JCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Vol.-7, Issue-10, Oct 2019 E-ISSN: 2347-2693

A Survey Paper on Password Security Techniques

Ankita Hinduja[1*], Pradip Sharma[2]

[1,2] Dept. of Computer Science & Engg., ITM GoI, Gwalior, India

**Corresponding Author: myskhinduja@gmail.com*

DOI: https://doi.org/10.26438/ijcse/v7i10.130136 | Available online at: www.ijcseonline.org

Accepted: 07/Oct/2019, Published: 31/Oct/2019

Abstract - This paper proposes a scheme for password management by storing password encryptions on a server. The method involves having the encryption key into a share for the user and one for the server. The user's share shall be based only on a selected passphrase. The server's share shall be generated from the user's allocate and the encryption key. The security and conviction are achieved by performing both encryption and decryption on the client side. We also address the issue of countering dictionary attack by providing a further enhancement of the scheme.
Password is the most ordinary method for users to authenticate themselves when entering computer systems or websites. It acts as the first line of guard against unlawful access, and it is therefore critical to maintain the usefulness of this line of guard by strictly committed a good password management policy. This paper aims to grant a set of guiding principle and best practices for handling and managing passwords.

Keyword- Password Encryption, Password Storage, Identity Management, Secret Sharing

I. INTRODUCTION

Being the first valuable form of computer-based authentication, passwords are progressively more becoming a security problem in the modern age. There are an growing number of websites promising on the Internet, each demanding its own user id and password. A recent study reveals that Internet users, on average, have about 25 accounts that require password safeguard [1]. Users are affliction from "password fatigue" and this has led to some internet behavior on the part of users that makes them inclined to attack. As indicated in [2], users write down passwords (making them vulnerable to onlookers) or select common or easily-guessed passwords. Users need a more effective method for managing the passwords which they require. Optimally, it may be favorable to have a proper password management system. A straightforward solution would be to simply store the passwords in a certain location. Client password managers such as Robot form [3] keep the passwords on the client meaning that they can just be accessed when using the client machine. Browser-based password managers have been one of the most popular choices for user authentication and password management. Most popular browsers have provided a built-in feature for storing users' password (in encrypted form) and other login information into a database. However, as reported in [4], these password managers have vulnerabilities which can be exploited by attackers to decrypt passwords easily. It is also noted that recently, security analysis of some popular web-based password managers have been reported in [5,6], and security issues in auto fill polices, as well as vulnerabilities in one-time passwords, bookmark lets, and shared passwords have been identified. It would be more most favorable to store the passwords in a more accessible place, such as a web server. The problem with this approach is security and trust, as a user may not wish the server to have access to their plain passwords. This paper proposes a new scheme to assist with storing encrypted passwords on a server. The encryption key is split into a share derived from a user-selected passphrase and a share residing on a server. Only when both shares are combined can passwords be decrypted. Even though the data resides on the server, the processing is not performed by the user which means that the server does not have access to the plaintext passwords, nor does it have the potential of deriving them. Hence, even if the user passphrase is compromised to an attacker, the attacker will still need to bypass authentication to the server to decrypt passwords. The basic model in which this scheme will reside involves three parties: a user, a server, and a web service. The user will wish to store passwords and other login information on the server to be retrieved when it is needed. The user will then forward this data to the web service for processing. In practice, the web service will usually be a website which the user accesses, or some enterprise server. This will be of particular use as industry moves toward cloud computing.

The basic data flow in this model is as follows (Figure 1):
1) User and server authenticate themselves to each other.
2) Server sends data wanted to compute the password to the user.
3) User processes server data and sends to web service.

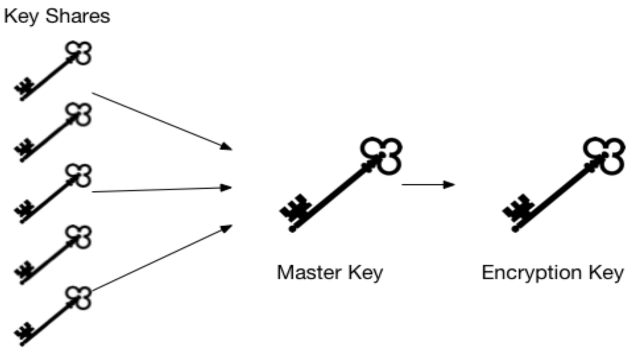

Figure-1:

The authentication between consumer and wine waiter can be achieved by several different means. This may involve certificates, user id and password, two-factor authentication, or any other ways of network layer security protection. This step is out of scope for this paper. In the third step, the password data is forwarded to the web service. In practice this could be something such as a website or enterprise server. This step is dependent on mechanisms which already exist and thus is also out of scope for this paper. We will note here, though, that the security of this model will only be as good as the security of the third step. For example, if passwords are sent in the clear from the user to the third party, then they can be easily viewed by attackers despite of any security mechanisms in place in the other two steps. The second step is the one in which the split-key algorithm will be effectual. The passwords would need to be stored and retrieved from the wine waiter by the user after flourishing authentication. There are many different functional and security necessities for this step, including a essential security feature of performing both password encryption and decryption on the patron side. From the detailed description of these functional and security necessities and their execution we shall see that several security issues are addressed in this model [7]. Some ideas of this paper were patented in 2010 and the patent was approved in 2014.

II. BACKGROUND

In today's multifarious IT environments, users normally have multiple passwords to manage for a range of applications or access levels. 80% of all users have 3 or more passwords to manage, but many more are possible depending on their system access needs. Users end up squiggle passwords on note paper, thus hostile the security of the system, or they are simplifying life by reusing passwords or choosing weak ones where the system permits, even though these are unsafe security practices.

Meanwhile, security-minded IT organizations often implement strong password measures which accidentally add to the burden, requiring users to recall complex formulations and change them frequently. Compliance with auditing and regulatory requirements, such as Sarbanes-Oxley and HIPAA, further tighten the password security net and in turn, further confound the user.

Quite simply, password practices that improve security are by their nature taxing to the user, resulting in passwords difficult to remember which are often changed about the same time they have finally become memorized. Yet password security remains a keystone of system security: as much as 80% of security breaches take place not through arcane hacking and virus attacks, but through system infiltration facilitated by use of a password.

It remains in the best interests of the organization to have strict password requirements that militate against system access by antagonistic parties. These are also the kinds of passwords best designed, it seems, to be elapsed by busy and woolly users.

Given this environment, it is no disaster that diligence analysts find that 30% of all help desk calls across the industry are about password issues, at a cost averaging $30 to $60 per call. This support cost when multiplied across an organization can be huge. While necessary to system security, troubleshooting passwords is large times descend for system administrators or help desk workforce. This low-level, cyclical activity siphons costly support resources away from higher-priority or more productive tasks.

- Self-Service Reset Utilities
- User corroboration
- The Reset Process
- Benefits
- Help Desk
- Compliance
- Strategic Service and Best Practices

III. PASSWORD SECURITY AND TECHNICAL DESIGN PROBLEMS

Early password security systems classified user picking to upper case and numerals, thus giving the attacker a much reduced space of attack (the permutations and combinations of valid input data). Later password security systems used better and lesser case and this improved things a bit in terms of the number of attempts the attacker had to make before he could find it by 'brute force'(still not all eight bits of each byte since not everything is on the keyboard).

Afterwards password security systems renewed the password into a 'hash' or one way encrypted field so that it could not be willingly reverse engineered by an attacker. Unfortunately the hashing systems were not necessarily very effective, and even when they were, the amount of space they give you is not that large and the attacker can choose any password that gives them a valid hash, not just the single the user selected.

Please note that when passwords are used other own (that is without a separate Identity field), the attack space is abridged by the number of passwords that have actually been issued, since for the attacker any official password is good enough.

Even afterwards some restrained password security systems combined the user id and the password into a hash. This created the potential for more space, although the length of both parts and the way that they were combined was critical to the quality of the result.

Network systems and services, and the introduction of the PC as a networked device as well as a stand-alone computer, together created the idea that it must be promising to have unbounded retries at getting the password right. (In the case of the PC, concern was focused upon the problem of having its owner get locked out with no way to recover the situation. Therefore, some password security systems had physical password reset buttons to get round this problem.)The attacker was being given a massive advantage!

The Internet, built for hardiness and information connection, integrated the idea of an ID / password security, but did not provide encryption to guard the password and allowed infinite retries to get it right. As a result, passwords are usually transmitted unprotected, and may be sent with every page that needs access to a password protected area as well as allowing the attacker all the time the site is up to try and splinter it.

IV. THE REAL PURPOSE OF PASSWORD SECURITY

The password, as we use it these days, is more often than not the 'secret' that unlocks systems capabilities or grants authorizations (including access control). In future services it will be used to authorize cryptographic secrets, most likely held in software, and then later in hardware. These 'key stores' may hold various secrets, conceivably even plus other passwords that are transparent to the user. Where infinite retries are possible, the use of short passwords will represent a considerable and preventable weakness which designers may one day be called to account for.

Ultimately, the real purpose of a password security system is to try and make the user's life easy whilst making the attacker's life difficult. Password security systems that ignore the user are going to fail with the very community they are supposed to serve.

Every time users cannot manage the password security systems they are given, an advantage is being given to the attacker because they will exploit those aspects of the system first. Similarly, a poorly designed password security system will fail and will compromise the very users it is supposed to protect. Poor design is much harder to fix than bad coding or errors in implementation.

V. SHORTING-COMINGS OF EXISTING PASSWORD BASED AUTHENTICATION

Password and PIN-based user authentication have numerous deficiencies. Regrettably, many security systems are designed such that security relies entirely on a secret password. Cheswick and Bellowing point out those weak passwords are the most common cause for system break-ins.
The main flaw of knowledge-based authentication is that it relies on precise recall of the secret information. If the user makes a small error in entering the secret, the authentication fails. Unfortunately, precise recall is not a strong point of human cognition. People are much better at imprecise recall, particularly in recognition of previously experienced stimuli.

The human restriction of precise recall is in direct conflict with the requirements of sturdy passwords. Many researchers show that people pick easy to guess passwords. For example, an early study by Morris and Thompson on password security found that over 15% of users picked passwords shorter or equal to three characters. Furthermore, they found that 85% of all passwords could be trivially broken through a simple exhaustive search to find short passwords and by using a dictionary to find longer ones. They describe an effort to counteract poor passwords, which consists of issuing random pronounceable passwords to users. Regrettably, the random number generator only had 215 distinct seeds, and hence the resulting space of ``random" passwords could be searched quickly. Klein conducted a wide-reaching study of password security in 1989 and notes that 25% of all passwords can be broken with a small dictionary.

Other notable efforts to design password crackers were conducted by Filmier and Kern and Muffed. Because of these password cracker programs, users need to create unpredictable passwords, which are more difficult to memorize. As a result, users often write their passwords down and ``hide" them close to their work space. Strict password policies, such as forcing users to change passwords periodically, only increase the number of users who write them down to aid memo ability.

As companies try to increase the security of their IT infrastructure, the number of password protected areas is growing. Simultaneously, the number of Internet sites which require a username and password combination is also increasing. To cope with this, users employ similar or identical passwords for different purposes, which reduce the security of the password to that of the weakest link.

Another problem with passwords is that they are easy to write down and to share with others. Some users have no qualms about revealing their passwords to others; they view this as a feature and not as a risk.

The mainstream of solutions to the problems of weak passwords falls into three main categories. The first types of solutions are proactive security measures that aim to identify weak passwords before they are broken by constantly running a password cracking programs. The second type of solution is also technical in nature, which utilizes techniques to increase the computational overhead of cracking passwords. The third class of solutions involves user training and education to raise security awareness and establishing security guidelines and rules for users to follow.

VI. CHALLENGES

Most current password security systems for the Internet are faulty. Designs that were almost acceptable 10 and 15 years ago have not been updated. Instead of moving to integrating authentication services under a cryptographically sound approach the IT industry has continued to proliferate multiple incompatible systems. Users are increasingly exposed by suppliers who feel no pressure to do anything better. There are parallels with the situation where web site page design methods are increasingly being rejected by security software because they represent known security weaknesses that have been exploited by hackers and viruses.

According to some studies, 85% of cyber attacks are performed through stolen credentials, key loggers, social engineering, and phishing attacks-all target passwords. Even complex password policies and password rotation policies have failed to provide security against sophisticated attacks designed to target password.

The password security also related to some other challenges like *The challenges of Internet, Human nature, The cost of forgetting password* etc.

The Internet was created for flexibility and information sharing, and it included very early on the concept of an ID and password security system, but did not provide the necessary encryption to protect them. Consequently, passwords are usually transmitted unguarded and could smooth be send with every page that needs access to a password protected area, meaning an attacker is chiefly left continuous to try and crack it while the site is live.

You could pick a series of complex passwords for a number of different apps, thereby making it ‘strong’ in terms of it being guessed. However, the risk is that if a site is hacked and the website or server doesn’t store passwords in an encrypted format, then your personal details and corporate data are compromised. Even if passwords are encrypted, they can be stolen and the encryption can be splintered.

Now, if we see the next challenge, i.e., human nature, we see that today each and every time we sign up at a new website, open a new app on our mobile device, or log in at work we are confronted with the challenge of what we should enter as a password. And here, our human nature comes into play – and in the process the inherent weakness of existing password protection is revealed.

Often, the easiest route is to pick a simple password that is easy to remember or we use an existing password. This means a user can access business applications and systems faster. However, this is where the issue lies. Passwords that are easily entered and remembered are necessarily weak as they can be second-guessed and therefore compromised by a hacker, thus presenting another fundamental flaw.

The cost of forgetting or losing password may be very high. There are also those who still sit at their desk with the password for their corporate network on a Post-It for all to see. Lose this and you will then need to contact the IT administrator for a password reset – costing both time and money to fix. And then there are those who choose to create a mental algorithm as a password. But these are easily guessed and, since we’re all still human, the chances are the user may have created an algorithm they minimally fail to remember.

As if this is not challenging enough, computing power has increased so much that a simple graphics card can crack a strong password.

To counter the user’s attempt to make their own lives easier, password security systems adapted to ensure that passwords themselves were changed on a regular basis, compelling the user to create a new and different password, checked against a list of formerly used ones. More sophisticated passwords have now been developed with enforced rules requiring them to be structured using letters and digits in non-repeating patterns. But the password itself still exists. What also still exists are the costs associated to the business when people forget their passwords.

VII. LITERATURE SURVEY

Kenneth Giuliani, V. Kumar Marty, Gangway CSU [9] Password management by storing password encryptions on a server, The method involves having the encryption key split into a share for the user and one for the server. The user’s share shall be based solely on a selected passphrase. The server’s share shall be generated from the user’s share and the encryption key. The security and trust are achieved by performing both encryption and decryption on the client side. We also address the issue of countering dictionary attack by providing a further enhancement of the scheme.

Keyed Parmer, Dervish C. Pinwale [10] in wireless sensor networks, secure data aggregation protocols target the two major objectives, namely, security and en route aggregation. In this paper, we propose an integrity and privacy preserving end-to-end protected data aggregation protocol. We use symmetric key-based homomorphism primitives to provide end-to-end seclusion and end-to-end integrity of reverse multicast traffic. As sensor network has a non-replenish able energy supply, the use of symmetric key based homomorphism primitives improves the energy efficiency and increase the sensor network's lifetime. We moderately evaluate the performance of the proposed protocol to show its efficacy and efficiency in resource-constrained environments.

Elman Alaric, Noah Alsulami, Omar Batarfi [11] Dragonfly is Password Authenticated Key Exchange protocol that uses a shared session key to authenticate parties based on pre-shared secret password. We proposed an alternate enhancement to keep this protocol secure without any extra computation cost that was known as "Enhanced Dragonfly". This solution based on two-pre-shared secret passwords instead of one and the rounds between parties had compressed into two rounds instead of four. We prove that the enhanced-Dragonfly protocol is secure against off-line dictionary attacks by analyzing its security properties using the Scythes tool. A reproduction was developed to measure the execution time of the enhanced protocol, which was found to be much less than the execution time of patched Dragonfly. The off-line dictionary attack time is extreme for few days if the dictionary size is 10,000. According to this, the use of the enhanced Dragonfly is more efficient than the patched Dragonfly.

Santana Chatterjee, Sandip Roy, Ashok Kumar Das, Samiran Chattopadhyay, Neeraj Kumar, Member, IEEE, and Athanasios V. Vasilakos, Senior Member, IEEE[12] Multi-server authentication architecture, user can manage authentication to various servers using single identity and password authentication scheme for multi-server environmet In this scheme, the Chebyshev chaotic map and biometric verification along with password verification for authorization and access to various application servers. The proposed scheme is light-weight compared to other related schemes. Only use the Chebyshev chaotic map, cryptographic hash function and symmetric key encryption-decryption in the proposed scheme. In this scheme provides strong authentication, and also supports biometrics & password change phase by a legitimate user at any time locally, and dynamic server addition phase. We perform the formal security verification using the broadly-accepted AVISPA (Automated Validation of Internet Security Protocols and Applications) tool to show that the presented scheme is secure. In addition, we use the formal security analysis using the Burrows-Abadi-Needham (BAN) logic along with random oracle models and prove that our scheme is secure against different known attacks. High security and significantly low computation and communication costs make our scheme is very suitable for multi-server environments as compared to other existing related schemes.

Ari Juels | Cornell Tech Thomas Ristenpart | University of Wisconsin[13] Honey Encryption Encryption beyond the Brute-Force Barrier, The problem of pathetic passwords prompted us to introduce honey encryption (HE) HE creates a ciphertext that, when decrypted with an incorrect key or password, yields a valid-looking but bogus message. So, attackers can't tell when decryption has been successful.

Bruno Blanchet [14] Automatically Verified Mechanized Proof of One-Encryption Key Exchange In this paper, He present these extensions, with their application to the proof of OEKE. All steps of the proof, both automatic and manually guided, are verified by CryptoVerif. Keywords.

Joseph Bonneau[15] The scope of proposals He survey is also extensive, including password management software, federated login protocols, graphical password schemes, cognitive authentication schemes, one-time passwords, hardware tokens, phone-aided schemes and biometrics. Our comprehensive approach leads to key insights about the difficulty of replacing passwords.

VIII.CHARACTERISTICS OF A GOOD PASSWORD

A good password may have many characteristics, but some of them are necessary to make a password good or we can say tough to be hacked. The characteristics are such as:

- 12 characters or more
- Mixed and Matched Caps, Symbols and numbers
- No obvious substitutions
- Neither in the dictionary nor a name, number or address
- Neither saved by the browser nor shared with anyone
- It should be unique

IX. SINGLE PASSWORD VS MULTIPLE PASSWORDS

From the users perspective, memorising one solitary password is easier than managing multiple passwords, even if the single password is a complicated one . In addition , if only one password is enough to authenticate all systems, there is higher consciousness among users in protecting their passwords. However, using a single password for all systems might not be technically feasible, in particular on legacy systems, or across multiple operating system platforms.

For systems that a user accesses only occasionally, it is quite possible for the user to forget a rarely used password. This generates increased workload for support staff who have to reset passwords. In addition, users tend to find ways to bypass difficult controls, such as writing down passwords, or selecting a weak and easy - to - remember password.

For attackers, the single - password approach means that all systems will automatically be compromised once passwords in a weakly protected system are successfully hacked.

Therefore, when an organisation decides to use the single - password approach, all systems must be protected at the same level of security.

X. TECHNOLOGY FOR PASSWORD MANAGEMENT

Apart from implementing a security policy and guidelines to enforce good password management, some of the technologies highlighted below offer effective and user-friendly password management.

PUBLIC KEY INFRASTRUCTURE

Public Key Infrastructure (PKI) is a technology that uses mathematical algorithms and processes to facilitate secure transactions by providing data confidentiality, data integrity and authentication. PKI makes use of digital certificates to provide proof of identity for the individual. A digital certificate is a kind of digital document that binds a public key to a person for authentication, rather like a personal identity card. A trusted Certificate Authority (CA) creates the certificate and digitally signs it using the CA‟s private key, thereby authenticating the identity of the requestor. A person can use his or her certificate for authentication with different applications, and the applications then check the user‟s identity by verifying the digital signature with the issuing CA.

PKI is particularly useful for user authentication in on-line transaction and public applications, because there is no advance pre-registration process required for each application. Users only need to apply for a certificate from a trusted CA to authenticate themselves with various applications.

Deploying PKI requires some worth noting security considerations as follows:

1. The private key must be protected and stored in a safe place, such as in a security token or smart card secured by a PIN.
2. Relevant password restrictions should be imposed on the PIN of the security token / smart card to prevent unauthorised access to the private key inside.
3. There should be proper procedures in place to handle key life - cycle management, issuing a nd revoking of certificates, storing and retrieving certificates and CRLs (Certificate Revocation Lists).
4. For private key backup, the key must be copied and stored in an encrypted form and protected at a level not lower than that of the original private key.
5. As not all applications support the use of PKI, there may be interoperability issues.

SINGLE SIGN-ON

With the use of Single Sign-On (SSO) technology, users are able to identify themselves with the authentication server only once to access a variety of applications, including both internal and external systems. Users can enjoy the benefit of choosing one password to access multiple applications, instead of memorising many different passwords. However, compromise of one authentication event could result in the compromise of all resources that the user has access rights to.

Implementing SSO requires the following worth noting security considerations:

1. As one single authentication controls access to all resources, it is important that the authentication process is secure enough to protect those resources. This protection should satisfy the requirements of the most critical application. The single authentication process should not be weaker than the original authentication method used by the various applications, otherwise, the result is a downgrade in security level.
2. A second factor of authentication, such as a security token and smart card, can be used to strengthen the authentication process.
3. Relevant password restrictions, such as the minimum password length, the password complexity, the maximum number of trial attempts and the minimum time for renewal, and so on, should be imposed.
4. As the authentication server may become an attractive target for attack, it should be well protected so that intruders cannot access authentication information which could then be used for unauthorised access to all the systems.
5. Auditing and logging functions should be used to facilitate the detection and tracing of suspicious unsuccessful login attempts.
6. Encryption should be used to protect against authentication credentials transmitted across the network.

ONE TIME PASSWORD TOKEN

Another technology that may be used to facilitate password management is the one – time - password token. Users authenticate themselves with two unique factors, something they have (the token) and something they know (the PIN). Users do not need to choose or memorise passwords. The token will generate a unique, one – time - use password for each authentication process, based on the PIN and other factors, granting access to protected resources.

The following are some considerations when implementing one-time-password tokens:

1. A token is needed for each user of the authentication process, which implies additional investment.
2. Users must carry the token at all times, and they will not be able to access the system if they lose the token or forget to bring it with them. Unlike software - based access control systems, which only require a password reset, users may not be able to use the system for hours or days if the token is lost.
3. Users should be aware of the physical security of the token and ensure that the token is properly protected at all times.
4. Most of the current one -time-password authentication schemes only authenticate the initial connection. Connections thereafter are assumed to be authenticated, and these connections are susceptible to being hijacked.
5. Security tokens may not support all applications or servers.

XI. CONCLUSION

While password is the most commonly used method of authenticating users entering computer systems, passwords are frequently targeted by attackers wanting to break into systems. It is critical that this first line of defense against unauthorized access is effective by rigorously practicing good password management policies. Different passwords should be used for different systems with respect to the security requirements and the value of information assets the need to be protected.

Make use of other access control mechanisms to facilitate password management and reduce the effort required by users in memorizing a large number of passwords. This should be enforced with good security policies and guidelines, supported by user awareness training and education on the best practices in choosing and handling passwords.

In addition, for effective information security management, consideration should also be given in areas including but not limited to physical security, data and application security, network security, and technologies for strengthening security protection, such as firewalls, VPN and SSL.

REFERENCES

[1] Florêncio, D. and Herley, C. (2007) A Large-Scale Study of Web Password Habits. Proceedings of the 16th International Conference on World Wide Web, Banff, May 2007, 657-666. http://dx.doi.org/10.1145/1242572.1242661

[2] Hayday, G. (2002) Security Nightmare: How Do You Maintain 21 Different Passwords? Silicon.com.

[3] (2016) Roboform Reference Manual. Siber Systems Inc.

[4] Zhao, R. and Yue, C. (2013) All Your Browser-Saved Passwords Could Belong to Us: A Security Analysis and Acloud-Based New Design. Proceedings of the 3rd ACM Conference on Data and Application Security and Privacy, San Antonio, February, 2013, 333-340. http://dx.doi.org/10.1145/2435349.2435397

[5] Silver, D., Jana, S., Boneh, D., Chen, E. and Jackson, C. (2014) Password Managers: Attacks and Defenses. 23rd USENIX Security Symposium (USENIX Security 14), San Diago, August 2014, 449-464.

[6] Li, Z., He, W., Akhawe, D. and Song, D. (2014) The Emperor's New Password Manager: Security Analysis Ofweb- Based Password Managers. 23rd USENIX Security Symposium (USENIX Security 14), San Diago, August 2014, 465- 480.

[7] Haque, T., Wright, M. and Scielzo, S. (2013) A Study of User Password Strategy for Multiple Accounts. Proceedings of the 3rd ACM Conference on Data and Application Security and Privacy, 173-176. http://dx.doi.org/10.1145/2435349.2435373

[8] Giuliani, K. and Murty, V.K. (2014) Split key Secure Access System. U.S. Patent No. 8,892,881.

[9] Kenneth Giuliani1, V. Kumar Murty1, Guangwu Xu2 Copyright © 2016 by authors and Scientific Research Publishing Inc. . http://www.scirp.org/journal/jis http://dx.doi.org/10.4236/jis.2016.73016

[10] Keyur Parmar, Devesh C. Jinwala http://file.scirp.org/pdf/JIS_2015010814240810.pdf

[11] Eman Alharbi, Noha Alsulami, http://file.scirp.org/pdf/JIS_2015031214001850.pdf

[12] Santanu Chatterjee, Sandip Roy, Ashok Kumar Das, Samiran Chattopadhyay, Neeraj Kumar, Member, IEEE, and Athanasios V. Vasilakos, Senior Member, IEEE[12]

[13] Ari Juels | Cornell Tech Thomas Ristenpart | University of Wisconsin Honey Encryption Encryption beyond the Brute-Force Barrier,

[14] Bruno Blanchet Automatically Verified Mechanized Proof of One-Encryption Key Exchange

[15] Joseph Bonneau The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes 2012 IEEE Symposium on Security and Privacy