Application of Number Theory in Generating One-Time Passwords

Naufal Baldemar Ardanni - 13521154[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13521154@std.stei.itb.ac.id

Abstract*—The use of one-time passwords has become increasingly important in modern authentication. One-time passwords are codes that are generated uniquely by a system and are intended to be used only once, as the name says. These codes are often used as an additional layer of security in authentication systems, such as when logging into an online account; therefore, such authentication is called multi-factor authentication. Number theory, the study of the properties of numbers, plays a critical role in generating these codes. This paper will discuss the basics of one-time passwords, the role of number theory in their generation, and the advantages and limitations of using number theory in this context. Overall, this paper aims to demonstrate the importance of number theory in ensuring the security and efficiency of one-time passwords.

***Keywords*—Hash functions, number theory, one-time passcodes, one-way functions.**

I. Introduction

One-time passwords (OTP) are unique codes that can only be used once to access accounts or systems. They are often used as an additional layer of security in addition to a username and traditional password. OTPs are crucial because they provide a high-level of security by ensuring that even if a username and password are compromised, the attacker will not be able to access the protected account or system without the OTP. This makes it much more difficult for attackers to gain unauthorized access.

For example, when you log into an online account where multi-factor authentication (MFA) is enabled, you may be required to enter an OTP that has been sent through SMS to your phone or through email as an additional security measure. This OTP is typically only valid for a short period, and once you use it, it cannot be used again. This ensures that even if someone else gets hold of your username and password, they will not be able to access your account without the OTP.

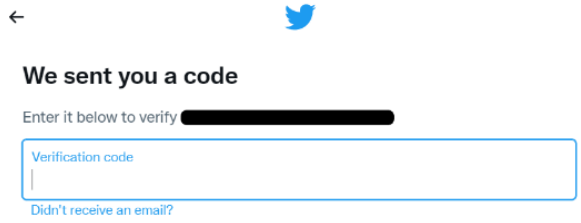

Figure 1 Twitter Asking for a Verifiction Code

Source: trustedreviews.com

OTPs are commonly used in a variety of contexts, such as online banking, online shopping, and accessing corporate systems. They are also used to protect sensitive information, such as medical records and personal identification numbers (PINs).

In recent years, the use of OTPs has become increasingly important due to the growing threat of cyber-attacks and the need to protect sensitive information. As a result, there is a growing demand for secure and efficient methods for generating OTPs.

One of the key areas where number theory is used is in the generation of OTPs. Number theory, the study of the properties of numbers and the relationships between them, plays a critical role in generating these codes. By using number theory techniques, it is possible to create OTPs that are difficult for attackers to guess or predict. This increases the security of the OTPs and makes it more difficult for attackers to gain unauthorized access to sensitive information.

For example, one common method for generating OTPs is to use modular arithmetic, a branch of number theory that deals with the properties of numbers when they are divided by a fixed integer. This allows for the creation of OTPs that have a limited lifespan and can only be used once, making them more secure than traditional passwords. Another common technique is to use prime numbers, which are numbers that are only divisible by 1 and themselves. This allows for the creation of OTPs that are difficult to factorize and therefore more secure.

II. Basic Theory

A. Modular Arithmetic

Modular arithmetic is a branch of number theory that deals with the properties of numbers when they are divided by a fixed integer. It is often referred to as “clock arithmetic” because it is used to find the remainder when a number is divided by another number, like how a clock uses a 12-hour cycle to find the time. In modular arithmetic, numbers are represented as “congruent” to each other if they have the same remainder when divided by the same integer.

For example, in the modular arithmetic system with modulus 12, the numbers 18, 30, and 42 are all congruent to 6 because they all have a remainder of 6 when divided by 12. This means that in this system, 18 is the same as 6, 30 is the same as 6, and

42 is the same as 6. This is because in modular arithmetic, the number 12 is the "equivalent" of 0, so 18, 30, and 42 all have the same remainder as 6 when divided by 12.

To perform modular arithmetic, the modulo operator (%) is usually needed, which is the symbol used in most programming languages to represent the remainder when a number is divided by another number. For example, to find the remainder when 18 is divided by 12, the following equation would be used:

$$18 \% 12 = 6 \quad (1)$$

This would give the result 6, which is the remainder when 18 is divided by 12.

B. Prime Factorization

Prime factorization is the process of finding the prime numbers that can be multiplied together to form a given number. Prime numbers are numbers that are only divisible by 1 and themselves, such as 2, 3, 5, and 7. Prime factorization is an important concept in number theory because it allows us to break a number down into its prime factors, which can be useful in many different contexts.

For example, to find the prime factorization of the number 30, we would need to determine which prime numbers can be multiplied together to give us 30. To do this, we can start by dividing 30 by the smallest prime number, which is 2. This gives us a result of 15, which is not a prime number. We can then divide 15 by the next smallest prime number, which is 3. This gives us a result of 5, which is also not a prime number. We can then divide 5 by the next smallest prime number, which is 5. This gives us a result of 1, which is not a prime number.

At this point, we have found all the prime factors of 30, which are 2, 3, and 5. To write the prime factorization of 30, we would simply list these prime factors in order, separated by multiplication signs. So, the prime factorization of 30 is 2 * 3 * 5.

Prime factorization is useful because it allows us to determine the prime factors of a number quickly and easily. This can also be useful in finding the greatest common divisor (GCD) of two numbers. In finding the GCD of two numbers, prime factorization is used to determine the largest number that can divide both numbers without leaving a remainder.

C. One-Way Function

A one-way function, also known as "trapdoor function," is a mathematical function that is easy to compute in one direction, but difficult to compute in the opposite direction without certain information. This information is often called the "trapdoor," and it allows the easy inverse computation of the function.

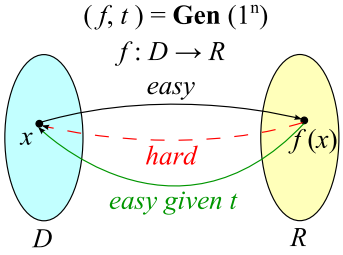

Figure 2 One-Way Function

Source: commons.wikimedia.org

In general, a one-way function is a function that satisfies two properties:

1. It is easy to compute the function given an input.
2. It is difficult to determine the input given the output of the function, unless certain information is provided (i.e., the trapdoor).

D. Hash Function

A hash function is a mathematical function that takes an input of any size and produces a fixed-size output called a "hash value" or "digest." The input to a hash function can be any type of data, such as a file, a message, or a traditional password, and the output is always a fixed-length string of characters.

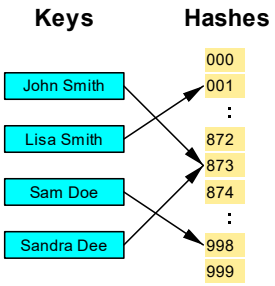

Figure 3 Hash Function

Source: commons.wikimedia.org

Hash functions are often used to create digital signatures, to verify the authenticity and integrity of a message, and to store passwords securely.

One of the key properties of a hash function is that it is a one-way function. This means that it is easy to compute the hash of an input, but it is difficult to determine the input given only the hash. This property is important for security, as it makes it difficult for an attacker to reverse the function and determine the original input.

Another important property of a hash function is that it is deterministic. This means that given the same input, the hash function will aways produce the same output.

In addition, a good hash function should be collision-resistant, which means that it should be difficult to find two different inputs that produce the same hash value. An attacker cannot create a different input that has the same hash value as the original.

III. How OTPs Are Generated

The are several different approaches to generating OTPs. One common method is based on time synchronization between the authentication server and the client providing the password. In this approach, OTPs are only valid for a short period of time, such as a few minutes. This means that even if a password is intercepted by an attacker, it will be useless once the time period has expired.

Another approach is to use a mathematical algorithm to generate a new password based on the previous password. In this case, OTPs are effectively a chain, and they must be used in a predefined order. This approach can help to prevent an attacker

from being able to predict future OTPs, as each password is dependent on the previous one.

A third approach is to use a mathematical algorithm where the new password is based on a challenge, such as a random number chosen by the authentication server or transaction details, and/or a counter. This approach can help to ensure that each OTP is unique and cannot be easily predicted by an attacker.

A. Time-synchronized

A time-synchronized OTP is a type of one-time password that is generated using a combination of the current time and a secret key. This type of OTP is often generated using a small physical device called a security token, which may resemble a calculator. The security token contains an accurate clock that has been synchronized with the clock on the authentication server.

To generate an OTP, a user enters the secret key into the security token and the current time is used as an input to a mathematical algorithm. The algorithm calculates a new and unique OTP based on the current time and the secret key. Because the OTP is only valid for a short period of time, typically a few minutes, it is important for the clock on the security token to be accurate and synchronized with the clock on the authentication server.

Time-synchronized OTPs are often used in applications where security is of high importance, such as multi-factor authentication systems. This type of OTP provides an additional layer of security, as it is difficult for an attacker to predict the OTP based on the current time and the secret key. An example of a standard for time-synchronized OTPs is the time-based one-time password (TOTP) standard.

In addition to dedicated security tokens, some applications, such as Aegis Authenticator or a password manager, can be used to generate and manage time-synchronized OTPs. These applications are usually installed on a mobile phone and use the device's clock as the source of time for generating OTPs. This allows users to access their OTPs on the go, without the need for a separate hardware token.

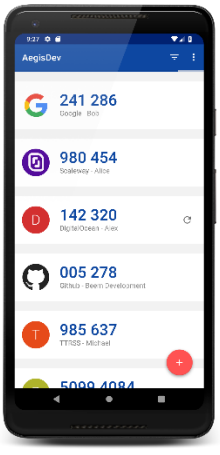

Figure 4 Aegis Authenticator

Source: getaegis.app

B. Hash Chains

This OTP system works by creating a new OTP from past OTPs that have been used. To create this new OTP, one-way function (f) is applied repeatedly to a chosen seed value (s). For example, if f is applied 1000 times to s, the result is stored on the target system. When a user logs in for the first time, a password (p) is derived by applying f 999 times to s, or $f^{999}(s)$. The target system can authenticate this password by checking that $f(p)$ is $f^{1000}(s)$, the value stored on the system. The value is then replaced by p and the user is authenticated. The process is repeated for each subsequent login, with the password being derived by applying f one fewer time than in the previous login. This can be validated by checking that when hashed, the password gives the value stored during the previous login. Hash functions are designed to be difficult to reverse, so an attacker would need to know the initial seed s in order to calculate the possible passwords, while the target system can confirm the validity of the password by checking that it gives the previously used login value when hashed. To find the next password in the series, the inverse function f^{-1} must be calculated, which is difficult to do because f was chosen to be one-way. If f is a cryptographic hash function, it is considered a computationally intractable task. An attacker who sees a one-time password will only have access for one time period or login, and the password will become useless once that period expires.

C. Challenge-Response

The use of challenge-response one-time passwords is a common authentication method that requires a user to provide a response to a challenge presented by the system. This challenge is typically generated by a token, which is a device or piece of software that generates a unique value that can be used as a one-time password. To use this method, the user must input the value generated by the token into the system, either by typing it in or by using some other method such as scanning a QR code.

To avoid duplicates, challenge-response one-time password systems often include an additional counter that increments each time a new password is generated. This ensures that even if a user receives the same challenge twice, the resulting one-time passwords will be different. However, the computation of the one-time password does not usually involve the previous password; instead, a different algorithm is used to generate each password. This makes it difficult for an attacker to guess the next password based on previous ones.

IV. Advantages And Limitations

One of the main advantages of using number theory in generating one-time passcodes is improved security. By incorporating mathematical concepts such as modular arithmetic and prime factorization into the passcode generation system, it becomes much more difficult for potential attackers to predict or crack the passcodes. This enhanced security can help prevent unauthorized access to sensitive systems and data, protecting against potential breaches and other security threats.

Another advantage of using number theory in generating one-time passcodes is unpredictability. Because the passcodes are generated using a combination of secret keys and other factors, such as the current time, it becomes almost impossible for potential attackers to predict what the next passcode will be. This makes it much harder for them to gain unauthorized access

to sensitive systems and data, even if they were able to obtain some of the passcodes through other means.

Despite these advantages, there are also some limitations to using number theory in generating one-time passcodes. One of the main limitations is the potential for computational complexity. Depending on the specific system and the mathematical operations used, generating one-time passcodes using number theory can be computationally intensive. This can be a problem in situations where the passcodes need to be generated quickly, such as when a user is trying to log into a system and needs to enter the passcode in a timely manner.

Another limitation of using number theory in generating one-time passcodes is the need for secure key management. For the passcode generation system to work effectively, the secret keys used to generate the passcodes must be kept secure and protected against potential attackers. This requires proper key management, such as securely storing the keys and regularly rotating them to prevent compromise. If the keys are not managed properly, it could potentially undermine the security of the passcode generation system.

Overall, while there are some limitations to using number theory in generating one-time passcodes, the advantages in terms of enhanced security and unpredictability make it a valuable and necessary tool for protecting against unauthorized access to sensitive systems and data.

V. Conclusion

In conclusion, the application of number theory in generating one-time passcodes is a crucial method for enhancing the security of sensitive information. Through the utilization of mathematical concepts such as modular arithmetic and prime factorization, one-time passcodes can be generated that are highly unpredictable and difficult to crack by potential attackers. This makes them an invaluable asset in the protection against unauthorized access to sensitive systems and data. Furthermore, the use of number theory in generating one-time passcodes provides a strong layer of defense for ensuring the confidentiality and integrity of sensitive information. As a result, the application of number theory in generating one-time passcodes is a valuable and necessary tool in today's digital age.

VI. Acknowledgment

I am deeply grateful to God for the guidance and support throughout my academic endeavors. My heartfelt thanks also go to my family, whose love and encouragement has been a constant source of motivation and inspiration.

I would also like to extend my sincere appreciation to Ms. Fariska Zakhralativa Ruskanda, S.T., M.T., who teaches class K2 of Discrete Mathematics (IF2120), for her exceptional teaching in the course where this assignment was given. Her knowledge provided the foundation for my work on this paper, and I am deeply grateful for the opportunity to learn from her and apply my knowledge to this assignment.

Finally, I would like to express my sincere thanks to my friends, who have been a constant source of support and encouragement throughout my studies. I am deeply grateful for their unwavering friendship and support

I am truly thankful to all these individuals for their invaluable contributions to my education and the completion of this paper.

References

[1] K. Richards and I. Wigmore, "What is a one-time password (OTP)? Definition from SearchSecurity," Techtarget, [Online]. Available: https://www.techtarget.com/searchsecurity/definition/one-time-password-OTP. [Accessed 12 December 2022].

[2] K. H. Rosen, Discrete Mathematics and Its Applications, Eight Edition, New York: McGraw-Hill Education, 2019.

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Desember 2022

Naufal Baldemar Ardanni
13521154