has its strong and weak sides. So, stronger authentication techniques combining multiple factors are also in use. For example, in two-way authentication [6] system, a user may be asked to present his physiological feature as well as a secret password. Hardware token that needs as input a secret password is also a two factor authentication technique.

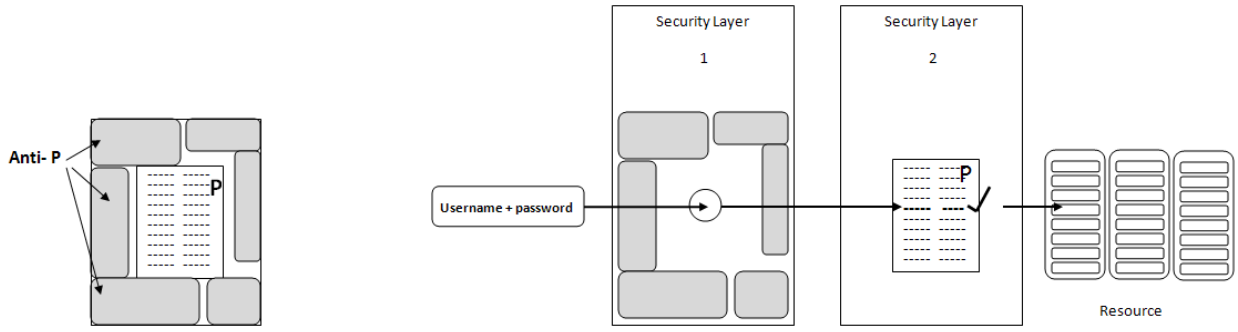
3 Negative Authentication:

Negative Authentication is the process of confirming user identity by use of negative information as opposed to valid or positive information. Information other than valid credentials is termed as negative information in this context. In Negative Authentication, a subject is tested to check if it is not invalid rather than verifying it to be valid. In this technique, profile of invalid credential information is used for checking user authenticity. Essentially negative authentication works in the opposite way, that is, it actually probes invalid requests. A success in the match means a bad request while a failure means the request may be valid. As the profile consists of invalid entities, it is called a complement profile or negative profile. This complement profile represents the negative abstraction of valid credentials and actually is the problem specific realization of negative database described in [12]. The use of negative authentication for system login application was first examined in [10].

Let U represent the set of all possible user credentials (username-password pair) and P be the set of all valid credentials. A traditional password authentication scheme matches an authentication attempt x against the set P. If $x \in P$, then x is regarded as valid. In negative authentication, a set N which is complement of P, that is $N = U - P$ is used for credential verification. A set of Anti-Passwords or Anti-Ps constitute the set N. If an attempt x matches with N, that is $x \in N$, then x is regarded as invalid. N is called the negative or complement profile. If the complement profile covers all possible invalid passwords, then it essentially means the same thing on the part of a guessing attacker. Because the complement of a complete complement profile is the positive profile itself; the attacker can decide unambiguously which guess is correct and which one is not. That is why it is desirable not to cover all the complement space of invalid passwords by the complement profile and thereby, to keep some ambiguity intentionally. So a subset $N' \subset N$ is used for negative profile. As such, negative authentication by itself is not enough for a real authentication purpose; rather it comes as part of a two phase authentication approach. Here negative authentication is used in the first phase and the actual credential database itself is used for positive verification in the second phase for the sake of security enhancement. The approach is illustrated in fig 1.

This approach has two benefits. First of all, the negative authentication module is supposed to detect and filter out most of the invalid requests. Therefore, such invalid requests have a low probability of accessing the positive database. On the other hand, a harder-to-crack negative database is placed in the front security perimeter. It is more vulnerable to malicious access and hence more vulnerable to offline guessing attacks. But as it will be shown later, cracking a negative database is harder than cracking a positive one due to the designed degree of ambiguity

introduced intentionally in that database. So, exposing the negative database in the front part reduces the overall password cracking risk.



(a) (b)

Fig 1: Illustration of Negative Authentication System. (a) From a credential file P, a set of Anti-Ps are generated that constitute N′ ⊂ N = U-P. (b) N′ and P are deployed in two separate layers. When a legitimate user makes an authentication request, no Anti-P in layer 1 detects that, and then it is verified by P in layer 2.

The concept of negative checking is used in other areas in different forms. For example, negative authentication in biometric identification means establishing a negative hypothesis. When two biometric samples β and β' are compared, the traditional null hypothesis checks whether they match or not. But, in negative authentication the null hypothesis checks if they don't match. Consequently the alternate hypothesis is to check their match [5].

The null hypothesis: $H_0 : \beta \neq \beta'$

The alternate hypothesis: $H_a : \beta = \beta'$

Caching of negative responses termed as negative caching [36] is prevalent in DNS queries and routing techniques. Failed events are cached and checked later when the same attempt is made again. Keeping 'blacklist' of hosts for avoiding Distributed Denial of Service (DoS) attacks or for avoiding spam emails [37] are also similar forms of negative information checking. Use of negative information is therefore seen in different computing systems. In this paper, negative profile in the context of password authentication has been experimented.

4 Scope of Negative Authentication:

Negative authentication may work in situations in which positive identification and verification works. Therefore, the scope of negative authentication is defined to some extent by that of positive authentication. From this viewpoint the applicable systems for negative authentication are discussed here.

In the context of World Wide Web, authentication is generally connection oriented which determines the authenticity of the subject at the other end of the connection. Here the subject may appear through a web browser or some web services client application. Some form of positive identification and verification are followed in most of the connection oriented authentication systems, e.g. systems using underlying operating system's authentication features or authentication by custom-configured credential database. These connection oriented authentication systems are either static password based or challenge-response based. Negative authentication is feasible in either form as will be discussed in section 5 and 8. Web service authentication in the other form is document oriented, for example cryptographic digital signatures are used in documents being transferred over untrusted network. In this case, although the authenticator has to store the appropriate key to verify the signature, the verification is not done through a lookup process. The exact key for a user is needed for decrypting message and verifying signature. Hence, the concept of invalid keys come of no use here and negative authentication is out of scope in this case.

Operating systems have their authentication services available for use in stand-alone systems, domain controllers and web servers. Windows LDAP authentication is a positive authentication technique where active directory is used as credential store. As another example, Security Account Manager (SAM) database is used in windows NTLM authentication. Positive authentication is also used in Linux through positive databases, e.g. /etc/passwd or /etc/shadow. Negative authentication is applicable in this kind of scenarios. But, there are some authentication systems, like Kerberos, that do not follow positive authentication methodology. In Kerberos, although the authenticator stores keys, it does not work in a password lookup way. Like document-oriented authentication, Kerberos is also out of scope for negative authentication.

5 Negative Authentication Model and Implementation Technique:

The proposed model involves two separate modules operating on negative and positive authentication processes. The overall model is shown in fig 2. The negative authentication process uses an Anti-P set that is constructed by processing the user credential list. This Anti-P set is updated at regular interval and each time there is any change in credential file. Although regular update is expensive, it makes cracking harder. If a login request passes through the negative authentication module, it is regarded as a probable valid request and checked again in the next positive authentication module. Two modules are placed in two different security tiers. Negative authentication module being less vulnerable to attacks is placed at front security tier. Sensitive positive authentication module is placed in secure back-end tier.

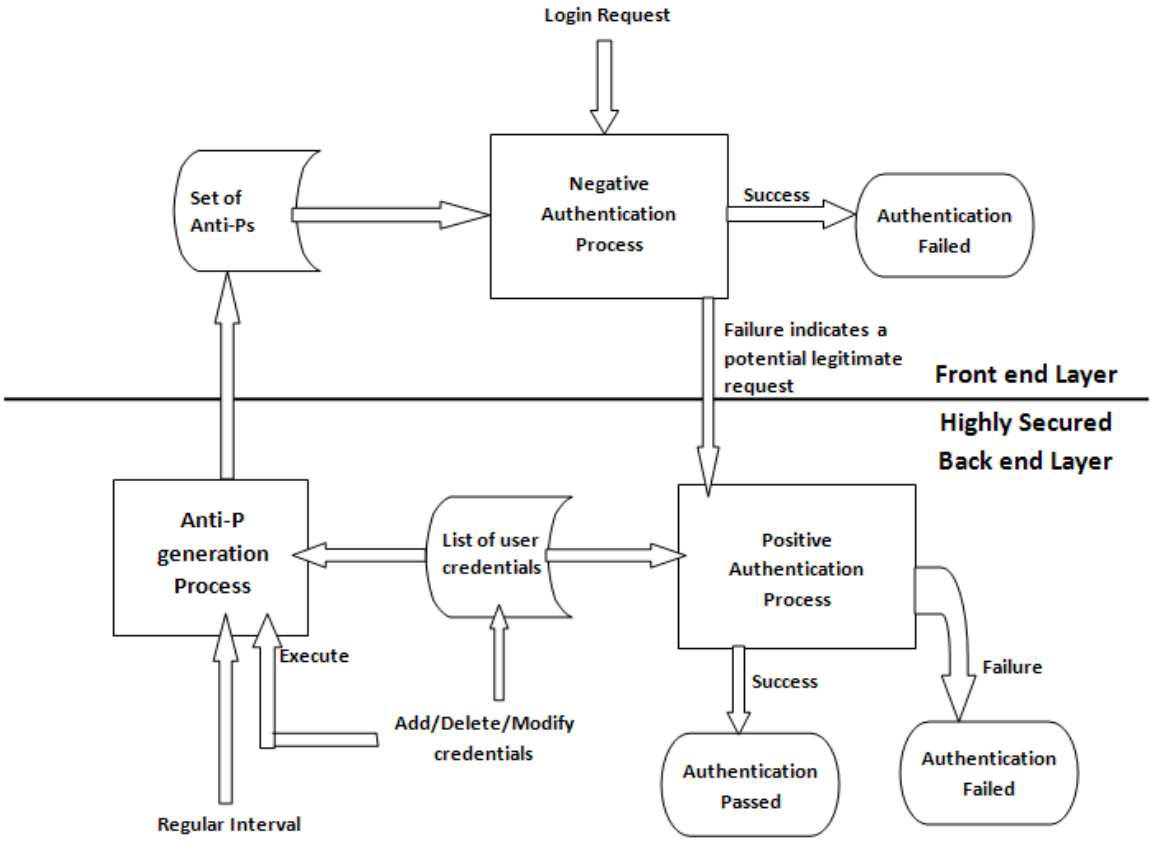


Fig 2: Framework of two stage authentication process. The front layer contains the negative authentication process that primarily checks a login request against the anti-P set. If anti-Ps fail to recognize the request, then positive authentication process at back-end layer checks finally against the credential file. The set of anti-P is updated regularly and each time credential file is modified.

5.1 Representation of username-password

Representation of passwords is important in negative authentication systems, because complement profile size depends on the representation. It is a challenge to limit the size of complement profile which inherently covers a bigger space compared to the limited profile of valid passwords. The idea of negative authentication is based on the generic negative selection algorithm from artificial immune system domain. An Anti-P in negative authentication refers to a detector in negative selection. Various detector generation techniques have been discussed in [11]. Several of those techniques such as Dynamic programming approach, greedy approach, NSMutation, Binary Template etc work for entities in binary representation [1] [9] [35]. However, in a real scenarios, passwords are generally hashed by some cryptographic algorithms like MD5, DES etc and turned into a large fixed length number. As the space hashed password space is large, we consider fragmenting it and turning to a real valued vector. In the literature, there is a collection of negative selection algorithms as well for real valued representation such as evolutionary approach [14], fuzzy detection rules[15], randomized approach[15] etc. The other related work [10] on negative authentication has adopted an evolutionary approach called Niching GA [23] for this purpose. However, in this paper, a simple and efficient statistical approach called real valued negative selection algorithm – V-Detector [17] has been used to generate detectors (Anti-Ps) from the hashed passwords. The algorithm for detector generation as described in [17] is shown in fig 3.

5.2 Anti-P generation

Before using the V-Detector algorithm the username and password pairs are converted to n dimensional real valued vectors as shown in fig 4. The password is hashed by the MD5 hash technique. This hash value concatenated with the username is again hashed with MD5 to get the final fixed length hash value. This is segmented into four parts as a design choice. Each of the parts is standardized between 0 and 1. These four components make up a vector which is represented in a four dimensional space. The vectors corresponding to all of the username-password pairs are fed as input to the detector generation algorithm as proposed in [17], see fig 3. The output detectors from the process are actually Anti-Ps to be used for negative authentication.

The generic algorithm works as follows. Inputs to the algorithm are some self points in real valued space. Other parameters are self radius and desired coverage of nonself space. The detector generation algorithm runs for a maximum of T_{max} time passed as another input parameter.

First of all, self hyper-spheres are constructed by the self points as center and the self radius as radius. Then a number of points are sampled randomly. If a sample point falls outside all the self regions and all detectors, then a detector hyper-sphere is constructed around that point with the maximum possible radius crossing no self hyper-sphere. In this way, a sufficient number of detectors are sampled until the desired coverage of the nonself space is achieved. The expected coverage is determined in the following way as pointed out in [18]. If *m* consecutive non-self points sampled fall inside any of the already constructed detectors, then the probability of having uncovered non-self space can be regarded as at most *1/m*. Hence the covered non-self space is probable to be at least *(1-1/m)*. If this value becomes equal to or more than the desired coverage the generation process ends. The calculation of this coverage is based on the assumption that the distribution of all passwords is uniform over the whole unit hyper-cube space. This is a probabilistic estimation of coverage.

In this algorithm, self regions in the shapes of 4D spheres are specified in four dimensional spaces for each of the username- password pairs. We call the radius of the hyper-sphere the confusion parameter, because volume of the hyper-sphere is meant to introduce some kind of confusion for the attacker. Some username-password pairs other than the valid ones sitting at the center may fit into the hyper-sphere and avoid detection. Hence, the attacker takes those escaped invalid entities as candidate entities for cracked password. The bigger the radius the more confusion the attacker faces.

V - Detector - Set(S, T_{max}, r_s, c_o)

S : set of self samples

T_{max} : maximum number of detector

r_s : self radius

c_0 : estimated coverage

1 : D ← ∅

2 : Repeat

← 0

T ← 0

5 r ← inifinite

x ← random sample from $[1, 0]^n$

7 Repeat for every d_i in D = $\{d_i . i = 1, 2, ...\}$

8 d_d ← Euclidean distance between d_i and x

9 if $d_d \leq r(d_i)$ then, where $r(d_i)$ is the radius of d

10 t ← t +

if $t \geq 1/(1 - c_0)$ then return D

12 go to 4 :

13 Repeat for every s_i in S

14 d ← Euclidean distance between s_i and x

15 if $d - r_s \leq r$ then $r \leftarrow d - r_s$:

16 if $r > r_s$ then $D \leftarrow D \cup \{< x, r >\}$, where < x, r > is a detector

17 else T ← T + 1

18 : if T > 1/(1 - maximum self coverage) exit

19 : Until | D |= T_{max}

20 : return D

Fig 3: Real valued variable sized Detector set generation algorithm as proposed in [1]

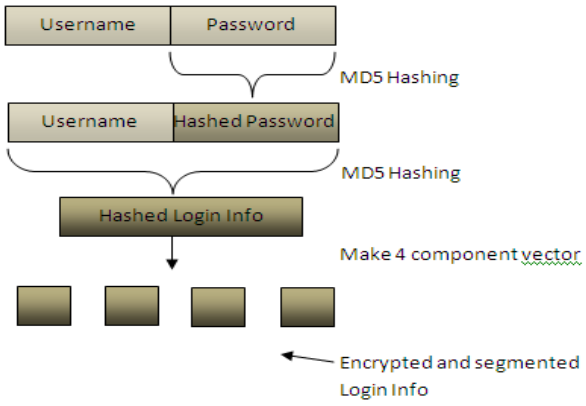


Fig 4: Representation of username-password in real-valued space. Username and passwords are hashed and fragmented to create vectors.

6 Security Consideration of Negative Authentication

The applicability and usefulness of a negative authentication system has been considered from security perspective in this paper. From network security point of view, negative authentication does not add value in the communication link. But, it reduces vulnerabilities related to guessing attacks, particularly in case of offline guesses.

Both positive identification-verification and negative filtering have to use either SSL or challenge-response methodology to secure information transmission, so the same security risk is involved in transmission link. But, as far as security in storage of authentication information is concerned, negative authentication comes with a higher security promise. Real storage of usernames and passwords can be kept in a tighter-security tier, while negative authentication offers a good outward shield. Gaining access to the system containing a positive database is assumed harder. Decoding a valid password from a positive database itself is not straightforward, because passwords are stored in hashed form. It turns out that Anti-Ps are even less risky to reveal password compared to a hashed password list. This is based on the assumption that the hash function is a proper randomizing function and hash outputs are uniformly distributed over the possible range. That is, proximity in plaintext passwords is not preserved in corresponding hash forms, rather they are randomly placed. From the password file, in case of offline guessing attack, the attacker knows which hashes are valid so he can try with a rainbow file. When Anti-P file is compromised, the attacker knows about the region where valid hash points exist; but he has no idea of which specific points are valid ones or how many of them are valid. So, a successful match in negative profile does not simply lead to a successful password crack.

For example, for alphanumeric passwords of 8 characters, the hacker has to try $36^8 \approx 2.82 \times 10^{12}$ instances in worst case scenario to decode a valid password from a hash code. For negative database, it gets even harder. For the same password criteria (8 character alphanumeric), if the confusion parameter is 0.001, then 8×10^6 possible passwords are expected fit to a single password hyper-sphere if uniform distribution of those password instances are assumed. So, the probability that a decoded password is a valid one is extremely low here. And, given the fact that most authentication systems block a malicious user after several consecutive failed tries, the chance for a malicious user to get a password becomes even lower.

In case of online attack too, negative authentication does not compromise with security. If an attacker compromises the system that carries the positive authentication module, he may see output decisions for inputs from rainbow file. The decision is a binary one – Either a guess is valid or invalid. So the attacker gets no extra information about a valid password from an unsuccessful try. However, if a guess is successful, the attacker comes to know a valid password. In the same way, let assume that the attacker can see the output decisions from the negative authentication module, once he has compromised the system. But, seeing guesses coming out successful through the negative authentication module, the attacker cannot become sure to have guessed a real valid password. However, he may get an idea of positioning of password and anti-password space from those successful and unsuccessful guesses. But, this does not come to any help, as long as the hash function is irreversible, randomizing and uniform. However, if the negative and positive authentication modules coupled together is seen as a black box to the attacker, then this black box means to him the same thing as a mere standalone positive authentication module. But, the front negative authentication module acts as an interface to the secure system and meant to make the sensitive password file secure. So, the fact that, Anti-P file is harder to crack than password file is important. Negative authentication is therefore a useful way to address security issues in authentication domain.

7 Experiments and Results

The objective in negative authentication is twofold:

(i) decrease the number of Anti-Ps and
(ii) achieve a good detection rate.

There are three parameters involved:

(i) number of valid passwords,

(ii) confusion parameter and
(iii) Anti-P coverage.

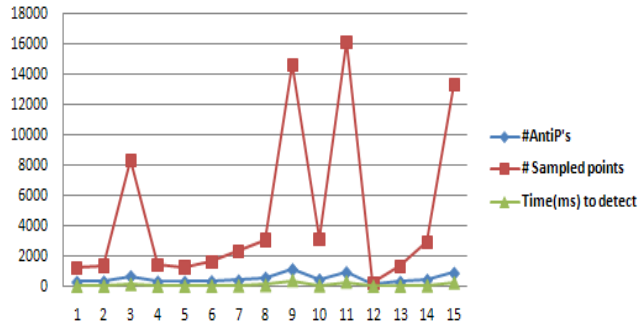


Fig 5: Correlation among #Anti-P's, Sampled points for Anti-P generation and required time to detect invalid requests(ms) in a 2.4GHz Quad core machine.

		# Passwords = 500		# Passwords = 1000		#Passwords = 5000		# Passwords = 10000		# Passwords = 50000	
AntiP Coverage	Confusion Parameter	# AntiP	DR	# AntiP	DR	# AntiP	DR	# AntiP	DR	# AntiP	DR
0.9	0.001	463.2	0.6053	804.8	0.558061	2980	0.458422	4530	0.366202	14694	0.256098
	0.05	420.7	0.573337	727.65	0.546811	713	0.175171	781	0.118492	201	0.00436
	0.1	210.35	0.417279	185.4	0.219994	130	0.027142	137	0.015214	146	0.003035
0.95	0.001	757.3	0.792968	1344.55	0.758761	5640	0.700707	11681	0.721334	56708	0.703825
	0.05	692.95	0.7772	1121.6	0.730133	2827	0.577756	2108	0.313175	333	0.008116
	0.1	317.35	0.573521	286.05	0.342639	164	0.030095	178	0.016677	512	0.007796
0.99	0.001	1500.4	0.956289	2696.95	0.95195	12819	0.9234	23655	0.935514	88908	0.865814
	0.05	1389.05	0.951816	2281.8	0.940283	9283	0.8345	6794	0.5643	712	0.016624
	0.1	811.2	0.853774	814.4	0.671861	312	0.09422	203	0.02345	505	0.007429

Table 1: The table summarizes the results of experiments with the following variation: size of password files, Anti-P coverage and Confusion parameters. The shaded results indicate that for a smaller password file a reasonable size Anti-Ps and good detection rates are found in around 99% coverage and for small confusion parameters. But, very large password files lead to poor result.

Time and space complexity of negative authentication is a factor of the number of Anti-Ps. Both the generation time of Anti-Ps and detection time of an invalid entity are related to the number of Anti-Ps as illustrated in fig 5. The number of samples used to generate enough Anti-Ps indicates the generation time of those Anti-Ps. This number, quite expectedly, has been found to have a very high correlation (around 0.92) with the number of Anti-Ps across different experiments. Similarly, the detection time is also highly correlated to the number of Anti-Ps, as each requested username-password has to be checked against the set of Anti-Ps.

The desired detection rate, the other objective, is subject to the choice of the designer. But, it should not be too high or too low. A very high detection rate essentially means the same effect as by positive authentication, because positive profile is highly guessable from this complete negative profile. On the other hand, a very low detection rate

adds no value to the negative detection module. So, the detection rate introduces a trade-off between efficiency of the module and cracking vulnerability. If the detection rate is x%, then around (100-x)% of tested invalid username-passwords will expectedly escape Anti-P detection. For example, if a dictionary of 0.4 million words is used for an offline guessing attack on an Anti-P profile having detection rate 90%, then around forty thousand wrong passwords will escape detection, all of which will be regarded by the attacker as candidates for valid passwords with a probability 1/40000. It is a design choice to consider how low a value for this probability is satisfactory. This can also be variably chosen depending on the other criteria – number of Anti-Ps. We consider 90% detection rate to be a roughly good choice in this experiment.

Table 1 shows the summary of number of Anti-Ps and detection rate performances for different settings of parameters. All the results have been taken by averaging on twenty runs of the same experiment. As seen from the table, good results are found for limited sized password files. But, for very large password files, performance gets poorer. For example, for a password file of fifty thousand entries, it needs a significantly high number of Anti-Ps to achieve even an 86% detection rate. Hence, for the rest of the experiments password file size will be restricted to three thousand entities. Problems with large files will be discussed in section 7.4.

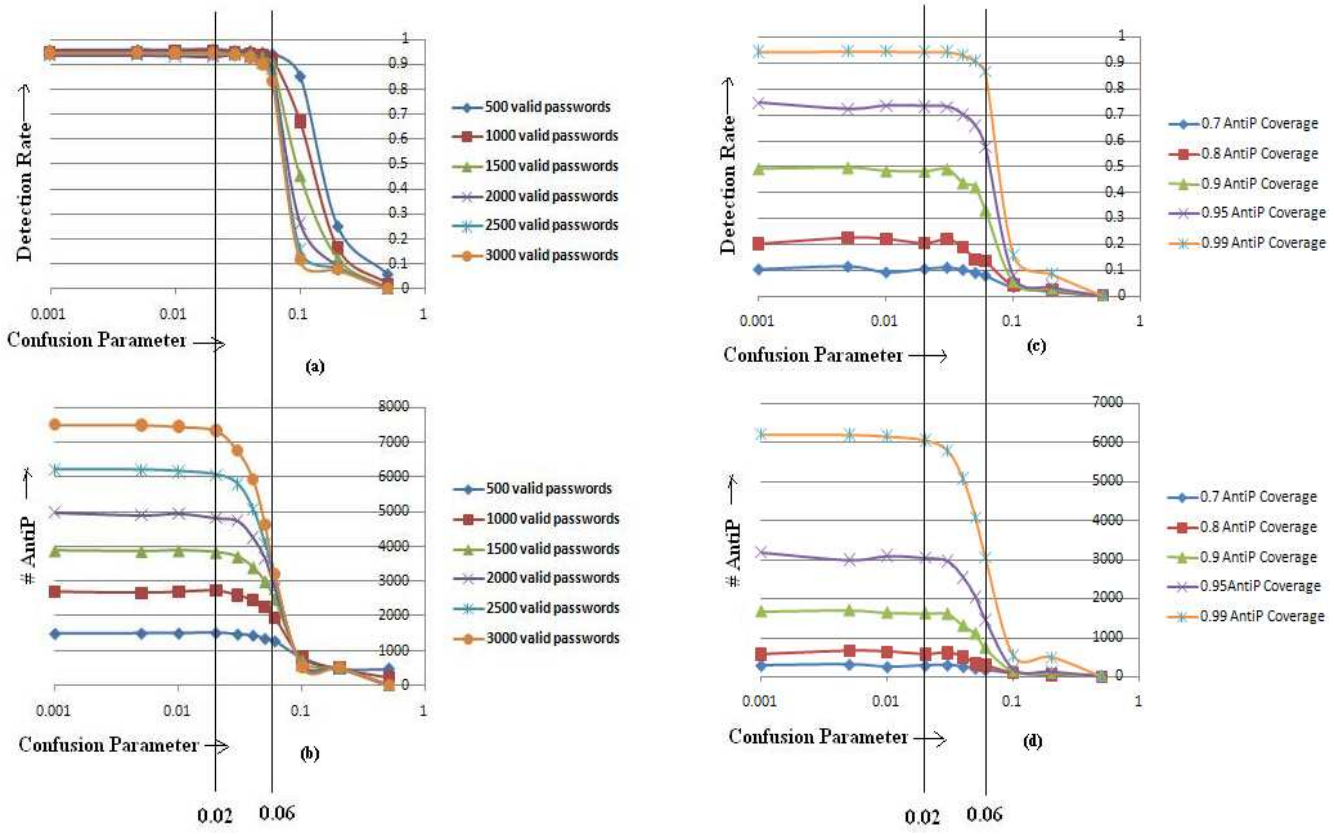


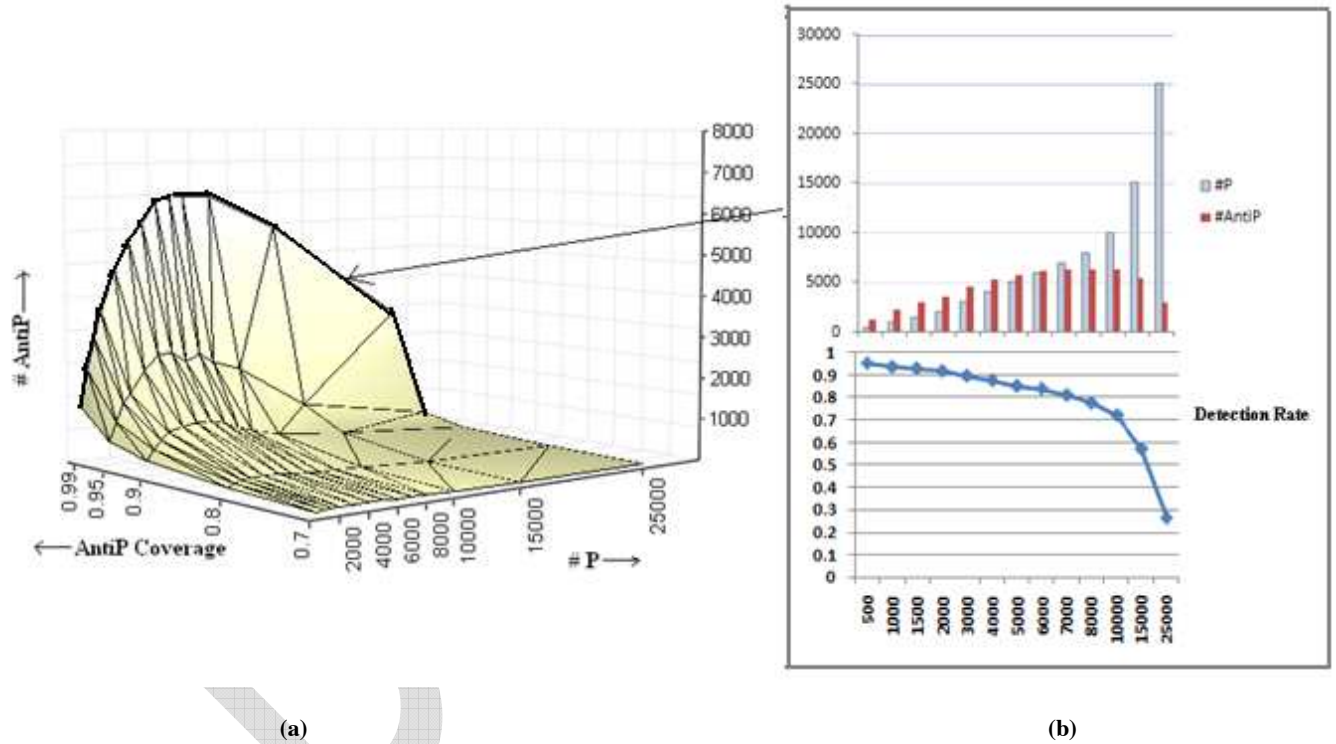
Fig 6: Figure shows the changes of detection rate and the number of Anti-Ps with variations in confusion parameter. In particular, figure 6 (a) exhibits detection rate with varying sizes of password files while setting the Anti-P coverage to 0.99. It is to be noted that in the interval 0.02-0.06 of confusion parameter, while the detection rate does not change significantly (figure 6(a)), the number of Anti-P's decrease rapidly (figure 6(b)). For example, with #P = 3000 and confusion parameter =0.05, detection rate is 0.9 can be achieved with 4625 Anti-Ps, compared to 0.94 detection rate with 7446 Anti-Ps as in the case of confusion parameter of 0.01 . It is also observed that the same interval (0.02-0.06) of confusion parameter produces good results in terms of number of required Anti-Ps for desired coverage as in Figure 6(c) and Figure 6(d), using a fixed size password file (3000).

7.1 Selecting Confusion Parameter

Table 1 indicates very little change in number of the Anti-Ps and detection rates for a wide range of confusion parameters (0.001 to 0.05). However, for relatively higher values (0.1), this change becomes significant. The change of the number of Anti-Ps and detection rate is shown in fig 6.

The higher confusion parameter shrinks the non-self space. But interestingly, the number of Anti-Ps doesn't decrease that much for increase in confusion parameter up to some level. This is probably due to the sparse nature of the valid password points. Anti-P set size probably is dictated by the interleaving gaps between corresponding hyper-spheres, rather than by the total area of Anti-P space, as long as this total area does not become too small. In the scattered space of valid password points, the confusion parameter appears to have little effect on the Anti-P set size upto the values around 0.01-0.02 of confusion parameter.

After 0.02 the number of Anti-Ps falls rapidly, whereas the detection rate falls after a higher threshold. As seen in fig 6, for the password files of up to three thousand entities, the range 0.02-0.06 is a good choice for confusion parameter. In this rage, the number of Anti-Ps suffers a rapid fall whereas a good detection rate is almost retained. For small compromise in detection rate, a reasonable decrease of Anti-P set size is achieved in this range. The same result follows for different Anti-P coverage too. Therefore, the value 0.05 has been chosen for confusion parameter in the next experiments



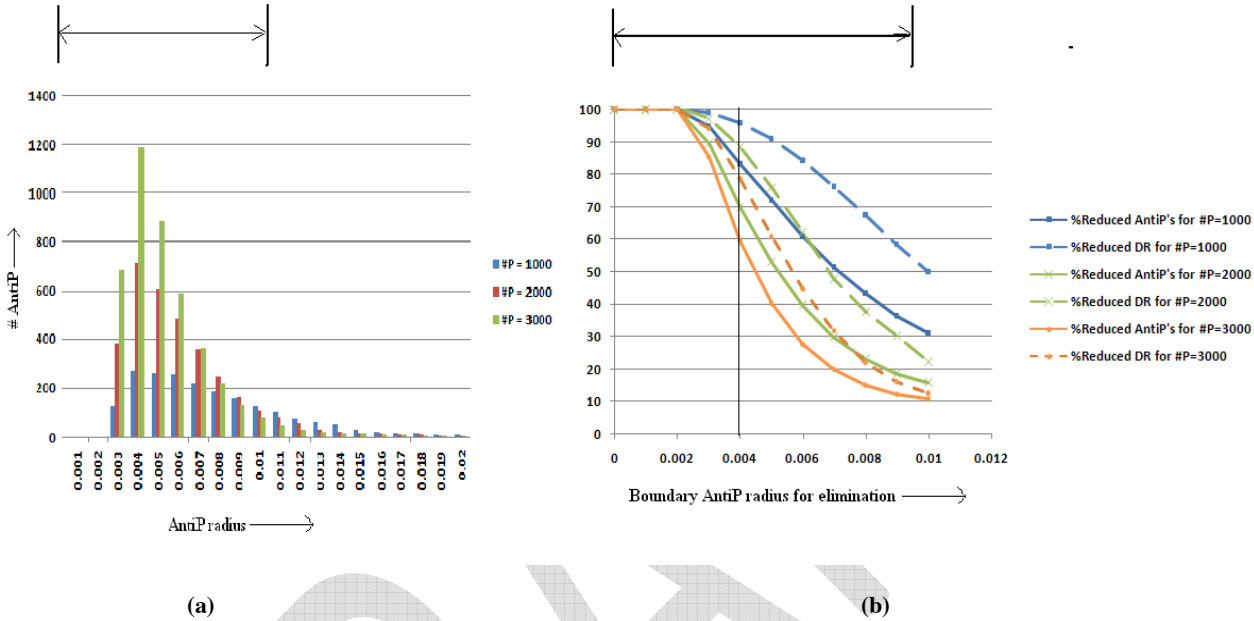
(a) **(b)**

Fig 7: (a) Variation of Anti-P set size with change in Anti-P coverage and password set size. (b) Variationof Anti-P set size and detection rate with change in password set size for coverage 0.99. Anti-P set size varies almost exponentially with coverage whereas it has a limit on increase for varying password set size

7.2 Effect of Number of Passwords and Anti-P Coverage

The number of Anti-Ps increases quite exponentially for increase in Anti-P coverage, see fig 7. So, estimated coverage is to be chosen carefully to maintain a reasonable time and space complexity. However as seen from the figure, with the increase in password file size, the number of Anti-Ps increases to some limit and then decreases. This is expected because many hyper-spheres in a limited hyperspace leaves many holes which necessitates many Anti-Ps to cover; but for very large number of hyper-spheres the whole space is almost absorbed within those hyper-spheres that leave little room for anti-Ps to cover. Although the size of Anti-P set decreases for very high password

sets, it affects adversely on the detection rate. To get rid of this problem, a low confusion parameter can be chosen; or the space of the universe set can be expanded by increasing the dimension from four to a higher number (discussed in subsection 7.4). Both of these schemes arrange for more space in nonself space. The trend of Anti-P set size has been shown effectively for coverage 0.99.



(a) **(b)**

Fig 8: (a) The variation of sizes in a pool of Anti-P's produced for password file sizes of 1000, 2000 and 3000. (b) Each pair of the same colored lines represents variation of Anti-Ps and detection rates for a specific password file size. It is clear from the figure that if smaller sized Anti-Ps are eliminated, then the Detection rate decreases, but the number of Anti-Ps decreases faster than Detection rates. For example, if in the case of 1000 valid passwords, the Anti-P sizes of 0.04 or smaller are eliminated, then the #Anti-P counts reduce to 83% of the original number, whereas, detection rate falls only to 95%.

7.3 Elimination of smaller Anti-Ps

It turns out that many small Anti-Ps are generated to fulfill the desired coverage. Some of them may not contribute that much in detecting invalid passwords. So eliminating such Anti-Ps may lead to better performance.

Fig 8(a) shows the distribution of Anti-P size in terms of radius. As seen from the figure, many small size Anti-Ps are produced. If smaller Anti-Ps are eliminated the impact on Anti-P set size and detection rate is shown in fig 8(b). Both detection rate and number of Anti-P's decrease, but the decrease of Anti-Ps is faster. This presents an opportunity to eliminate some small Anti-Ps so that reduction of Anti-Ps can be achieved at the cost of comparatively less effect on the detection rate. How much compromise of the detection rate can be tolerated is a design choice.

7.4 Increasing Dimension

As pointed in Table 1, Anti-P generation yields poor results for large password sets. Having more and more passwords means stuffing more and more hyper-spheres into the same limited size hypercube; this leaves less amount of space for Anti-P generation. The problem can be alleviated by decreasing the confusion parameter. However, this cannot be done beyond a specific limit, because some extent of confusion has to be incorporated for security purposes. Another alternative is to increase the dimension. That is, instead of fragmenting the password

hash into four parts, more fragments can be generated and plotted in a higher dimensional space. The impact of dimension increases for large password sets has been shown in fig 9.

Higher dimension is a concern in general for classification instances because of the difficulty of representing a region in a high dimensional space. But, in this negative authentication instance, there is no issue of representing an unknown region. The supplied valid passwords are all that define the valid region. Furthermore, high dimension is beneficial for VDetector algorithm. As pointed out in [24], VDetectors are in fact maximized by design which gives a good opportunity to cover an increased higher dimensional space with relatively fewer number of detectors.

Fig 9 shows the trend of detection rate and number of Anti-Ps for variation in dimension. The infeasible detection rates found in four dimension for large password sets with confusion parameter set to 0.05 or higher, is recovered in higher dimensions. Although number of Anti-Ps increases in high dimensions, but detection rate becomes quite reasonable.

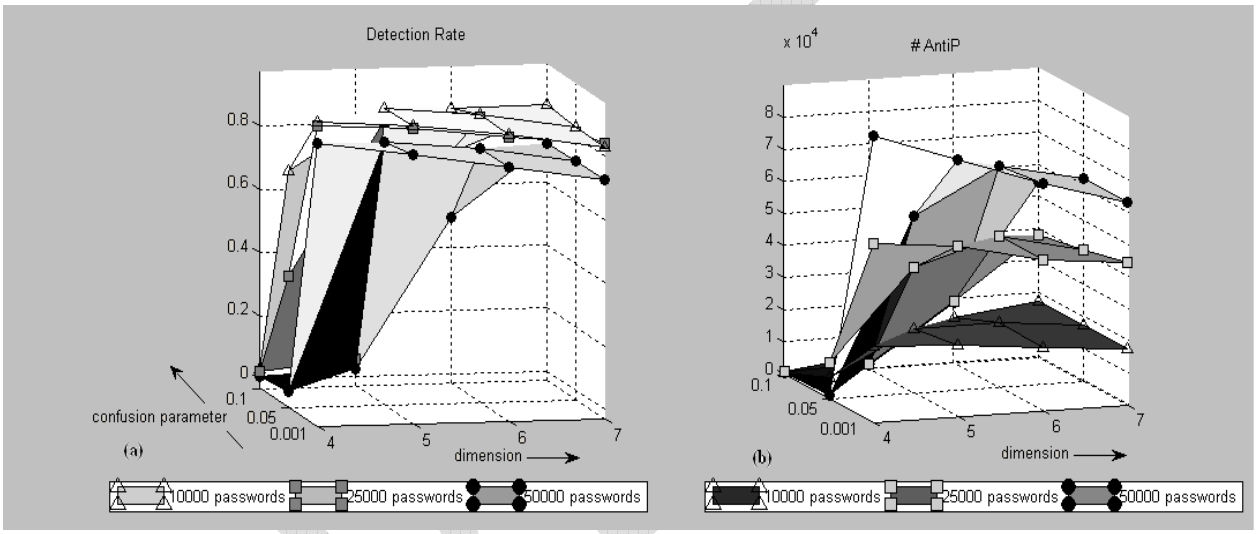
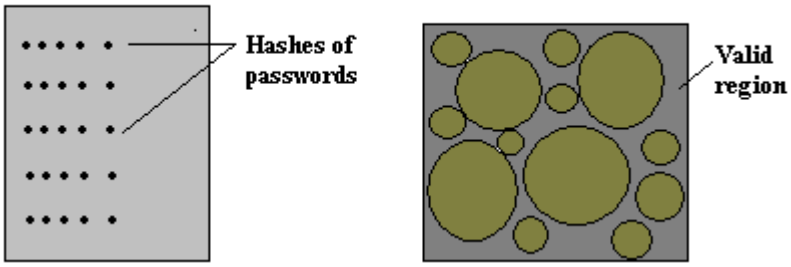


Fig 9: Variation of (a) Detection Rate and (b) Anti-P set size with change in dimension for different confusion parameters and different size of password sets. As dimension is increased from 4, detection rate becomes high for all settings of confusion parameters. But, at the same time number of Anti-Ps also increases.

8 Conclusion and Future Directions

The performance aspect of Anti-P generation has been studied in this paper. However, an implementation of a prototype authentication system with decoupled positive and negative authentication modules across separate hosts in network scenario will be an interesting model to investigate the detailed feasibility and security issues of the model. As another extension, using several Anti-P sets instead of just one may be a good model to confuse hackers more about the valid usernames and passwords. In this regard, efficient and ambiguous distribution of Anti-Ps over different sets is sought for such a model. The principle of challenge response in the context of negative authentication also demands more than one Anti-P sets.

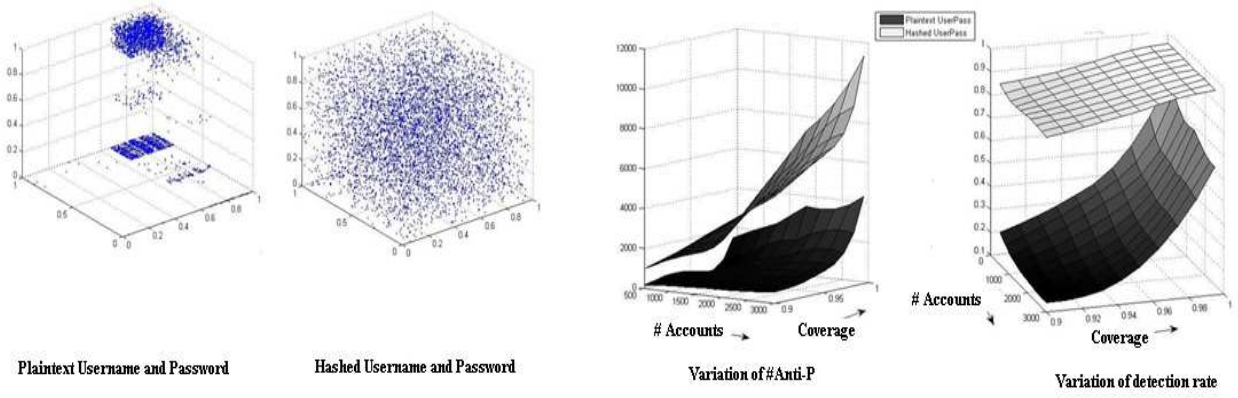


Password file Anti-P file

Fig 10: Comparison of Information from password file and anti-password file. Exact valid hashes are found from the password file. But, anti-password file only reveals a region, from which any point (a point corresponds to a hash) may be valid or invalid.

One of the major problems of the system of negative authentication proposed so far is its inability to dynamically update credential modification, addition or deletion. Each time a credential is changed, the whole set needs to be changed. This is a problem for negative selection in general. How to get around this problem is set as future work.

One expectation from generic negative selection algorithm is that the number of detectors should not exceed the number of self samples [25], because otherwise efficiency is compromised. But in this case of negative authentication, Anti-Ps or detectors are not used for efficiency, rather for security purpose. These are meant to make password profiles more ambiguous and hence secured. So, having a large detector set size, if not too larger than the password file, is tolerable in negative authentication as long as the detection rate is good enough.



11(a) **11(b)**

Fig 11: (a) Scatter plot of 5600 username and passwords (taken from openwall project [33])pairs – plaintext credentials (left plot) and hashed credentials (right plot). (b) Performance comparison of plaintext and hashed account information – variation of Anti-P (left graph) and detection rate (right graph) with change in parameters.

There is one intrinsic problem in this technique which is the sparsity issue of hashed plots that necessitates a higher number of Anti-Ps. Fig 11 shows that usual username and password plots are intrinsically clustered; but if hashed they become scattered which is obvious from the security goal and mechanisms of hash functions. That is why, the number of Anti-Ps required in hashed space become higher compared to that in plaintext space. Although the detection rate is favorable in hashed space, as seen from fig 11, the number of Anti-Ps poses scalability problems. The issue of sparsity limits the scalability of this method. For practical use in large-scale systems like in e-commerce sites, scalability problem needs to be addressed, too, which is left as future work.

There are other existing techniques e.g. Bloom Filter [42], Cuckoo Hashing [43] which serve the similar purpose of obfuscating information. Both of them offer false positives like Negative authentication does. These techniques too have the scalability issue. Although they are more space efficient compared to negative authentication, there are some security issues. Hashed bit stream in those techniques are more easily subject to change and alteration, whereas encrypted negative information needs more effort to alter.

The Negative authentication technique has been proposed in this paper to address the security issue of password file. The enhanced security of the system comes at the price of compromising its efficiency. However, as computing power is increasing day by day, this is not a very serious problem. The relevance of negative authentication remains valid if there is security risk of password file being compromised.

9 References

[1] M. Ayara, J. Timmis, R. De Lemos, L. De Castro, R. Duncan, "Negative Selection: How to generate detectors", *Proceedings of the 1st International Conference on Artificial Immune System*, 2002.

[2] B. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva and M. Yung, "Systematic Design of a Family of Attack-Resistant Authentication Protocols", *IEEE Journal on Selected Areas in Communications*, 11(5): 679-693, June, 1993.

[3] J.C. Birget, D. Hong, N. Memon, "Graphical passwords based on robust discretization", *IEEE Transactions on Information Forensics and Security*, 1(3), Sept. 2006, pp 395-399.

[4] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors". *Communications of the ACM*, Vol 13, Issue 7, 1970.

[5] R. Bolle, J. H. Connell, Sharath Pankanti, Nalini K. Ratha, Andrew W. Senior, "Guide to Biometrics", *Springer*, Nov, 2003.

[6] D. de Borde, "Two-factor authentication", *Siemens Insight Consulting*, June, 2007.

[7] W. Burr, D. Dodson, and W. Polk, "Electronic authentication guideline", *NIST Special Publication*, 800-63, 2004.

[8] S. Carlton, J. Taylor, and J. Wyszynski, "Alternate authentication mechanisms", *In Proceedins of the 11th National Computer Security Conference*, Washington, DC, 1988. National Bureau of Standards.

[9] P. D'haeseller, S. Forrest, P. Hellman, "An immunological approach to change detection: Algorithms, analysis and implications", *Proceedings of the 9th IEEE Computer Security Foundations Workshop*, pp 18-26, 1996.

[10] D. Dasgupta, R. Azeem, "An Investigation of Negative Authentication Systems", *Proceedings of 3rd International Conference on Information Warfare and Security*, Omaha, USA, April, 2008.

[11] D. Dasgupta, F. Nino, "Immunological Computation: Theory and Applications", CRC Press, 2008.

[12] F. Esponda, E.S. Ackley, P. Helman, H. Jia, and S. Forrest, "Protecting Data Privacy through Hard-to- Reverse Negative Databases", *Proceedings of Ninth Information Security Conference (ISC'06)*, Springer LNCS 4176, pp.72-84, September 2006.

[13] D. C. Feldmeier and P. R. Karn, "UNIX password security - ten years later", *In Proc. CRYPTO '89*, Vol 435 of LNCS, pages 44–63. Springer, 1989.

[14] F. Gonzalez, D. Dasgupta, "Anomaly detection using real-valued negative selection", *Genetic Programming Evolvable Machines*, Vol 4, 2003.

[15] F. Gonzalez, "A study of artificial immune systems applied to anomaly detection", Phd Dissertation, The University of Memphis, May, 2003.

[16] B. Hartman, D. J. Flinn, K. Beznosov, S. Kawamoto, "Mastering Web Services Security", *Wiley Publishing Inc*, 2003.

[17] Z. Ji, D. Dasgupta, "Real-Valued Negative Selection Using Variable-Sized Detectors", *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, Seattle, Washington, June, 2004.

[18] Z. Ji, D. Dasgupta, "Estimating the Detector Coverage in a Negative Selection Algorithm", *Proceedings of Genetic and Evolutionary Computation(GECCO)*, Washington, D. C., June, 2005.

[19] Z. Ji, D. Dasgupta, "Applicability issues of the Real-Valued Negative Selection Algorithms", *Proceeding of the Genetic and Evolutionary Computation Conference (GECCO)*, Seattle, Washington, July, 2006.

[20] S. T. Kent and L. I. Millett, Editor, "Who Goes There?: Authentication Through the Lens of Privacy", *Committee on Authentication Technologies and Their Privacy Implications*, National Research Council, 2003.

[21] D. Klein, "Foiling the cracker: A survey of, and improvements to, password security", *Proceedings of UNIX Security Workshop II*, Berkeley, Calif., Usenix Association, 1990.

[22] L. Lamport, "Password authentication with insecure communication", *Communications of the ACM,* Vol 24(11), 1981.

[23] W. S. Mahfoud, "A comparison of parallel and sequential niching methods", *Proceedings of the 6th Int. Conf. on Genetic Algorithms*, Morgan--Kaufmann, 1995, pp 136--143.

[24] R. Morris and K. Thompson, "Password security: a case history", *Communications of the ACM,* Vol 22(11), pp 594-597, Nov, 1979.

[25] A. Narayanan and V. Shmatikov, "Fast Dictionary Attacks on Passwords Using TimeSpace Tradeoff", *Proceedings of the 12th ACM conference on Computer and communications security (CCS)*, November 7–11, 2005.

[26] B. C. Neuman, T Ts'o, "Kerberos: An authentication service for computer networks", *IEEE Communications magazine*, Vol 32,Issue 9, 1994.

[27] P. Oechslin, "Making a faster cryptanalytic time-memory trade-off", *In Proc. CRYPTO'03*, volume 2729 of LNCS, pages 617–630. Springer, 2003.

[28] R. Pagh and F. F. Rodler, "Cuckoo hashing", *J. Algorithms*, Vol 51, pp 122–144, 2004

[29] N. Provos, D. Maziers, "A Future-Adaptable Password Scheme", *Proceedings of the Annual USENIX Technical Conference*, 1999.

[30] B. Schneir, "Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish)", *Fast Software Encryption*, pp 191-204, 1993.

[31] R.E. Smith, "Authentication: from passwords to public keys", *Addison-Wesley*, 2002.

[32] E. H. Spafford, "The internet worm program: an analysis", *SIGCOMM Comput. Commun,* Rev. 19, 1 , Jan, 1989, pp 17-57.
[33] E. H. Spafford, ''The Internet Worm: Crisis and Aftermath,'' *Communications of the ACM,* 32(6) pp. 678-687 (June 1989).
[34] T. Stibor, J. Timmis, C. Eckert, "A comparative study of real-valued negative selection to statistical anomaly detection techniques", *Proceeding of International Conference on Artificial Immune Systems (ICARIS)*, pages 262-275,2005.
[35] S. Wierzchon, "Discriminative power of the receptors activated by k-contiguous bits rule", *J. Comput. Sci. Technol.*, Vol 1 (3), 2000.
[36] M. Andrews, "Negative Caching of DNS Queries", RFC 2308,Mar 1998.
[37] "IronProt Email Authentication", ex summary, www.ironport.com/pdf/ironport_email_authentication_wp.pdf.
[38] Openwall Project. Wordlists collection, http://www.openwall.com/wordlists/, 2009.
[39] Openwall Project. John the Ripper password cracker, http://www.openwall.com/john/, 2009.