A SIMPLISTIC MOBILE SOLUTION FOR PASSWORD SECURITY

Sahana M D[1], Sujana[2], Sumera Asma[3], Ranjan J[4], Mahesh[5]

[1,2,3,4]TR Department of Computer Science and Engineering, T John Institute of Technology, Bengaluru, India
sahani.sahanamd@gmail.com, sujanaachari31@gmail.com, asmasumera@gmail.com, jbranjan@gmail.com
[5]Professor and Head, Department of Computer Science and Engineering, T. John Institute of Technology, Bengaluru, India
mahesh@tjohngroup.com

Abstract

Authentication through passwords are the most common method for a person to login into a remote server. However, they are very frustrating when it comes to security. Every server asks their users to avoid common passwords which is easy guess by snoopers. Often the usersare required to enter a combination of alphabets, numbers and special characters. Due to which more often than not users usually reuse the same passwords for all servers. As a matter of fact users have fewer than three passwords set up all their online log-ins. Even if one of those master passwords are compromised the user will need to go through a hectic procedure to retrieve their passwords where in most likely they would need to enter a new passwords since the old one was compromised. In this paper, we introduce an application that amazingly uses all the different methods of user authentication to come up with a unique way to store and retrieve the users passwords for utmost security. Each password is encrypted with a unique key for utmost security. To access the passwords which are stored in this cryptic format on the server, the user needs to only say the predefined words set by the user himself and the application will detect the speaker and give him access. The user also needs to go through another authentication system where he needs to set a color pattern to detect the pattern based on the colors. Once this is done, the user will be able to access his secured data which comes from the secured server in an encrypted format and is decrypted on the users phone. The user does not need to trust the server since every data is stored in an encrypted format and cannot be viewed by the server administrators.

Keywords: *Chip Multi Processor, Scratch Padmemory, Genetic Algorithm, Data Allocation.*

----------***----------

I. INTRODUCTION

Authentication through passwords are the most common method for a person to login into a remote server. However, they are very frustrating when it comes to security.Often the users are required to enter a combination of alphabets, numbers and special characters.Due to which more often than not users usually reuse the same passwords for all servers.As a matter of fact users have fewer than three passwords set up all their online log-ins. Even if one of those master passwords are compromised the user will need to go through a hectic procedure to retrieve their passwords where in most likely they would need to enter a new passwords since the old one was compromised.Unfortunately, these passwords are usually not unique and hence they are easy to guess for snoopers.

Password manager applicationswere the solution the users went to. These applications were used to store our passwords and never rely on our untrustworthy memory to do the job. In this way users could have different and a unique password. Generally, three classifications of password managers come into mind. The first category isours desktops local memory. This could be done by storing our passwords on our desktops and checking up the directory it was stored in when we needed to check for any passwords. The second category is one where the users opted for trusting third party cloud servers to manage their passwords and storing it on the cloud. The third category is the most rudimentary method of storing passwords on our physical devices like pen drives, hard disks and flash drives.

There is a serious threat with all three methods. Hence, we introduce an application that amazingly uses all the different methods of user authentication to come up with a unique way to store and retrieve the users passwords for utmost security. Every password stored by the user on our server will be encrypted with a fresh key for utmost security. The passwords are stored in our servers in an encrypted format and does not reveal any user information to the system administrators. The user also can backup these encrypted passwords on to any cloud service if they wish to. To retrieve a password stored, the user only has to go through two authentication screens where the user sets a set of words and utters the names using which the application will recognize the speaker and take them to the next screen. The next screen will contain a color pattern also predefined by the user and once the user enters his pattern the user is logged in. Now the user has to either utter the name of the server whose password he requires or key it in and our adaptive algorithm will figure out the password the user requires and fetch it from the server in the encrypted format which it is stored in and decrypt it on the phone of the user. Then the user will be able to view the password and use it on the server while the set password on our server is still stored in an encrypted format. One of the most important service that our application provides is that it gives an option to the user to view the password of the service he

wishes to and not all the passwords set the user are shown. Another huge advantage of this application is that the user wont have to trust our server or any other cloud storage since every single data is stored in an encrypted format wherein which even our administrators cant view the passwords.

II. EXISTING SYSTEM

The Password Managers which were to be the solution to the problems caused by traditional passwords were good to an extent that they would offer the users the ability to save and store any password for any server they needed. This was considered to be really good for a while since all the passwords could be stored and saved on a server and even if the users forgot the passwords it was not a problem. This soon changed since people started realizing that passwords were surely stored on the servers which were offering this ability but who is to say that those people offering these services could not misuse the passwords set by users. When people started to realize that they need to save their passwords on an untrusted third party server just cause they could not remember the passwords they came out off these services.

These password managers gave all the flexibility and features to the user except one most important feature, the proof that they are storing the users data in a secured fashion. Now that storing passwords on these untrusted servers was not an option people started to use two other methodologies to store their passwords so as to remember them. One was to store them on their desktop computers, locally. Other was to store the passwords on their physical drives such as hard drives, USB, flash drives and so on. This still was not trust worthy since storing passwords locally would cause hackers to easily hack into the users vulnerable systems and steal their passwords. On the other hand storing the passwords on the physical devices meant thecomplicated problem of carrying them around everywhere the users went to keep them safe. Now who is to say that the users physical devices will not get lost or stolen. Then what?. These questions are still unanswered since there is no proper solution for them. All the three methods of safe guarding the passwords mentioned above are vulnerable to one or another type of attacks. Passwords stored on untrusted servers, passwords stored on the users vulnerable desktops and passwords stored on the physical devices which can be stolen or lost very easily are not an option to keep the users passwords safe from being compromised. An intelligent way to save passwords would be if any service said that they would save their passwords in an encrypted format and provided proof for it. This will be explained in the next section.

III. PROPOSED SYSTEM

In the proposed system, we introduce a peculiar approach. In the journey of providing users their privacy we have come up with a new application which does all the work that those untrusted password managers do but better since we provide the user the security they crave for. Our mobile based application uses all the different methods of user authentication. Every password stored by the user on our server will be encrypted with a fresh key for utmost security and the user also has an option to upload these encrypted keys onto the any cloud of the users choice. Every single key is stored in an encrypted format on our server. To retrieve the keys, the user will need to go through two authentication systems. First being speech and speaker recognition. Then the user will need to go through a color patter screen where he will key in the colors that he chose during registration. After which only the passwords related to the server that the user wishes to retrieve will be sent back to the users phone. The application retrieves the keys from the cloud server and uses them to decrypt the passwords obtained from the server. The user in this case does not need to trust our server or any third party server for that matter since every single element is stored on our servers in an encrypted format such that even the system administrators can not view the data.

User's sensitive data are always encrypted with a fresh key and stored both on the device that the user uses and backed up in a cloud of user's choice. Even if this is compromised it reveals nothing about the users profile since every key stored will be kept in an encrypted format and nobody has control over it except for the user himself. This is one of the main reasons why the users would nothave any trust in our servers. This is again because the keys stored on the server reveal nothing about the users profile and the passwords are encrypted with the help of keys that are stored on the users cloud again in an encrypted format. So this indirectly means that the passwords are double encrypted when it is placed on our servers. Now this is all the more reason for the users to never have to trust our servers or even their own cloud servers of choice.

This brings us to the question that what if our server and the cloud collude? The answer is that even if this were to happen by any chance, no information about user's data is revealed. This type of distribution of storage gives a very high security to the users data.

The most important advantages of our application is that we combine the best of all the approaches. Secondly, since every password has been encrypted completely with keys that keep changing, users can enjoy the versatility of our application with no compromise when it comes to security. Thirdly, the keys are stored in a cryptic format on our servers and only can be decrypted on the users phone using our decryption algorithm we created with the application. Fourthly, the server only provides the keys to the user only when the user goes through a couple of authentication schemes created by us. Fifthly, the users enjoy the pleasure of communicating with our servers to retrieve their password without typing since that is also an option for them at their disposal. Sixthly, the keys in their encrypted format show nothing about the users data.

The below diagram shows the entire architecture which the application runs on.

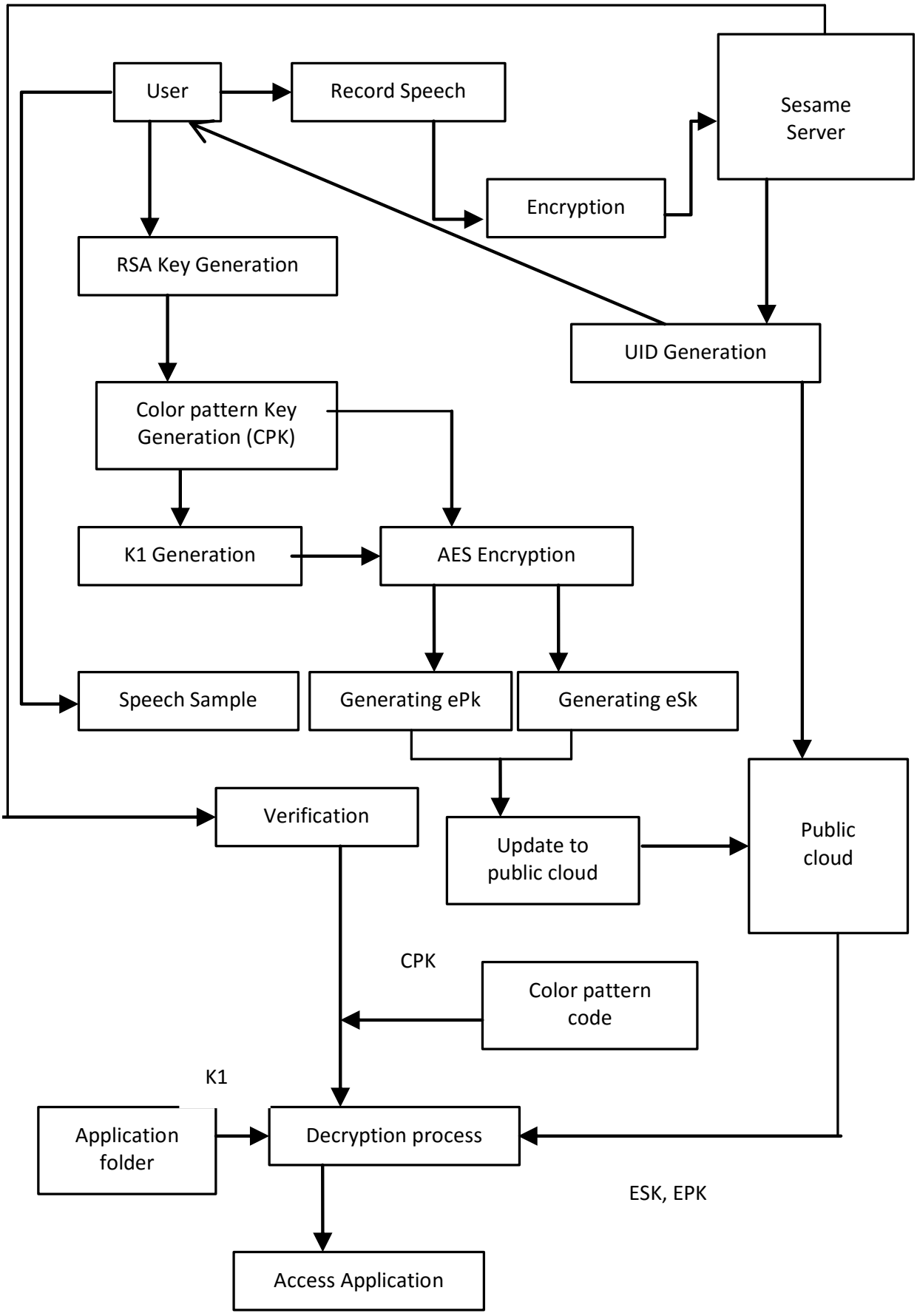

Fig 3.1: System Architecture

IV. METHODOLOGY

In this work, we explain the complete methodology of how the system is working. In the above system architecture we have seen that the user records an entire set of words for his future authentication. Those words which are recorded are sent to the server in an encrypted format. We use the key-logger algorithm to encrypt the voice sample on the server. Now that is done, the server then generates a unique user id called UID and sent back to the application. The user then uses that to enter into the next phase of authentication called color pattern generation. Now, while the user is doing this, on the background an RSA algorithm is run to generate two keys namely secret key and public key. Below is the algorithm used for generating the secret key and public key.

- Let n = p×q, where p and q are primes. Note that n is a composite number.
- Let **M** = **C** = Z_n = {0, 1, 2, …, n−1}.
- **K** = {(n, p, q, d, e): e × d ≡ 1 (mod ϕ(n))}.
  - We will see that ϕ(n) = (p−1)(q−1).
- For K = (n, p, q, d, e), define
  - $E_K(m) = m^e \bmod n$, and
  - $D_K(c) = c^d \bmod n$, where m, c ∈ Z_n.
- The (n, e) comprise the "*public key*."
- The (p, q, ϕ(n), d) comprise the "*private key*."

Fig 4.1: RSA Algorithm

This step goes through on the background of the users application. So, while the public and secret keys are generated, the user has the task of setting up his second authentication scheme which is a color pattern generation. In this step, the user has his eyes set on making a pattern by remembering the pattern of colors of his choice. Here, the user is presented with a screen wherein he would have to chose a set of four colors and also remember the pattern of those colors. Once the screen pops up for the user he will be asked to choose a set and remember them for future use since he will have to go through this every time he tries to login to the application. This layer is just added to the application for extra security. The catch here is that the next time the user tries to login to the application, the colors that he selected do not appear at the same position when he registered. Hence he will need to find the colors on any part of the screen and login using that. This is done for extra security of our users details. Now that this is done the next step is that the public and secret keys that the RSA algorithm generated will be moved onto the public cloud in an encrypted format.The public cloud is any cloud server of the users choice. A 256-bit AES Encryption is used to encrypt pk and sk. For encrypting pk and sk another key K1 is generated using another algorithm called the key logger algorithm. Once K1 is generated, we use K1 to encrypt pk and sk using AES encryption. This encrypted versions of pk and sk are denoted as epk and eskare sent to the public cloud. Now the user is presented with the next screen where he can enter the server whose password he wants to store and can proceed with storing the passwords. These passwords once set for a specific server will be encrypted and sent directly to the server to be stored. Once this phase is done and all the passwords are stored by the user, the next time user tries to login and view the passwords he has stored, the application presents him with two authentication screens. First one being the words that he has set and second being the color pattern. When the user speaks any of the word he had stored at the time of registration to get past his first authentication screen the application retrieves the words that the user has stored in the server and decrypts it on the application and then verifies it for authenticity with what the user has told. The catch here is that the user is allowed to only say one of the many words he has set and that word cannot be the word he said during his last login. This word will be kept in a different array to be verified against then next time the user logs in. The next authentication screen as said before is the color pattern generation. Here the user is said to choose the four colors and in that patter in which he had set during the registration phase. These colors will not be in the same order or the same manner as it was during registration and will require the user to be vigilant in finding the colors and the pattern on the screen. This is once again given for higher security to the users sensitive data. Once both the authentication screens are passed the next step for the user is to retrieve his passwords from the server. Again this is done with the help of the keys stored in the users public cloud. The keys epk, esk and uid are retrieved from the users cloud and K1 is used to decrypt it to obtain pk and sk. Once pk and sk are obtained the next step is to retrieve the passwords stored in the server. The user can either speak the server name or can choose the button provided to them to select the server whose password he requires. Once he selects pk and sk which were decrypted are used to again decrypt the application passwords that were stored on the server. Now the application displays the password of the user for that particular server. The best part about this technology is that the user will be able to view only those passwords which he can wants to view. Others are not retrieved from the server. Once the user views his password he can log out and close the application and use the password to enter into the server to do his work. The next time the user tries to login into the application, the application presents the user with the same two screens of authenticationfor the user to go through and retrieve another password. This is the entire working procedure and methodology of the application.

V. CONCLUSION

Concluding our entire discussion on what this proposed application does we can safely say that an application like this has not yet been put into existence. Password manager applications are the most common way to store passwords than before. Also they allow us to have a stronger and a very hard to crack passwords for each an every web service. These strong passwords might make us susceptible to forgetting our passwords but that is the main reason why we have our password managers. We in this application have provided such a vault to store and keep our passwords in an encrypted and safe manner, away from prying eyes and eavesdropping ears. This application produces all the necessities that a user wants. Giving the user the feeling of

privacy by not having to trust any server or their cloud service of choice is a wonderful way of giving the user security. The concept of end-to-end encryption was an overrated concept and now it has become a common phenomenon that all users expect. We are one of the only ones to provide such a facility and are proud to give such a service to our worthy users. Now that the concept of privacy has been explained the only other thing to tell is the ease of use for any user. Ever single user will feel that this application is a piece of cake to use and will love using it. It has come out so well that the users will feel like coming back to the application even when there is no need to. The concept of this application is explained is simple and wonderful in its craft and something that has never been done before though some of them claim to. This application is the start of something revolutionary in terms of security.

REFERENCES

[1] Cookeyah.available online at. http://virtualwallet.cookeyah.com/.
[2] Keepass. available online at. http://keepass.info/download.html, 2014.
[3] Mobile technology fact sheet. available online at. http://www.pewinternet.org/fact-sheets/mobile-technology-fact-sheet/, 2014.
[4] Openssl project. available online at. https://www.openssl.org/.
[5] Splashid key safe. available online at. http://www.splashdata.com/splashid/keysafe/, 2014.
[6] Worldwide smartphone usage to grow 25% in 2014. available online at. http://www.emarketer.com/Article/Worldwide-Smartphone-Usage-Grow-25-2014/1010920, 2014.
[7] Anne Adams and Martina Angela Sasse. Users are not the enemy. Communications of the ACM, 42(12):40–46, 1999.
[8] Victorinox Swiss Army.Victorinox slim. available online at. http://www.swissknifeshop.com/victorinox-slim-flash, 2014.
[9] Speech at Carnegie Mellon University. Open source toolkit for speech recognition. available online at. http://cmusphinx.sourceforge.net/wiki/.
[10] Security Lab at CSULB. Sesame. available online at. https://play.google.com/store/apps/details?id=net.sesamepass&hl=en.
[11] Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas. Score normalization for text-independent speaker verification systems. Digital Signal Processing, 10(1):42–54, 2000.
[12] RSA Laboratories B. Kaliski. Pkcs #5: Password-based cryptography specification version 2.0. available online at. https://www.ietf.org/rfc/ rfc2898.txt, 2000.
[13] Jean-Franc¸oisBonastre, Fr´ed´ericWils, and Sylvain Meignier. Alize, a free toolkit for speaker recognition. In ICASSP (1), pages 737–740, 2005.
[14] Eran Gabber, Phillip B Gibbons, Yossi Matias, and Alain Mayer. How to make personalized web browsing simple, secure, and anonymous. In Financial Cryptography, pages 17–31. Springer, 1997.