http://www.casestudiesjournal.com/
Impact Factor: 4.428

# Passphrase Generation Methods: A Review

**Author's Details:**
**B.T Shehu[1], S.U Suru[2], T.M Emmanuel[3]**
[1]Department of Computer Science, Federal University Birnin Kebbi.
*bashartukurshehu@gmail.com*
[2]Department of Computer Science, Kebbi State University of Science and Technology Aleiro.
*surusalihu@yahoo.com*
[3] Sales and Distribution, MTN Nigeria
*Etimothy324@gmail.com*

***Abstract:***
*One of the most crucial concerns in information security is user authentication. Numerous authentication mechanisms have been created over time. Some of these authentication methods rely on the user's knowledge, while others are dependent on tokens (something you possess, like an ATM), and others are based on biometrics. The most common and popular type of authentication mechanism is the use of alphanumeric password (Knowledge-based). But overtime the alphanumeric password proved to be vulnerable especially when short and simple passwords are chosen. Passphrases, which are sets of natural language words separated by spaces, have long been advocated as secure and practical because they are longer than passwords. In this paper, we describe the categories of passphrase authentication scheme, the recent research on Passphrases Authentication Schemes and outlined a future research direction.*

***Keywords:*** *Authentication systems, passphrase, mnemonic, system-generated, user-chosen, security*

## 1. Introduction

The procedure of identifying the right user and giving them the right resources after verification is referred to as user authentication [10], [31]. Although textual passwords are frequently employed in user authentication nowadays, if a password is too short or based on a predictable pattern, it is likely to be cracked. Making passwords longer is one technique to increase their security. Longer passwords increases the password space which in turn mitigates the guessing attack. One type of longer password is a passphrase.

Passphrases have been discussed as a possible more enduring and secure alternative to short passwords in academic literature for the past three decades [22] . According to researchers [30] , passphrases are a group of words, either related (e.g., "mother chicken apple") or unrelated (e.g., "I love apple juice"). Passphrases typically consist of a string of words, often separated by spaces. The user can select the words, which might range from well-known phrases to obscure or made-up ones. These words could constitute a statement, such as "I love my mum," or they could be unconnected, such as "brain community coconut". Passphrases can contain statements that are known to a user (such as verses from a favourite song) which makes them more memorable than passwords. The trick is to pick terms that stick in the user's memory while being challenging for outsiders to decipher [14].

NIST defined a passphrase to be a memorized secret consisting of a series of words or other text that a user uses to authenticate their identity [9]. Intuitively, passphrases are likely to be easier to remember than passwords due to their closeness to natural language for users as well as harder to guess due to their length for attackers.

[25] demonstrated via a lab study that even employing passphrases consisting of just three to four words can be comparable in terms of entropy with passwords generated using more complicated approaches while also accounting for memorability, emphasizing the importance of strong passphrases. Passphrases are utilized in password managers, bitcoin wallets, and SSH key security today [19].

A passphrase's length and complexity can change based on the needs of the system it is being used for. A specified minimum length or a combination of uppercase and lowercase characters, numbers, and symbols may be necessary for some systems.

A passphrase has the benefit of being simpler to remember than a random string of characters [22]. This can lessen the chance that the user will write down or forget their password, which could jeopardize security.

**The passphrase construction techniques used in earlier study are described here:**

**User Generated Passphrase:** Without any extra restrictions, a user creates a passphrase, such as "I Love Apple Juice."

**System Generated Passphrase:** To create a passphrase, a machine picks a random word from a dictionary and concatenates it (for example, "Correct Horse Battery Staple").

**Mnemonic-Guided Passphrase:** A system generates a mnemonic alphabet, and the user selects a word that begins with each mnemonic (for example, given the mnemonic "ABALO," the user can form the passphrase "Apples bread and lox order"). Each technique for creating passphrases has a distinct.

## 2. Passphrase Generation Methods

### 2.1 User-Generated Passphrase

Researchers have used many construction techniques of generating passphrase. One of them is user-chosen passphrase. User-generated passphrases are a popular way of authentication and access control that rely on user-generated secrets.

A user-generated passphrase is a form of password that the user creates using a mix of words or phrases that are simple to remember yet challenging for others to guess or crack. they are typically longer and more complex than traditional passwords, and therefore considered to be more secure against brute-force attacks [4].

Despite their potential advantages, however, user-generated passphrases can also have some drawbacks. One issue with user-generated passphrases is that users are more prone to mistype longer passphrases. The fact that user authentication interfaces typically do not support character-by-character echoing makes the situation worse because users often do not catch their mistakes until they have completed lengthy inputs. To address this issue, [20] proposed a one-way hash function-based echo-back system; Instead of echo-off asterisks during the passphrase authentication operation, the monitor shows a chain of hashed values. The user may catch his mistakes before completing the entire typing process if he remembers some of the hashed data.

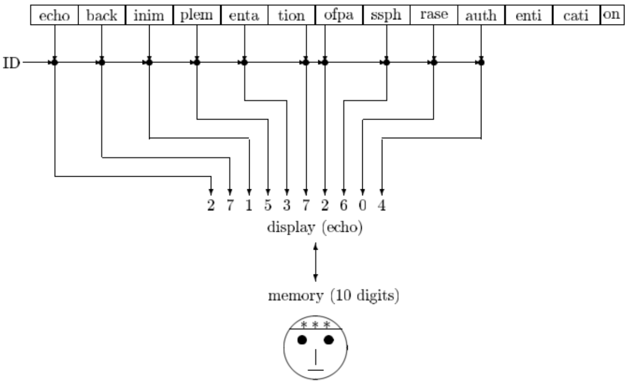


**Figure 1:** Echo-back system

Another issue with user-generated passphrases is that they are often predictable. To address this issue, various techniques have been proposed to encourage the use of more secure and unpredictable passphrases, such as the use of random words, or the use of a passphrase generator [8]. This predictability makes them vulnerable to dictionary attacks and other types of guessing attacks, as attackers can use lists of common words or phrases to try to guess users' passphrases [28].

Another issue with user-generated passphrases is that users may have difficulty remembering them, especially if they are too long or complex. To address this issue, research has explored various approaches for improving the memorability of passphrases, such as the use of mnemonic techniques, or the use of personalized passphrase policies that take into account users' individual characteristics and preferences [28]. The difficulty to remember can lead to users either writing down their passphrases, or reusing the same passphrase across multiple accounts, which can compromise their security.

Building a high-entropy, more memorable passphrase can be accomplished by letting users choose from a variety of random words. Choosing from a selection effectively discourages people from selecting favourite terms that are simple to predict [3].

When users are allowed to choose their own secrets, they can construct stronger secrets by using a security meter [25]. As they type, security meters provide the user with a visual cue on the strength of the secret, directing them in the right direction. According to Bauer et al. (2012b), strict security meters drive users to spend more time entering secrets, strengthening security. To measure how secure a secret is, there isn't a universal scale for measuring security, though.

[27] observed that the majority of users found passphrases to be rather challenging to utilize because they were unfamiliar with them. But as they get more experience, their actions and reactions could change.

Overall, user-generated passphrases can be a useful method for authentication and access control, but their security and usability can be improved through the use of more secure and unpredictable passphrases, as well as through the use of techniques for improving their memorability.

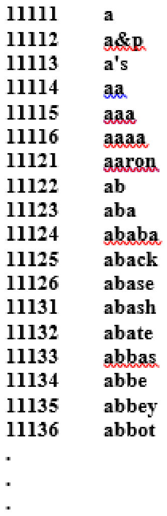
### 2.2 System-Generated Passphrase

Although studied to a lesser degree, system-generated passwords and passphrases have been considered as an alternative to user-generated passphrases and traditional passwords [25] .

This form of password is created by other software, not by the user. System-generated passwords typically lack easily discoverable linkages to specific individuals as a result. These passwords are generated without using any personal data [18].

In comparison to user-generated ones, system-generated passphrases provide a number of benefits. These passphrases tend to be longer, more random, and less predictable than those made up by users, according to researchers [5] . Increased security against brute-force and dictionary attacks is a result of the randomness that has been included. But according to [25] system-generated passphrases and passwords are irritating to users and ease to forget.

Early in the 1990s, Arnold G. Reinhold created diceware. Diceware is a method for creating passphrases, passwords, and other cryptographic variables by utilizing a regular dice to generate hardware random numbers. For each word in the passphrase, five rolls of the die are required. A five-digit number, such as 41134, is created by adding together the numbers from 1 to 6 that appear on the rolls. Using that number, the word is then located in a word list.



| | |
|---|---|
| 11111 | a |
| 11112 | a&p |
| 11113 | a's |
| 11114 | aa |
| 11115 | aaa |
| 11116 | aaaa |
| 11121 | aaron |
| 11122 | ab |
| 11123 | aba |
| 11124 | ababa |
| 11125 | aback |
| 11126 | abase |
| 11131 | abash |
| 11132 | abate |
| 11133 | abbas |
| 11134 | abbe |
| 11135 | abbey |
| 11136 | abbot |
| . | |
| . | |
| . | |

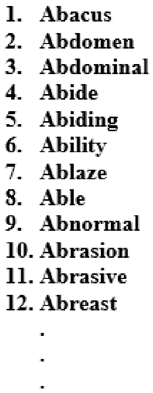
**Figure 2:** Diceware word list sample

Several words can be generated sequentially to create a long passphrase that is strong yet simple to remember passwords [24]. For many people and organizations, the work of Arnold G. Reinhold on Diceware has considerably improved password security.

According to [4], the Diceware list can provide strong security, but offers some challenges to usability. In particular, some of the words on the list can be hard to memorize, hard to spell, or easy to confuse with another word.

- It contains many rare words such as *buret, novo, vacuo*
- It contains unusual proper names such as *della, ervin, eaton, moran*
- It contains a few strange letter sequences such as *aaaa, ll, nbis*
- It contains some words with punctuation such as *ain't, don't, he'll*
- It contains individual letters and non-word bigrams like *tl, wq, zf*
- It contains numbers and variants such as *46, 99* and *99th*
- It contains many vulgar words
- Diceware passwords need spaces to be correctly decoded, e.g. *in* and *put* are in the list as well as *input*.

The Electronic Frontier Foundation (EFF) created and published the EFF Dice-Generated Passphrases, also known as the EFF Wordlists in 2011. These wordlists are a variation of the Diceware passphrase generation method and were created by the EFF to provide users with a set of words that could be used to generate strong and memorable passphrases.



1. Abacus
2. Abdomen
3. Abdominal
4. Abide
5. Abiding
6. Ability
7. Ablaze
8. Able
9. Abnormal
10. Abrasion
11. Abrasive
12. Abreast

.
.
.

**Figure 3:** EFF Word List sample

Coinware is similar to diceware except it uses coin to select from a word list that can be monolingual, bilingual, or multilingual. Each face of the coin is labelled as binary bit ‘0’ or ‘1’ respectively. Four coin values are used to derive a hexadecimal digit. It is especially efficient for word list in binary, octal, and hexadecimal orders. There are readily built word lists for Han characters in Unicode-encoded CJK languages [16].

[17] proposed a passphrase using semantic noises generalizing the punctuation marks, capitalization, permutation, mnemonic substitution, associative morphing, and misspelling. Passphrase with semantic noises has higher information rate, bigger unicity distance, and more spurious keys, which strengthens the login protection with limited attempts. They also provide an ASCII mutual substitution table and its proof on information rate increment is provided.

In order to make the Diceware passphrase generation mechanism even simpler to learn, implement, and utilize, [6] described a number of improvements; the smaller number of words & fixed-length words. They explained how our system with fixed words provides almost the same resistance to brute force attacks and claimed that it is similar to many existing password policies in widespread use, if not outright superior.

Other systems use partially system-assigned passphrases and passwords. For instance, Forget et al. insert randomness into user-chosen passwords to increase strength [7].

Several prior works focused on generating and remembering secure passwords (and passphrases) using techniques like contextual cues, portmanteau, or mnemonic based generation [13] , [15] , [32] . The most often used system passphrase generating technique is Diceware, according to [25]. Although Diceware is extremely secure, users have a very difficult time remembering the passphrases, meaning that its memorability is quite low [25].

Template-based Diceware, commonly known as TemplateDice, is a method that is currently being employed to improve Diceware passphrase memorability. TemplateDice algorithm has a dictionary of English words divided into different parts of speech tags and has 27 syntactic templates for the English language, whose components are the tags, embedded within the algorithm. The idea of using syntactic templates has handled the issue of memorability very well.

Passphrases in TemplateDice are produced using predefined English syntactic templates. The templates consist of different parts of speech including nouns, verbs, adjectives, etc. These parts of speech will be changed with appropriate terms from a vocabulary that has been organized similarly. The passphrases created in this manner are somewhat simpler to recall (for example, "when does a bellboy spike an elect but not a sidebar").

Although TemplateDice improved upon Diceware, it comes with a compromise in security. According to researcher [21], TemplateDice is not scalable as the information required by the system are all internally encoded within the algorithm. Again, and more importantly, the security of the passphrases does not scale well with length.

[21] noted that the guess ranks of these passphrases gets saturated around length 8-guess rank of 8-word passphrase is nearly the same as that as of 13-word. The potential reason is the constraint imposed by the underlying hardcoded and extremely limited syntax rule patterns of TemplateDice.

[21] proposed a scheme they called MASCARA, attempts to provide a balanced trade-off between security and memorability. They identify 72,999 user-chosen English passphrases from previous password leaks using an algorithm leverages word segmentation in the noisy text to identify these passphrases. The syntactic structures of these passphrases are distinctly different from Diceware-favoring memorability over guessability.

Additionally, they develop measurement frameworks for passphrase memorability and guessability. They identified distinct and significant characteristics of a sentence which affect the memorability of a passphrase, building on earlier research on the memorability of natural language phrases. To gauge how easily passphrases can be guessed, the team also generated a Monte-Carlo estimate of the passphrase guess ranks. During the passphrase generating process, they employed this framework to strike a balance between rememberability and guessability.

Passphrases generated by MASCARA are more secure than those made by users and do not have any of the weaknesses of the current techniques. One of the MASCARA scheme's main drawback is that authors only took English passphrases into account.

### 2.3 Mnemonic-Guided Passphrase

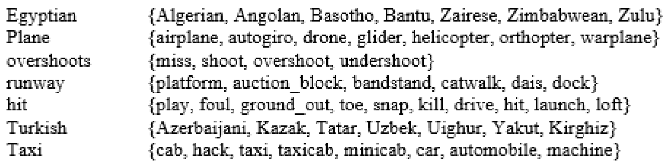
The previous paragraphs discussed different types of passphrase generation methods. One particular type is called mnemonic password. A mnemonic password only uses the initial letter of each word, as opposed to allowing the passphrase to be made up of full words strung together. As a result, a 10-word passphrase has similarities to a 10-character mnemonic password [18].

The online dictionary Merriam-Webster (2019) defines the term mnemonic as "assisting or intended to assist memory". This explanation provides a first hint regarding the expectable memorability of this particular password class and holds out the prospect of advantageous memorability properties.

Mnemonics enhance recall of information by linking it with another representation, such as an abbreviation, a rhyme, or a pattern.

A mnemonic phrase-based password is where users select a memorable phrase and then use one character from each word, commonly the first, to construct a password [15]. When the mnemonic approach is utilized, only the first letters of an underlying sentence's words are used to compile a password [1]. This concept was possibly first proposed by [2] , who suggested a number of mnemonic mappings. This tip considerably lowered guessability without reducing recall, according to research by [34].

In order to help users remember their passwords, [12] developed a system that would generate a fake news headline as a mnemonic phrase. The system was evaluated using lowercase passwords that were produced at random, and it was able to generate headlines for 80.5% of six-character passwords and 62.7% of seven-character passwords, respectively. The system's usability and user acceptance were not assessed [23].



| | |
|---|---|
| Egyptian | {Algerian, Angolan, Basotho, Bantu, Zairese, Zimbabwean, Zulu} |
| Plane | {airplane, autogiro, drone, glider, helicopter, orthopter, warplane} |
| overshoots | {miss, shoot, overshoot, undershoot} |
| runway | {platform, auction_block, bandstand, catwalk, dais, dock} |
| hit | {play, foul, ground_out, toe, snap, kill, drive, hit, launch, loft} |
| Turkish | {Azerbaijani, Kazak, Tatar, Uzbek, Uighur, Yakut, Kirghiz} |
| Taxi | {cab, hack, taxi, taxicab, minicab, car, automobile, machine} |

**Figure 4**: Examples of semantic relatives of the words in the sentence

[32] used passphrase abbreviations as mnemonic devices that were construct from the initial letters of the passphrase words.

[29] proposed using multi-item passes instead of passwords, offering examples such as "11th July 2018?," "Nanjing," and "San Antonio." They do not specify how many words can be used or whether other character sets are allowed.

[15] conducted a dictionary attack on mnemonic-phrase passwords and discovered that many users chose well-known quotes, song lyrics, etc. as their passwords, despite the fact that the security of these passwords was significantly greater than that of text passwords. Attackers who gather popular phrases and convert them to mnemonic passwords for use in dictionary attack could take advantage of this. [15].

Personal tweaks to letter case and the substitution of symbols for letters can further make the passphrase complex [15]. Mnemonics are an alternative to passwords that can increase security and memorability if users are trained to stay away from such widely used phrases [11].

Mnemonics could be used in two way; they can be used as hints to improve recall of passphrase [26] . A user chooses a passphrase, and the system creates a hint-mnemonic and stores it with the passphrase. For example, a user may choose “Mom loves apples and oranges” and the resulting hint-mnemonic becomes “MLAAO”. At authentication, the system asks the user for the passphrase, and displays the hint-mnemonic [33].

Mnemonics can also be used during the creation of passwords. Because users tend to use common word sequences, popular phrases, and grammatical rules in passphrases [15], many passphrases can be guessed by mining these common patterns from public sources. Mnemonics can be used during creation to improve randomness of word choices in passphrases, and to reduce the reuse of passphrases across different accounts [33].

## 3. Discussion

Although Porter first proposed long passphrases [22], and while Zimmermann strongly recommended them for privacy-conscious PGP users, many in the security community do not currently encourage their use among users.

A long passphrase, such as "A day that will live in infamy," is not only more secure than an 8-character random password, such as "&3Tw9#p!," but it is also easier to remember and typed. The results revealed that passphrases are a type of password that can be used in place of passwords.

While system-generated passphrase proved to be memorable, and long passphrase can protect against technical attacks. Social engineering, for which user awareness, attitude and education is essential is still an issue. Users' behaviour when generating passphrases themselves makes them vulnerable to attacks.

These studies suggests that system-generated passphrases are more secure; nonetheless, consumers prefer passphrases with proper syntax over a random sequence of words due to memorability.

Mnemonic-based passphrase were suggested as a good alternative to generate secure and memorable passwords. Despite this, they both have flaws that can be exploited by attackers, and there is no compelling proof that one is more secure than the other.

Both usability and security can be improved in numerous ways, but often improving one will have a detrimental impact on the other. To strike a balance, the researchers believe that involving users more and educating them will go a long way. Because tight restrictions can lead to disgruntled users who try to find methods around the enforced security. Different systems serve different objectives and have varying levels of security. As a result, policymakers must evaluate both security and usability in order to strike an appropriate balance for the specific system.

The ever-increasing power of attackers' machines permits cracking on ever-higher levels. We believe multi-language and culture-sensitive passphrase should be investigated by the researchers in the future.

## References

i. Andersson, D., & Svensson, A. (2013). Authentication with Passwords and Passphrases. October.

ii. Barton, B. F., & Barton, M. S. (1984). User-friendly password methods for computer-mediated information systems. Computers and Security, 3(3), 186–195. https://doi.org/10.1016/0167-4048(84)90040-3

iii. Blanchard, N. K., Malaingre, C., & Selker, T. (2018). Improving security and usability of passphrases with guided word choice. ACM International Conference Proceeding Series, 2018-Janua, 723–732. https://doi.org/10.1145/3274694.3274734

iv. Bonneau, J. (2012a). analyzing_70M_anonymized_passwords.pdf. Section VII.

v. Bonneau, J. (2012b). Guessing human-chosen secrets. May.

vi. Carnut, M. A., & Hora, E. C. (n.d.). IMPROVING THE DICEWARE MEMORABLE PASSPHRASE GENERATION SYSTEM.

vii. Forget, A., Chiasson, S., Van Oorschot, P. C., & Biddle, R. (2008). Improving text passwords through persuasion. SOUPS 2008 - Proceedings of the 4th Symposium on Usable Privacy and Security, July, 1–12. https://doi.org/10.1145/1408664.1408666

viii. Gassend, B., Clarke, D., Van Dijk, M., & Devadas, S. (2002). Silicon physical random functions. Proceedings of the ACM Conference on Computer and Communications Security, 148–160. https://doi.org/10.1145/586131.586132

ix. Grassi, P. A., Garcia, M. E., & Fenton, J. L. (2020). Digital identity guidelines.(National Institute of Standards and Technology, Gaithersburg, MD). NIST Special Publication 800-63-3, 1–75. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-3.pdf

x. Hemamalini, M., & Saranya, R. (2019). Graphical password authentication using hybrid pin keypad. Malaya Journal of Matematik, S(1), 554–559. https://doi.org/10.26637/mjm0s01/0100

xi. Hub, M., Čapek, J., Myšková, R., & Roudný, R. (2010). Usability versus security of authentication. International Conference on Communication and Management in Technological Innovation and Academic Globalization - Proceedings, 34–38.

xii. Jeyaraman, S., & Topkara, U. (2005). Have the cake and eat it too - Infusing usability into text-password based authentication systems. Proceedings - Annual Computer Security Applications Conference, ACSAC, 2005(August), 473–482. https://doi.org/10.1109/CSAC.2005.28

xiii. Joudaki, Z., Thorpe, J., & Martin, M. V. (2018). Reinforcing system-assigned passphrases through implicit learning. Proceedings of the ACM Conference on Computer and Communications Security, 1533–1548. https://doi.org/10.1145/3243734.3243764

xiv. Keith, M., Shao, B., & Steinbart, P. J. (2007). The usability of passphrases for authentication: An empirical field study. International Journal of Human Computer Studies, 65(1), 17–28. https://doi.org/10.1016/j.ijhcs.2006.08.005

xv. Kuo, C., Romanosky, S., & Cranor, L. F. (2006). Human selection of mnemonic phrase-based passwords. ACM International Conference Proceeding Series, 149, 67–78. https://doi.org/10.1145/1143120.1143129

xvi. Lee, K. W., & Ewe, H. T. (2006). Coinware for multilingual passphrase generation and its application for Chinese language password. 2006 International Conference on Computational Intelligence and Security, ICCIAS 2006, 2, 1511–1514. https://doi.org/10.1109/ICCIAS.2006.295312

xvii. Lee, K. W., & Ewe, H. T. (2007). Passphrase with semantic noises and a proof on its higher information rate. Proceedings - CIS Workshops 2007, 2007 International Conference on Computational Intelligence and Security Workshops, 652–655. https://doi.org/10.1109/cisw.2007.4425580

xviii. Lennartsson, M. (2019). Evaluating the Memorability of Different Password Creation Strategies : A Systematic Literature Review. http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1321294&dswid=-5573

xix. Liu, Y., Li, R., Liu, X., Wang, J., Zhang, L., Tang, C., & Kang, H. (2018). An efficient method to enhance Bitcoin wallet security. Proceedings of the International Conference on Anti-Counterfeiting, Security and Identification, ASID, 2017-October, 26–29. https://doi.org/10.1109/ICASID.2017.8285737

xx. Matsuura, K. (2001). Echo back in implementation of passphrase authentication. Proceedings of the 2001 International Workshop on Cryptology and Network Security, Taipei, Taiwan, 238–245.

xxi. Mukherjee, A., Murali, K., Jha, S. K., Ganguly, N., Chatterjee, R., & Mondal, M. (2023). MASCARA: Systematically Generating Memorable And Secure Passphrases. In Proceedings of ACM Conference (Conference'17) (Vol. 1, Issue 1). Association for Computing Machinery. http://arxiv.org/abs/2303.09150

xxii. Porter, S. N. (1982). A password extension for improved human factors. Computers and Security, 1(1), 54–56. https://doi.org/10.1016/0167-4048(82)90025-6

xxiii. Ranganayakulu, S. (2012). a System-Generated Password and Mnemonic Approach To Optimize the Security and Usability of Text-.

xxiv. Reinhold, A. G. (2000). Picking a Strong Passphrase with Diceware.

xxv. Richard, S., Kelley, P. G., Komanduri, S., Mazurek, M. L., Ur, B., Vidas, T., Bauer, L., Christin, N., & Cranor, L. F. (2012). Correct horse battery staple: Exploring the usability of system-assigned passphrases Richard. Usable Privacy and Security (SOUPS).

xxvi. Robert, B., Chiasson, S., & Van Oorschot, P. C. (2012). Graphical passwords: Learning from the first twelve years. ACM Computing Surveys, 44(4), 1–41. https://doi.org/10.1145/2333112.2333114

xxvii. Shay, R. (2015). Creating Usable Policies for Stronger Passwords with MTurk School of Computer Science Carnegie Mellon University Pittsburgh , PA 15213 Thesis Committee Lorrie Faith Cranor , Advisor Lujo Bauer Brian LaMacchia , Microsoft Research Submitted in partial fulf. February.

xxviii. Shay, R., Komanduri, S., Durity, A. L., Huh, P. S., Mazurek, M. L., Segreti, S. M., Ur, B., Bauer, L., Christin, N., & Cranor, L. F. (2014). Can long passwords be secure and usable? Conference on Human Factors in Computing Systems - Proceedings, 2927–2936. https://doi.org/10.1145/2556288.2557377

xxix. Shen, J., Choo, K. K. R., & Zeng, Q. (2019). Multi-item Passphrases: A Self-adaptive Approach Against Offline Guessing Attacks. In Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST (Vol. 259). Springer International Publishing. https://doi.org/10.1007/978-3-030-05487-8_11

xxx. Subangan, S., & Senthooran, V. (2019). Secure Authentication Mechanism for Resistance to Password Attacks. 19th International Conference on Advances in ICT for Emerging Regions, ICTer 2019 - Proceedings, September. https://doi.org/10.1109/ICTer48817.2019.9023773

xxxi. Suru, H. U., Muslim, A. A., Suru, S. U., & Suru, H. U. (2019). A Review of Graphical, Hybrid and Multifactor Authentication Systems. 10(1), 1447–1475.

xxxii. Woo, S. S. (2020). How Do We Create a Fantabulous Password? The Web Conference 2020 - Proceedings of the World Wide Web Conference, WWW 2020, 1491–1501. https://doi.org/10.1145/3366423.3380222

xxxiii. Woo, S. S., & Mirkovic, J. (2016). Improving Recall and Security of Passphrases Through Use of Mnemonics. https://www.semanticscholar.org/paper/Improving-Recall-and-Security-of-Passphrases-Use-of-Woo-Mirkovic/308e48f46bdcde59f1224fe178d36bd242a542cd

xxxiv. Yan, J., Blackwell, A., & Anderson, R. (2000). passwords – some empirical results. 500.

xxxv. A. G. Reinhold. (2006, October) Diceware passphrase home page. [Online]. Available: http://www.diceware.com/

xxxvi. https://www.eff.org/dice

xxxvii. Murray Grant. 2014. Template based diceware algorithm. https://github.com/ligos/MakeMeAPassword.

## Author Profile

**B.T Shehu** received his B.Sc. and M.Sc. degrees in Computer Science from Usmanu Danfodiyo University Sokoto and Kebbi State University of Science & Technology Aleiro, respectively. He is currently working as a lecturer in the department of Computer Science, Federal University Birnin Kebbi. His area of research interests includes Human-Computer Interaction, Privacy and Information Security.

**S.U Suru** is a lecturer in the Department of Computer Science and was the Director of ICT in Kebbi State University of Science and Technology Aleiro from 2014 to early 2018. His area of research interest is Usability and Security of Graphical Authentication Systems.

**T.M Emmanuel** received his B.Sc. and M.Sc. degrees in Computer Science from Usmanu Danfodiyo University Sokoto and Kebbi State University of Science & Technology Aleiro, respectively. He is currently working as a lecturer in the department of Computer Science, Federal University Birnin Kebbi. His area of research interests includes Human-Computer Interaction, Privacy and Information Security.