

Password Entropy and Pronunciation-Based Spelling: A Study Using Katakana

Nathaniel Kofi Adjadeh¹, Masaki Narita¹

Abstract: A report from Ethnologue estimates that there are about 7,159 languages spoken globally, with 44% considered endangered. Around 3.7 billion people speak the twenty most widely used languages. Language is a fundamental aspect of culture, shaping daily life and influencing how individuals interact with technology, particularly in areas such as password creation. In this study, we compare the entropy of common English words with their Japanese Katakana translations, written in Romaji (the Romanized representation of Japanese). The goal is to investigate whether phonetic spelling, as represented in Romaji, impacts password strength, measured by entropy and to what extent it influences memorability. We selected one hundred frequently used English words, translated them into Katakana, converted them to Romaji, and calculated their entropy. The resulting passwords were then hashed using both BCRYPT and SHA-256. Finally, we ran various attack simulations to evaluate password resistance and overall security.

Keywords: Hashing, Katakana, Password Entropy, Phonetic Pronunciation, Romaji

1. Introduction

Language is a vital component of culture, shaping how communities think, communicate, and interact with the world [1]. It is reasonable to assume that language also influences the way people create passwords. Among the thousands of spoken and written languages globally, twenty are spoken by approximately 3.7 billion people [2].

Passwords are foundational to digital security and remain the most widely used method for protecting online accounts and systems. Since their inception, passwords have become mainstream, sometimes to the frustration of IT professionals owing to poor user password practices and systemic security challenges [3]. Over time, password security has weakened due to several key factors including:

- Limited user awareness of secure password creation practices.
- Increasing sophistication of attackers.
- Widespread password reuse, driven by the growth in online account usage.
- Enhanced availability of powerful password-cracking tools.

Despite their weaknesses, passwords remain dominant due to their ease of implementation and lack of reliance on specialized infrastructure. However, massive data breaches have exposed billions of passwords. For instance, Jordan Hart of Business Insider recently reported a leak of approximately 16 billion credentials,

allegedly containing login data for services such as Apple, Gmail, Telegram, Facebook, GitHub, and more [4].

To improve minimum account security, many organizations and websites implement passwords policies, the most popular among which is the minimum eight characters, including both lowercase and uppercase, at least one special character or symbol and number policy. Currently, there is no data that supports whether these policies improve account security [5]. In addition, over the last few years, there has been a push by experts to prioritize password length over complexity [6].

Passwords are typically made using Latin characters (alphabets a-z, A-Z), numbers and special characters. That begs the question, how do people whose native languages do not use Latin characters create appropriate passwords for the numerous internet services they subscribe to?

One of such languages is Japanese. Japanese is made up of three main character types, Hiragana, Katakana and Kanji. Katakana is used to refer to loanwords, words from other languages except Chinese, adopted and used in Japanese [7], this is our focus for this study. Romanized Japanese words or Romaji is a technique of writing Japanese using Latin alphabet.

In this study, we explore how pronunciation-based spelling, specifically using Romaji representations of English words affects

¹ Iwate Prefectural University, Graduate School of Software and Information Science

password entropy. Our goal is to determine whether this approach improves overall password strength and usability.

2. Related Papers

Password study is not a new topic, and neither is the study of phonetic based password.

[8] [1] talk about how diverse groups of people create their passwords based on their linguistic and cultural background. [1] Focuses on general syntax of passwords, investigating how some groups of people use names, numbers, mangling or repeating words in the creation of their passwords. [8] On the other hand, addresses the issue of people incorporating personal information into their passwords.

[3] investigates converting a dataset of user generated passwords into phenomes and calculate the pronounceability of the phenome-based representations. They proposed a pronounceability scoring scheme that can be used to estimate how usability of password from the memorability point of view.

There is a plethora of attacks that target user passwords of which the most prominent are brute-force and dictionary attacks. Brute force is an attack that uses the process of trial and error; the attacker attempts several different passwords until they get the correct password [9]. A dictionary attack is defined as an attack where an attacker tries to guess a value X and has a way to verify it. The verification can be in many forms but most commonly is when an attacker uses X to reproduce Y , which is referred to as the verifier [10]. Simply put, a dictionary attack involves an attacker using a list of passwords and attempting to match the hash he is trying to crack with the hash of the passwords he already has.

3. Motivation

Passwords are the primary security mechanism for numerous services we use every day, however, creating good passwords and implementing proper password management techniques has grown exponentially difficult due to the sheer number of internet accounts the average user has. A report from NordPass in 2024, stated that the average internet user is estimated to have one hundred internet accounts [11].

Microsoft's recommended password guidelines in [12] and ensuring that passwords do not contain personal information decrease the likelihood of an attacker guessing passwords. The passwords are required to also be unique across all your accounts, ensuring that compromise of one account does not affect all other accounts.

These requirements are unrealistic, and as a result most users typically reuse passwords, write down passwords in a notebook or save them in unsecured computer files. Other people also opt for

weak, easily guessable passwords simply because they are easier to remember but ultimately compromising security in favour of convenience.

4. Methodology

Password entropy is defined as the measurement of the strength of a password based on how effective a guessing or brute force attack would be on the password [13]. Password entropy is calculated with the formula:

$$E = \log_2 (R^L)$$

E , represents the password entropy, measured in bits. R stands for the range of characters (A-Z, a-z, special characters(symbols) and 0-9). L stands for the number of characters in a password i.e. the length of the password. In this project we select 100 popularly used Katakana words and their English translations. We converted the Katakana words to English using their phonetic characteristics. After which we ran various test to calculate and compare the entropy values. For this study, we refer to Katakana words translated to English phonetically as *Romaji* and refer to the English translation as *English*. In this study we hash the generated passwords using BCRYPT and SHA-256 and attempt to crack the hashes using hashcat.

BCRYPT is a hashing algorithm that was designed by Niels Provos and David Mazières based on the blowfish cipher to be a key derivation function used to hash password. BCRYPT is an adaptive function which incorporates a salt that protects against rainbow table attacks. BCRYPT is designed to be slow, making it resistant to brute-force attacks. A BCRYPT hash can be identified with the prefix "\$2a\$" and "2y" [14].

SHA-256 stands for Secure Hash Algorithm, and the 256 represents the digest. SHA-256 is not an encryption algorithm but is used in encryption protocols. SHA-256 is the solution to the SHA-1 hash which was deemed unsecure but is 2.2 times slower [15].

Currently, BCRYPT is among the most used password hashing algorithms used, however, due to its slow design, it can be resource intensive and require an exceptionally long time to crack, even with an 8-character hashes. A report by Marcus White, stated that an 8-character password made up only lowercase characters would take 4 days to brute-force using an Nvidia RTX 4090 graphics card [16]. Considering this, we substitute BCRYPT with SHA-256 in some of our tests to be able to assess our hypothesis.

While the results in this substitution do not perfectly reflect real-world outcomes using BCRYPT, they provide a useful approximation of what can be expected under actual BCRYPT conditions.

Finally, we used the tool Hashcat in our cracking tests. Hashcat is a self-proclaimed “world’s fastest password cracker.” Hashcat is distributed under the MIT license and has won five of the last 7 years of the “Crack me if you can” contest between 2012 and 2019. Hashcat is a password cracking tool that is fast, efficient, and versatile when conducting brute-force. Hashcat supports most operating systems and works with most hardware configurations [17].

For this experiment, we used a computer running Windows 11, equipped with an intel core i7-14700KF (14th Gen) Processor, 32GB of RAM and an ASUS GeForce RTX 5060 Ti GPU.

5. Passwords Generated

In this study, we selected 100 simple, commonly used English words as the foundation of the password creation dataset. These words were translated into Japanese using their Katakana equivalents and subsequently converted into Romaji based on their phonetic characteristics. This conversion was automated using a custom python script. Results can be seen in Table 1.

Next, we generated passwords using a random concatenation of these words, forming a total of 166,650 two-word and three-word passwords for both English and Romaji, a sample of the results can be seen in Table 2.

Table 1. English to Romaji conversion sample

Romaji	English
Amerika	America
Orinpikku	Olympics
Kukkii	Cookie
Sutoresu	Stress
Rajio	Radio

Table 2. English and Romaji generated passwords

Romaji	English
amerikakukkii	americacookie
orinpikkurajio	olympicsradio
orinpikkusutoresu	olympicsstress
ofisujaketto	officejacket
enjinchokoreeto	enginechocolate

6. Entropy Calculation

We calculated the entropy of each Romaji word alongside its English equivalent and compared the values. Out of the 100-word pairs, the entropy values were **equal in 23 cases, English words had higher entropy in another 23, while Romaji words had higher entropy in fifty-four cases**. These results support our initial hypothesis and provide the basis for further testing. The words used

were simple, everyday terms, examples of which are shown in **Figures 1 and 2**.

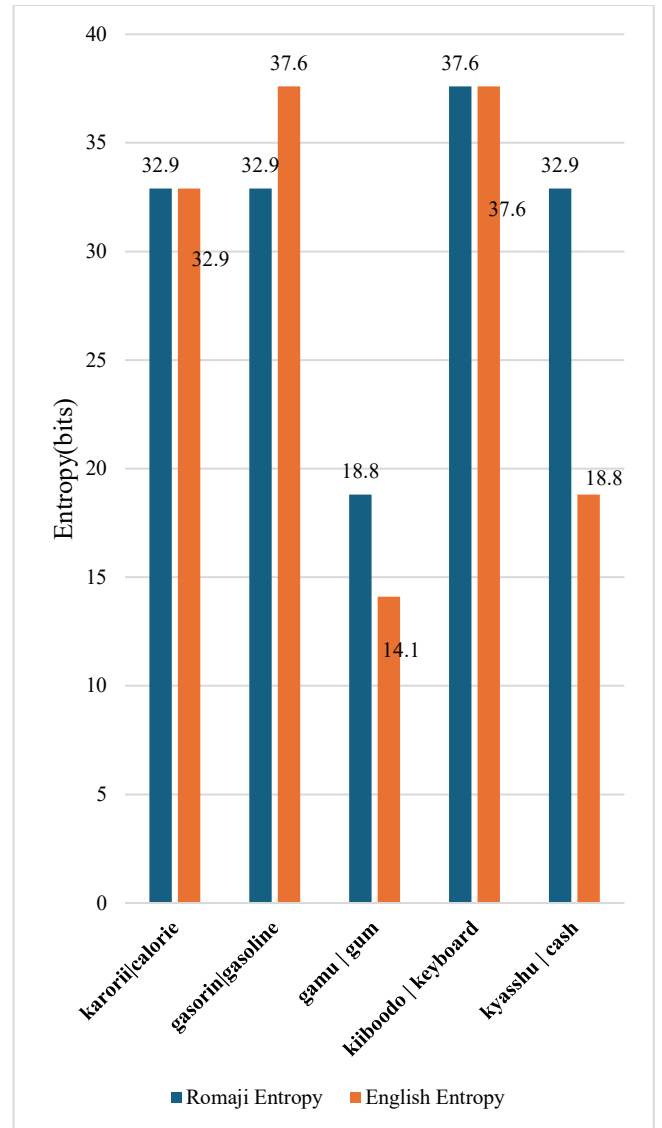


Figure 1: Romaji and English word entropy comparison

Next, we randomly generated combinations of 2 to 3 words, ensuring that Romaji words were combined only with other Romaji words, and English words only with other English words. Entropy was recalculated for these combinations and the results are shown in Figure 2.

Among five randomly selected samples from both the single-word and combined-word sets, the Romaji combinations showed higher entropy more frequently, twice for the individual words and three times for the word combinations.

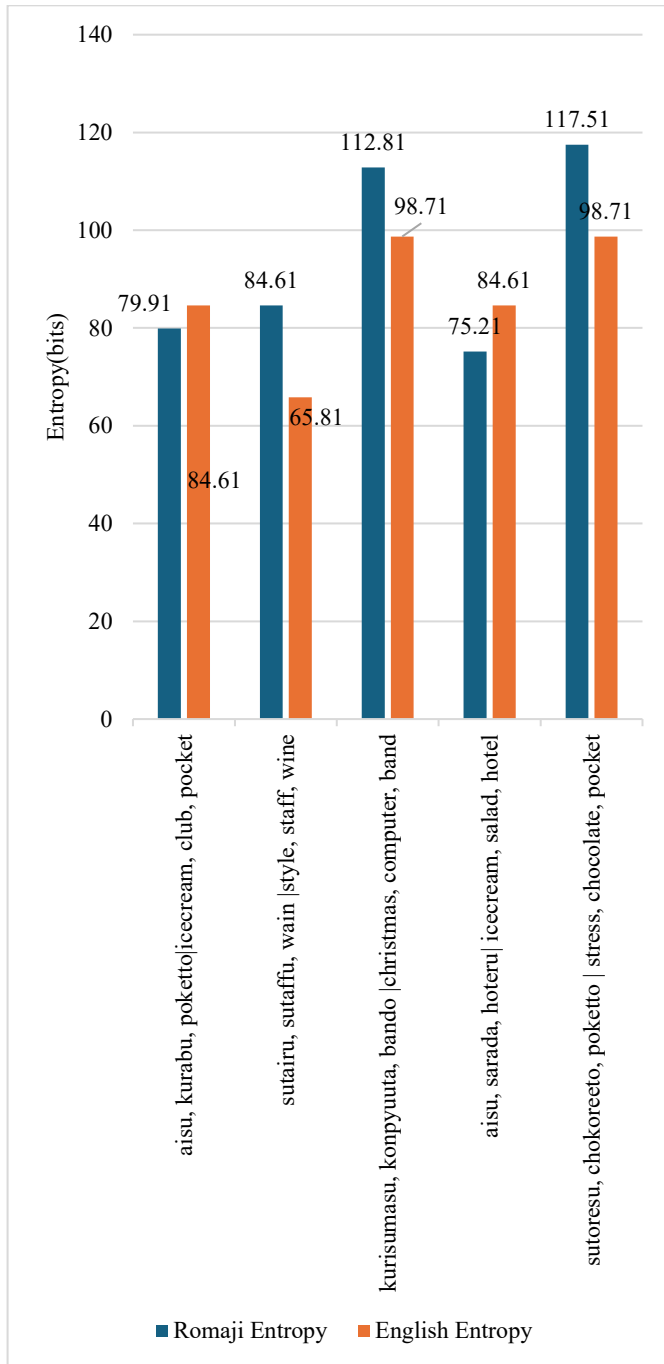


Figure 2: Romaji and English word entropy comparison

The analyses and word count of the randomly generated passwords are seen below in Figure 3.

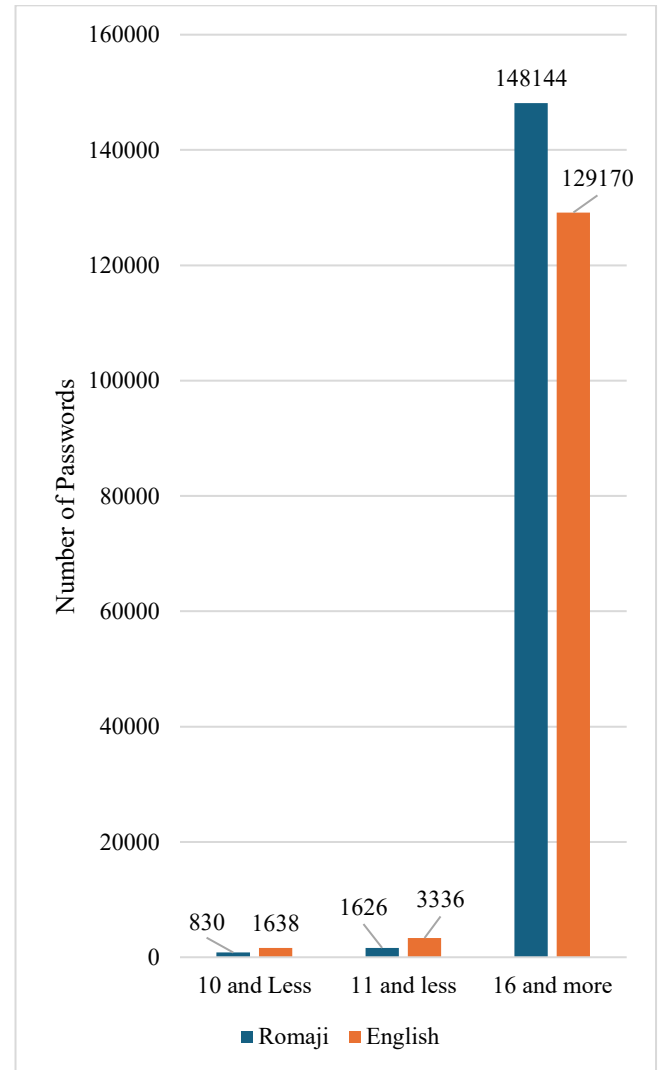


Figure 3: Character count of generated passwords

7. Attack Simulation

7.1. Shuffled Wordlist Test

We hashed both the Romaji and English password sets using the BCrypt algorithm and conducted cracking attempts using Hashcat. Two scenarios were evaluated: first, using the original password list as the wordlist; and second, using the same list but with the order randomized. The objective was to determine whether the position of passwords within a wordlist affects cracking efficiency. To ensure consistency, both lists were shuffled using the same random seed, simulating comparable conditions. The initial simulation estimated a completion time of 1 year and 259 days. Given the impracticality of running the tests for this duration, each test was constrained to a 48-hour window, after which the results were recorded for analysis in **Table 3 and 4**.

Table 3. Original wordlist crack

Original wordlist	Romaji	English
Words cracked	544	555

Table 4. Shuffled wordlist crack

Shuffled wordlist	Romaji	English
Words cracked	546	521

In the shuffled wordlist attack, 521 of the English password hashes were successfully cracked, compared to 546 for the Romaji set. This difference between the Romaji and English cracked hashes represents approximately 0.015% of the total 166,650 entries, an insignificant margin within the context of this study. Nevertheless, we deemed it necessary to acknowledge this discrepancy, given that the specific cause of the variation remains unclear.

In the original wordlist dictionary attack, 544 hashes were cracked using the Romaji list, while 555 were cracked using the English list. The increase from 521 cracked hashes in the shuffled list to 555 of English hashes represents a 6.52% improvement relative to the previous English result, and a 0.0204% increase when measured against the total dataset of 166,650 entries.

Again, in relation to the Romaji words cracked, there was a slight decrease of 0.3676% (from 546 cracked in the shuffled Romaji list to 544 in the original list). We believe that this is an insignificant difference, given that the increase from 544 to 546 accounts for a 0.0012% increase relative to the total of 166,650 cracked hashes. Figure 4 gives a clearer representation of the test results and make it easier to analyse the results.

From this experiment, we can conclude that, although the order of the wordlist relative to that of the hashfile might have some influence on the number of cracked passwords, the results show that the difference is statistically negligible.

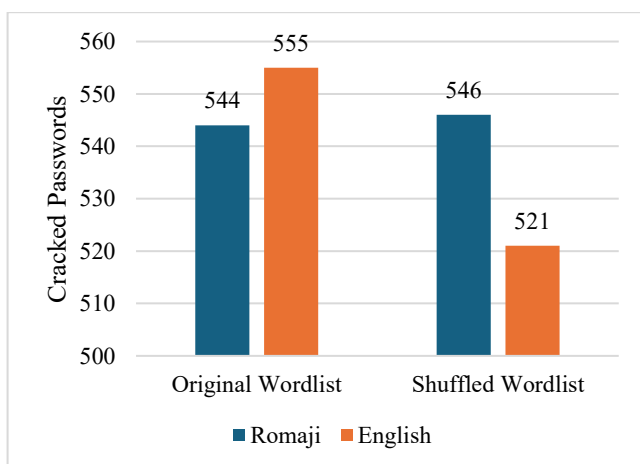


Figure 4: Original Wordlist versus Shuffled Wordlist

```
hashcat -m 3200 -a 1 -session bkat "path\to\hashfile"
"path\to\wordlist" "path\to\wordlist"
```

Above is a sample of the hashcat command used in the test

7.2. Password Dictionary Crack

Due to the slow nature of BCRPYT, making it resource intensive and time consuming to crack. To test our hypothesis, we hashed the passwords using **SHA-256** hashing algorithm.

We hashed both Romaji and English lists using SHA-256 and run a crack against popular password dictionaries namely **RockYou2024** and **Weakpass**. We recorded the crack duration and the number of passwords that were cracked in all instances. The results are in **Table 5** and **6**.

These results show two things:

1. Analysis of large publicly available password databases reveals that English-based passwords appear more frequently than their Romaji counterparts, although the difference is small.
2. Even when using a dataset of randomly generated words and their combinations, over 1,500 entries were found in the Weakpass database, and more than 2,000 appeared in the RockYou2024 list, highlighting that many random combinations may still overlap with known breached passwords.

Below is a sample of the syntax of the hashcat command used in the experiments.

```
hashcat -m 1400 -a 0 -session combkat "path\to\hashfile"
"path\to\password\dictionary"
```

Table 5. Weakpass crack results.

Weakpass	Romaji	English
Time (seconds)	812	841
Cracked	32	2662

Table 6. RockYou2024 crack results.

RockYou2024	Romaji	English
Time(seconds)	1166	1174
Cracked	46	3221

7.3. 10 and 11-character crack

The random password generation script produced the following results in Table 7.

Table 7. 10 and 11-character Password Count

	10-character or less	11-character or less
Romaji	830	1626
English	1638	3336

Given that all passwords consist solely of lowercase letters with no special characters or numbers, we performed a mask attack in Hashcat, configuring the maximum password lengths first to 10 and then 11 characters. The results were as we suspected. In both 10 and 11-character tests, hashcat was able to crack more than double the number of English passwords as Romaji.

The results of this test in **Table 8 and 9** prove to us that, at least on face value, Romaji passwords are longer thus, having a higher entropy and making them more difficult to crack than their English version.

Below is a sample of the syntax of the hashcat command used in the experiments.

```
hashcat -m 1400 -a 3-session maskattack -increment-  
increment-min=7-increment-max  
=11"path\to\hashfile"?l?l?l?l?l?l?l?l?l?l
```

Table 8. 10-character mask attack results

Ten characters	Romaji	English
Time (seconds)	33324	33616
Cracked	827	1628

Table 9. 11-character mask attack results

Eleven characters	Romaji	English
Time (seconds)	870118	864002
Cracked	1623	3326

Figure 5 provides a clear visual representation of the number of hashes that were successfully cracked in each instance. The data shows that in both instances; a greater number of English-based hashes were cracked compared to the Romaji-based hashes.

The results indicate a high success rate in all cases, with 99.64% in the Romaji passwords and 99.39% for the English passwords in the case of 10-character password set. The crack rates were even higher in the 11-character password set with 99.82% in the case of Romaji and 99.70% for English passwords.

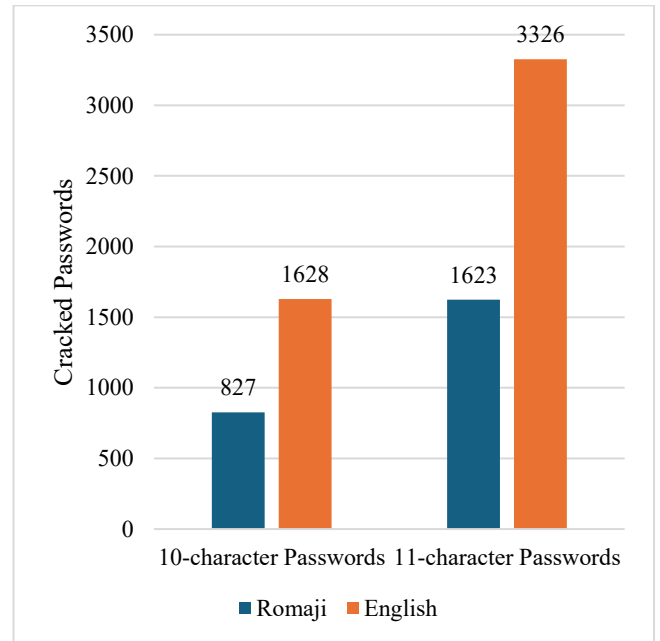


Figure 5: 10 and 11-character passwords cracked

7.4. Combinator Attack Crack

In this test, we performed a combinator attack in Hashcat using the same wordlist twice. This approach targets passwords formed by concatenating two words from the list. The objective was to measure how many passwords Hashcat could crack when provided with the constituent wordlist. For this test, we reverted to the BCrypt hashing algorithm but limited the amount of time to 48 hours.

Since the combinator attack feature native to Hashcat can only combine two words, we conducted a separate procedure to evaluate three-word combinations, consistent with our earlier statement that passwords were generated by combining two to three words. To generate these three-word passwords, we developed a Python script that concatenated each word with every other word to form all possible three-word combinations. We ensured that word order did not affect uniqueness and prevented any single word from being repeated three times within the same password.

From our one hundred words from both groups, we were able to generate 999,900 passwords formed by concatenating three words. We ensured that no exact combination was repeated in the same order.

During our initial random password's generation process, our custom script generated 4,650 unique passwords by concatenating two-words and 161,700 unique passwords by concatenating three-words. This process was not predefined and was entirely random, we were only able to do these analyses after the passwords were generated.

Table 10. Combinator & three-word attack results

	Number of Cracked Romaji Passwords	Number of Cracked English Passwords
Two-word Passwords	30	37
Three-word Passwords	456	473

hashcat -m 3200 -a 1 -session combinator "hashfile" "wordlist" "wordlist"

Above is a sample of the hashcat command used in the experiment.

Given a fixed cracking duration of 48 hours and a total of 10,000 and 999,900 possible combinations for the two-word and three-word password sets respectively, we can estimate the cracking rate based on the results of the test in Table 10.

For the Romaji-based passwords, we can estimate the average cracking rate of two-word passwords set was, 1 password every 96 minutes and 1 password every 6.32 minutes for the three-word passwords set. Within the stipulated 48 hours cracking duration, we were able to successfully recover approximately 0.645% of the total of 4,650 two-word password combinations and 0.282% of the total 161,700 three-word password combinations.

In the test for English passwords, the cracking rate for two-word passwords was 1 password every 78 minutes and three-word passwords was 1 password, every 6.08 minutes.

Based on this analysis, although the number of cracked two-word hashes was lower, the percentage of cracked hashes was more than twice that of the three-word hashes. This indicates as expected, a higher efficiency in the 2-word cracking process and further supports the premise that longer passwords are more resistant to cracking attempts.

8. Conclusion

In this study, we evaluated the strength of various password combinations derived from Katakana and English words. Our findings indicate that passwords generated using Romaji words consistently demonstrated higher strength than their English counterparts, both in brute-force scenarios and when evaluated against common password dictionaries.

We assert that effective passwords must meet two key criteria: security and memorability. This research supports the idea that randomly generated Romaji-based passwords offer enhanced security without sacrificing usability. Among the 166,650 password combinations evaluated, the average entropy of Romaji passwords was 90.75 bits, compared to 87.68 bits for English passwords, highlighting a measurable advantage in using Romaji for secure password generation.

Results from the various test we conducted including, combinator attacks, character-length-based mask attacks and dictionary attacks showed that the Romaji (Katakana) passwords consistently exhibited slower crack rates or less cracked passwords compared to the English passwords. These results prove that Romaji passwords are more resistant to cracking than English passwords under the same conditions.

This supports our hypothesis that pronunciation-based spelling, specifically using Katakana representations, enhances password strength. From a brute-force resistance perspective, these passwords demonstrate increased entropy and complexity.

When users incorporate additional elements into their passwords such as special characters, numbers, and mixed case letters the search space expands significantly, thereby increasing password strength, as previously noted by [18]. Moreover, some researchers argue that long passwords or passphrases exceeding 16 characters offer sufficient security on their own, without requiring added complexity [16] [19]. Analysis of our randomly password dataset shows that Romaji passwords tend to exhibit a higher character count, specifically 148,144 passwords exceeding 15 characters compared to 129,170 passwords in the English set. This suggests that, using these passwords in their unmodified state, i.e. without added complexity such as mixed cases, numbers and special characters, Romaji passwords inherently offer a stronger baseline security level for your account.

Simultaneously, users must also ensure that their passwords are unique across all services. Regardless of a password's theoretical strength, if it appears in a leaked credential database, the account becomes vulnerable to compromise. In such cases, it is only a matter of time before an attacker successfully gains access.

Our custom script was completely random with our only parameter requiring some passwords to be the concatenation of two-words and the others, three-words. As such, executing the random password generation script multiple times may produce distinct set of passwords with different properties.

9. Future work

The next stage of this study will involve assessing how easily Romaji-based passwords can be remembered by users, allowing us to further evaluate our goal of combining strong security with user-friendly memorability. We also plan to conduct some additional statistical tests on the results to further under them.

References

- [1] K. Mori, T. Watanabe, Y. Zhou, A. A. Haseagawa, M. Akiyama and T. Mori, "Comparative Analysis of Three Language Spheres: Are Linguistic and Cultural Differences

Reflected in Password Selection Habits?,” *IEICE TRANSACTIONS on Information and Systems*, vol. 103, no. 7, pp. 1541-1555, 2020.

- [2] Ethnologue, “How many languages are there in the world?,” Ethnologue, 2025. [Online]. Available: <https://www.ethnologue.com/insights/how-many-languages/>.
- [3] J. Hart, “A massive trove of 16 billion stolen passwords was discovered — here's what to do,” *Business Insider*, 2025. [Online]. Available: <https://www.businessinsider.com/how-to-protect-accounts-data-breach-password-leaks-2025-6>.
- [4] S. Komanduri, R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor and S. Egelman, “Of passwords and people: measuring the effect of password-composition policies,” in *Proceedings of the sigchi conference on human factors in computing systems*, 2011.
- [5] J. Yip, “Best practices for creating a corporate password policy,” *Smartdeploy*, 13 June 2025. [Online]. Available: <https://www.smartdeploy.com/blog/best-practices-for-creating-corporate-password-policy/>.
- [6] N. Hosokawa, “Katakana and Japanese National Identity: The Use of Katakana for Japanese Names and Expressions,” *Silva Iaponicarum*, vol. 56, pp. 119-136, February 2021.
- [7] M. AlSabah, G. Oligeri and R. Riley, “Your culture is in your password: An analysis of a demographically-diverse password dataset,” *Computers & security*, vol. 77, pp. 427-441, 2018.
- [8] K. S. Walia, S. Shenoy and Y. Cheng, “An empirical analysis on the usability and security of passwords,” in *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, 2020.
- [9] K. T. Dave, “Brute-force Attack “Seeking but Distressing,”” *International Journal of Innovations in Engineering and Technology (IJIET)*, vol. 2, no. 3, pp. 75-78, 2013.
- [10] S. Delaune and F. Jacquemard, “A Theory of Dictionary Attacks and its Complexity,” in *17th IEEE Computer Security Foundations Workshop (CSFW)*, Asilomar, Pacific Grove, United States., 2004.
- [11] K. Viezeyle, “Juggling security: How many passwords does the average person have in 2024?,” *NordPass*, 24 April 2024. [Online]. Available: <https://nordpass.com/blog/how-many-passwords-does-average-person-have/>.
- [12] Microsoft, “Create and use strong passwords,” Microsoft, [Online]. Available: <https://support.microsoft.com/en-us/windows/create-and-use-strong-passwords-c5cebb49-8c53-4f5e-2bc4-fe357ca048eb>.
- [13] W. Ma, J. Campbell, D. Tran and D. Kleeman, “Password entropy and password quality,” in *2010 fourth international conference on network and system security*, 2010.
- [14] P. Sriramya and R. A. Karthika, “PROVIDING PASSWORD SECURITY BY SALTED PASSWORD HASHING,” *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no. 13, pp. 5551-5556, July 2015.
- [15] Gueron, Shay, S. Johnson and J. Walker, “SHA-512/256,” in *2011 Eighth International Conference on Information Technology: New Generations*, 2011.
- [16] M. White, “[New research] How tough is bcrypt to crack? And can it keep passwords safe?,” *Specops*, 10 April 2025. [Online]. Available: <https://specopssoft.com/blog/hashing-algorithm-cracking-bcrypt-passwords/>.
- [17] R. Hranicky, L. Zabal, O. Rysavy and D. Kolar, “Distributed password cracking with BOINC and hashcat,” *Digital Investigation*, vol. 30, pp. 161-172, 2019.
- [18] M. Dell’Amico, P. Michiardi and Y. Roudier, “Password strength: An empirical analysis,” in *2010 Proceedings IEEE INFOCOM*, Sofia Antipolis, IEEE, 2010, pp. 1-9.
- [19] R. Shay, S. Komanduri, A. L. Durity, P. S. Huh, M. L. Mazurek, S. M. Segreti, B. Ur, L. Bauer, N. Christin and L. F. Cranor, “Can long passwords be secure and usable?,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2014, pp. 2927-2936.