Quantum Security Alliance (QSA)

Standard Password Practices for Organizations:
Relative Theory and Recommendations

Author:

Dr. Albert H. Carlson, QSA Chair Encryption & Entropy
ORCHID: 0000-0002-0087-6066

Technical Reviewer and Editor:

Dr. Keeper L. Sharkey, QSA Chair Quantum Applied Chemistry
ORCHID: 0000-0002-3767-6261

POC:

Dr. Merrick S. Watchorn, DMIST, QIS, QSA Program Chair
ORCHID: 0000-0003-2460-8689

(Dated: August 3, 2022)

I. EXECUTIVE SUMMARY

In June 2022, several emerging cybersecurity threats were reported, which were: 1) a phishing campaign targeted U.S. organizations in military, software, supply chain, healthcare, and pharmaceutical sectors to compromise Microsoft Office 365 and Outlook accounts and 2) the FBI, National Security Agency (NSA) and CISA announced that Chinese state-sponsored hackers targeted and breached major telecommunications companies and network service providers since at least 2020. As part of a quantum security public service announcement, the Quantum Security Alliance (QSA) was asked to explore plausible approaches to help out with individuals that were the target of these types of attack and others.

The QSA was established in December 2018, and has been working rapidly to provide context to the emerging security landscape for Quantum Computing. In the last several months of activity, the QSA has worked on numerous efforts including aiding the Cloud Security Alliance (CSA), Quantum Tech Congress, and the National Defense University (NDU) in building a working knowledge-sharing model that includes the University of Maryland, University of Phoenix, and Purdue Global Online University. "Cybersecurity methods are riddled with new technologies such as Artificial Intelligence, Machine Learning, Natural Language Processing and Operational Technologies (OT, IoT, SCADA, CPS, etc.), cloud computing, blockchain and Quantum Information Systems (QIS)" [1]. Recently, the QSA identified the need to enhance the knowledge share around password standards.

All organizations need consistent security standards in order to have practical procedures and to set logical user expectations. An important security philosophy is one that follows the axiom, "Keep the bad guys out [of your equipment, facility, and company]" [2]. Passwords are one of the most effective ways to electronically secure software, networks, and computers. Like all security measures, the measure must not be weak, otherwise it provides an attack vector that is easy for an attacker to exploit. Managing risks and vulnerabilities is often the purview of the organization. The Quantum Security Alliance (QSA) is dedicated to promoting effective security measures. In that vein, the following measures for passwords are presented as the minimum level standards for password protections by briefly describing entropy, Shannon redundancy, unicity distance, and random number generator (RNG) unicity distance. Subsequently, we list actionable items and recommendations.

II. BACKGROUND

A. Entropy, Shannon Redundancy, Unicity Distance, and RNG Unicity Distance

Entropy is a general scientific concept originally specified for use in physics and chemistry, and is a measurable physical property commonly associated with a state of disorder, randomness, or uncertainty. Although applied to different items in those fields, the math is similar. In information theory entropy is measured in bits, making base 2 natural for the measurement. In other fields the base may be different, resulting in a different look for the equation. However, the base in the equation may be converted to any other base using the base conversion equation:

$$log_a(b) = \frac{log_c(b)}{log_c(a)} \tag{1}$$

where a is the base of the desired log, b is the number of the target log, and c is the original log. At this point the standard representation for quantum information is still being decided upon. When accepted, the equation will be adjusted to reflect that base.

Entropy related to information is the amount of "surprise" and was first defined by Hartley [3] in the early 1920's where Shannon adapted it to his new study of Information Theory [4] when working on cryptography [5]. Using the principles of Abstract Algebra [6] and Topology [7], which says that if two problems share the same mathematics, then what works for one problem will work for the other; the two problems that we focus on herein are cryptography and password (p) control. In both cases, it is important to maximize entropy in the data to resist an attacker reconstructing the information. Therefore entropy becomes important for password usage and security protocols. At the core of the theory, entropy is a probabilistic function that is defined as:

$$H(x) = -\sum_{i=1}^{k} pr(x_i) ln(pr(x_i)) \tag{2}$$

where k is the number of symbols and $pr(x_i)$ is the probability of the selected symbol x_i appearing at that location in the block; i.e. the location in a password. Maximizing the entropy associated with a password, is required for creating strong security and is denoted as $H_{max}(x)$. For a unique language (L), the Shannon redundancy (R_L) relates the

entropy of repeated symbols used in a language to how many possible symbols in the language, and is given as:

$$R_L = 1 - \frac{H(x)}{H_{max}(x)} \tag{3}$$

which leads to the unicity distance (n), or amount of information required to unambiguously recover the original data. Shannon calculated this to be

$$\mathrm{n} = \frac{log\Big(\Big|K\Big|\Big)}{R_L log\Big(\Big|A\Big|\Big)} \tag{4}$$

where K is the key space, ($|K|$) is the size of the key space, and $|A|$ is the size of the alphabet for a particular language L. The goal of determining the unicity distance is to be able to assess how much data is needed to amass enough information to be able to know the underlying plain text data for an encryption. Nevertheless, passwords are not typically encrypted data. However, RNGs have a measure that is similar to in purpose to the unicity distance. This number is known as ρ and tells how much information is required to pattern the RNG. This data can then be used to assess the loss of entropy over its use and help set the time that a selection routine or password is safe. The time is known as the time to live (TTL) for the function.

B. Sharding

Breaking a message into smaller submessages involves a process known as "sharding." It was so named because of the comparison to throwing a rock through a pane of glass on which a message is written. The result is irregular size and shaped fragments with part of the message on each piece of broken glass. Shards are used to control the information available in the shard and keep it below the unicity distance for the data. The size of a shard is determined by the TTL associated with the information accumulation and content of the shard.

C. Polymorphic Encryption

Recent advance in cryptography has resulted in a concept known as "polymorphic" encryption. This type of encryption is best characterized by its most extreme example; the "One Time Pad" (OTP) [8, 9]. In the OTP example, each character is encrypted by a randomly selected cipher/key pair. Carlson [10] showed that selecting a new cipher/key can be done for a shard rather than a character while retaining the same security. Further, using the concepts of abstract algebra [6], it can be shown that the mathematics of passwords are equivalent to those of encryption. Random numbers are necessary to keep the selection process safe from attack [11, 12], requiring a regular injection of entropy in order to maximize the ρ and unicity distance of the password. Sharding with the effective use of TTLs provide the frequent, and irregular injection of entropy into the password and password selection process.

III. BRUTE FORCE ATTACK ON PASSWORDS

Assume the time needed to verify whether or not a password works is given by t_v. Also, assume that the possible number of characters that can be used in an alphabet is $|A|$. Characters in the alphabet can be composed of $[A - z]$, [0 - 9], spaces, and special characters found on the keyboard. This means that there are from 90 - 102 symbols that can be used in the password. Extending this to multi-character passwords, the a password of B characters where B is the number of bits to encode the characters in the language, the number of possible passwords ($|K|$) are given by

$$\Big|K\Big| = \Big|A\Big|^B \tag{5}$$

[13] Unlike the combinations that are found in language [10], all possibilities are possible. For each of these password sizes from the lower bound of the key size (lb) to the upper bound (ub) the number of combinations is given by

$$\left|K_{total}\right| = \sum_{i=lb}^{ub} \left|A\right|^{B} \tag{6}$$

Using Brute Force Attack [14] to break the password requires attempting to decrypt half of the possible passwords [15]. Now, adding the time it takes to evaluate a password, the time to evaluate the information indicates that, on the average, a password will be discovered in about

$$t_{avg} = \frac{1}{2} \sum_{i=lb}^{ub} \left|A\right|^{B} \tag{7}$$

While this is average, there is a 50% chance that the time will be shorter. So, pick some time fraction that the security analyst wants to remain below (p), and calculate the time that this fraction represents. Then the time to break (t_b) becomes:

$$t_b = \frac{p}{2} \sum_{i=lb}^{ub} \left|A\right|^{B} \tag{8}$$

The value of t_b gives the maximum number of possible passwords, and should represent the highest available security for passwords. However, some passwords are known to be weak - that is, they are easily guessed or cracked by password crackers, such as John the Ripper [16]. Some cracking tools also include "rainbow tables" which consist of commonly used passwords that can be rapidly searched. While it is desirable to maximize the search space the drawback of possibly using weak keys, which are normally checked first by attackers. Unfortunately, there is no way to know how many keys are weak and should be removed from the key corpus. To date there is no conclusive list of weak keys for even small password sizes. Still, the addition of more possible passwords can increase the key corpus even when weak keys are removed. If this set of weak (w) passwords is represented by $\{p_w\}$ then the total number of strong (s) passwords ($|p_s|$) is given by:

$$\left|p_s\right| = \left|K_{total}\right| - \left|\{p_w\}\right| \leq \left|K_{total}\right| \tag{9}$$

Many of these weak passwords consist of patterns, especially those which comprise words, phrases, sequences, or patterns. As the password size increases, the numbers of patterns and sequences falls off quickly. Carlson, et al, show that by the time the password is 6 characters (48 bits) the number of meaningful sequences is less than .003% of the total combinations [10, 17]. This indicates that removing sequences and words from the password corpus does not majorly affect the corpus size and avoids rapid password cracking.

The same argument also applies to having a large password. In general, the larger the password the better the security because of the multiplicative explosion that occurs as the password size increases. Even with the sequences and patterns removed the number of possible passwords will monotonically increase.

$$\left|\{p_{s_i}\}\right| \geq \left|\{p_{s_{(i+1)}}\}\right| \tag{10}$$

Randomly selected passwords are hard for humans to create and remember. Therefore, the selection and administration of passwords should be automated to both speed password selection and to enforce the rules for password selection. Rules should be library based and enforced. A history of at least the last 10 passwords needs to be kept in order to allow the system to compare the next selected password to those recently used. There are many ways to compare the new password with those previously used. If a new password happens to be too similar to older passwords an attacker can gain an advantage in breaking the new password. The algorithm for comparing passwords a and b consists of a "distance metric" ($d\langle a, b\rangle$) for deciding that similarity. Since there are many different types of similarity, it is likely that multiple distance metrics should be applied. The relationship between the two passwords is the minimum of the applied distance metrics. Approval of the new password assumes that the minimum is at least

five (5) degrees of separation between the passwords. It would be best to have a library of distance metrics available for calculations.

Along with the size and closeness metrics, the time to live for the password is important. Verification of the TTL should be done in parallel with the use of the password. Normally, this is done by using password cracking software in an attempt to ensure that an attacker is not using the same software to break into the protected system. By running that testing the protected organization assures itself the fastest notification and warning that a particular password may be compromised. If potentially broken via this testing, the password should be immediately replaced with a new password, following the established procedure for regular password updates.

The quality of the password does depend on all of the rules for passwords being followed by the user. Deviations from consistent use of the rules often comes from laziness, desire to simplify the process, or lack of knowledge. However, some of the rules are constraints on combinations of letters. For example, enforcing the rules about including upper case letters, numbers, and special characters while eliminating character sequences eliminate those passwords from the corpus. These constraints on the passwords have a small effect on the corpus size and the gains outweigh the loss of weak passwords.

IV. RECOMMENDATIONS

As a standard for the Quantum Security Alliance, the following recommendations for password use should be followed:

- Allow variations in password size - Passwords should have a range of lengths (sizes) from a minimum to a maximum size. Typically, this condition is given as "the password must consist of at least c number of characters."
- Use a large password size with a minimum of 32 characters and using as standard keyboard set of characters (102) - A password size of 32 characters gives a password space of $102^{32} \approx 1.88 \times 10^{64}$ possible passwords. At this size it would require a computer to attempt a minimum of 1657 password checks and the number of verification per second to break in that time period. This is highly unlikely, given the need to communicate to the server and process that data before a reply is given. If a larger password is used, that password space will rise accordingly.
- Change passwords at a maximum of every 30 days - A longer period for change quickly erodes security by allowing the attacker more time to work on the password and potentially break it.
- Vary the time between password changes (avoid a schedule with constant time schedule for changing passwords) - Do not change the password on a regular schedule. If possible, change the password on a randomly selected schedule with a minimum number of 5 days that a password remains constant. The minimum time for using the same password is dictated by convenience for the organization.
- Require the presence of at least one character from every set of characters for humans, remove this for computers that can be forced to do real RNG selection - This forces an attacker to try more obscure passwords and makes them harder to guess.
- A dictionary should be used to disallow passwords composed of words, dates, and phrases along with sequential numbers - Patterns are commonly used in human selected passwords. Such patterns are found in rainbow tables and are normally tried early in the cracking process. That makes passwords with patterns much more likely to be broken and should be avoided.
- Passwords should be generated by the machine, independent of human influence - Human biases are avoided if the machine is used to select the new password.
- A password history should be employed so that a password cannot be repeated for at least ten (10) passwords in a row - Password history requirements prevent the same password, or closely related passwords, are avoided. When passwords that are previously used the attacker can make use of this fact to begin when the attack left off for the last use of that password.

- A password checker that evaluates the distance between the selected password and those in the history file. The distance metric [18] used is critical and should be of at least the quality of Euclidean distance [15] - Passwords that are close variations of other recently used passwords allow the attacker to continue prior attacks and greatly cut down on the time required for cracking the password. By measuring the relatedness of passwords, those that would allow an attacker that had cracked a part of password to leverage that prior work to decrease the time to break the password.
- Employ the best in class RNGs, specifically polyRNGs - Password composition is highly dependent on the underlying RNG. The higher the quality of the RNG, the more diverse passwords will become. Newer RNGs are using polymorphic compositions are becoming available. These higher quality quality polyRNGs are the most like IRVs and are the most difficult to pattern and predict. They have the largest cycle and produce the most diverse output sequence and, consequently, passwords.
- Transmission for the sharing of new passwords should be done using sharding and polymorphic encryption - when passwords are shared over electronic media, the password should use the polymorphic principles of sharding and encryption so that the password will be kept secret for much longer than its period of use.
- A password cracker should be used on the selected password in use that includes rainbow tables. If the password is cracked, it must be immediately replaced - This is the parallel check that can determine if the password is weak and can be easily cracked. Immediately replacing the weak password is an important practice to protect the system and purge the weak password as quickly as is possible.

Maintaining these practices will limit successful attacks on a system or critical equipment.

V. CONCLUSION

We present a standard approach for password practice, which is one small aspect in security. The background mathematics and theory are presented in order to develop confidence in organizational procedures related to password security at all levels of the stack. These practices are equally applicable to classical computing and in the post quantum environment (PQE). The mathematics of passwords do not change because of the platform of which they exist. This maxim is further explored in a recently published white paper [19] which relates the theory of quantum chemistry virtualization to security protocol for additional threat detection, demonstrating this cross-platform applicability.

References

[1] M. Watchorn, J. Bishop, H. Mumm, and C. Brooks. Cybersecurity legal elasticity antecedent resilience (clear) system. *LinkedIn*, `https://www.linkedin.com/posts/quantum-security-alliance_qsas-view-on-a-clear-system-activity-6867932751069356032-KgKj?utm_source=linkedin_share&utm_medium=member_desktop_web`, 2021.

[2] Albert H. Carlson. Defense in depth: Perspectives on utility security from a military intelligence viewpoint. In *The 6th Western Power and Delivery Conference*, 2009.

[3] Paul Garrett. *The Mathematics of Coding Theory*. Pearson/Prentice Hall, Upper Saddle River, 2004.

[4] Thomas Cover and Joy Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc, New York, 2nd edition, 2005.

[5] Claude Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656 – 715, 1949.

[6] John B. Fraleigh. *A First Course in Abstract Algebra*. Addison-Wesley, 7^{th} edition, 2003.

[7] John Kelley. *General Topology*. D. Van Nostrand Company, Princeton, 1955.

[8] Uli Maurer. A universal test for random bit generators. *Journal of Cryptography*, 5(2):89–105, 1992.

[9] Albert Carlson, Bob Le Blanc, and Carlos Gonsalez. One time pad matrix, 2016.

[10] Albert Carlson. *Set Theoretic Estimation Applied to the Information Content of Ciphers and Decryption*. PhD thesis, University of Idaho, 2012.

[11] John Kelsey, Bruce Schneier, David Wagner, and Chris Hall. Cryptanalytic attacks on pseudorandom number generators. In *Fast Software Encryption, Fifth International Workshop Proceedings (March 1998)*, pages 168 – 188. Springer-Verlag.

[12] John Earl Haynes and Harvey Klehr. *Venona: Decoding Soviet Espionage in the United States (Yale Nota Bene)*. Yale University Press, 1999.

[13] Matthew Bishop. *Computer Security: Art and Science*. Addison-Wesley Professional, Boston, 2003.

[14] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley and Sons Inc., New York, 2nd edition, 1996.

[15] R. Lyman Ott and Michael T. Longnecker. *An Introduction to Statistical Methods and Data Analysis* 7_{th} *edition*. Cengage Learning, 2016.

[16] John the ripper password cracker, 10 July 2008.

[17] Albert H. Carlson, Garret Gang, Torsten Gange, Bhaskar Ghosh, and Indira K. Dutta. Evaluating true cryptographic key space size, the sixteenth international conference on systems and networks communications, icsnc 2021. 2021.

[18] Patrick Combettes. The foundations of set theoretic estimation. *Proceedings of the IEEE*, 81(2):182 – 208, 1993.

[19] A. Carlson, H. Mumms, K. L. Sharkey, and M. Watchorn. Quantum chemistry for detecting cybersecurity threats to information system. *LinkedIn*, `https://www.linkedin.com/posts/quantum-security-alliance_q-chem-for-detecting-cybersecurity-threats-activity-6951226103449489408-jPMu?utm_source=linkedin_share&utm_medium=member_desktop_web`, 2022.