I. J. Mathematical Sciences and Computing, **2024, 1, 31-38**
Published Online on February 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmsc.2024.01.04

An Unorthodox Trapdoor Function

Awnon Bhowmik*
Independent Researcher
E-mail: awnonbhowmik@outlook.com
ORCID iD: https://orcid.org/0000-0001-5858-5417
*Corresponding Author

Received: 10 September, 2023; Revised: 07 October, 2023; Accepted: 16 December, 2023; Published: 08 February, 2024

Abstract: At the bedrock of cryptosystems lie trapdoor functions, serving as the fundamental building blocks that determine the security and efficacy of encryption mechanisms. These functions operate as one-way transformations, demonstrating an inherent asymmetry: they are designed to be easily computable in one direction, while proving computationally challenging, if not infeasible, in the opposite direction. This paper contributes to the evolving landscape of cryptographic research by introducing a novel trapdoor function, offering a fresh perspective on the intricate balance between computational efficiency and security in cryptographic protocols.

The primary objective of this paper is to present and scrutinize the proposed trapdoor function, delving into a comprehensive analysis that unveils both its strengths and weaknesses. By subjecting the function to rigorous examination, we aim to shed light on its robustness as well as potential vulnerabilities, contributing valuable insights to the broader cryptographic community. Understanding the intricacies of this new trapdoor function is essential for assessing its viability in practical applications, particularly in securing sensitive information in real-world scenarios.

Moreover, this paper does not shy away from addressing the pragmatic challenges associated with deploying the proposed trapdoor function at scale. A thorough discussion unfolds, highlighting the potential hurdles and limitations when attempting to integrate this function into large-scale environments. Considering the practicality and scalability of cryptographic solutions is pivotal, and our analysis strives to provide a clear understanding of the circumstances under which the proposed trapdoor function may encounter obstacles in widespread implementation.

In essence, this paper contributes to the ongoing discourse surrounding trapdoor functions by introducing a new entrant into the cryptographic arena. By meticulously exploring its attributes, strengths, and limitations, we aim to foster a deeper understanding of the intricate interplay between cryptographic theory and real-world applicability.

Index Terms: Trapdoor function, cryptography, integer factorization, RSA, ElGamal, Pollard Rho

1. Introduction

Cryptography, an evolving and dynamic field, traces its roots to ancient civilizations where it served primarily as a means of concealing messages. However, the systematic exploration of cryptology as both a science and an art commenced only about a century ago. The early cryptographic systems involved basic mathematical operations, including shifting ciphers like substitution or transposition ciphers. Over time, cryptography has transformed from a decorative practice into a crucial element of secure communication within our contemporary digital landscape [1].

Modern encryption protocols hinge on the concept of a trapdoor function, a mathematical operation that is easy to compute in one direction but computationally challenging in the reverse direction [2]. As classical cryptography advanced, researchers identified key components vital for creating robust trapdoor functions, often referred to as one-way functions. Notably, studies have underscored the significance of prime numbers in numerous cryptosystems. With dedicated efforts, various mathematical concepts have been harnessed to generate stronger and more resilient cryptosystems.

In essence, cryptography has evolved from a historical curiosity to an indispensable tool, shaping the way information is secured and transmitted in our interconnected digital era. The intricate interplay of mathematical principles and computational complexity continues to fuel advancements in cryptographic techniques, ensuring the ongoing development and adaptation of this crucial science.

RSA and ElGamal are two prominent cryptographic algorithms that have played pivotal roles in ensuring secure communication and data protection in the digital realm. RSA, named after its inventors Rivest, Shamir, and Adleman, is widely recognized for its contributions to public-key cryptography. ElGamal, named after its creator Taher ElGamal, is another influential public-key cryptosystem with applications in encryption and digital signatures.

RSA: RSA, introduced in 1977, is based on the mathematical difficulty of factoring the product of two large prime numbers. Its core idea involves the generation of a public key for encryption and a private key for decryption. The security of RSA relies on the impracticality of factoring the product of two large primes, even with powerful computing resources. It has become a cornerstone in securing online communication, digital signatures, and various cryptographic protocols [3, 4].

ElGamal: ElGamal, proposed in 1985, is another key player in public-key cryptography. It operates in the realm of discrete logarithms, where the security is based on the computational difficulty of solving discrete logarithm problems in finite fields [5, 6]. ElGamal is versatile, providing not only encryption but also digital signatures, and it offers semantic security against chosen ciphertext attacks. The combination of these features underscores the significance of exploring this system with the aim of enhancing its efficiency [7].

Inspiration from RSA and ElGamal: Our proposed cryptographic system draws inspiration from the foundational principles of both RSA and ElGamal. Like RSA, our potentially improved system incorporates the concept of modular arithmetic and exponentiation at a later section, introducing a prime factorization element. This draws from RSA's strength in the difficulty of factoring large numbers.

Additionally, our system reflects ElGamal's influence by embracing the use of discrete logarithms and modular operations. The inspiration from ElGamal contributes to the versatility of our proposed system, accommodating functionalities such as encryption and digital signatures.

While inspired by these classical cryptographic approaches, our proposed system introduces a unique mathematical structure, involving specific combinations of exponentiation, prime factors, and modular arithmetic. This innovative blend aims to provide a distinct cryptographic solution, building upon the strengths of RSA and ElGamal while presenting a novel approach to address specific cryptographic requirements. Through careful analysis and validation, we aim to contribute to the evolving landscape of cryptographic systems, ensuring security and efficiency in digital communication and data protection.

2. Integer Factorization Problem

The Integer Factorization Problem is a fundamental mathematical challenge in the field of number theory and plays a crucial role in the security of many cryptographic systems. The problem involves the decomposition of a composite integer into its prime factors [8, 9]. Given a composite number, finding its prime factors is relatively straightforward for small numbers, but as the size of the number increases, the difficulty of the factorization process grows exponentially. The Integer Factorization Problem forms the basis for various cryptographic algorithms, such as the widely used RSA algorithm, which relies on the assumption that factoring large composite numbers is computationally infeasible within a reasonable amount of time [10]. The security of these cryptographic systems hinges on the presumed difficulty of solving the Integer Factorization Problem, and its status as a hard problem contributes to the strength of many encryption schemes. Advances in quantum computing pose potential threats to existing cryptographic systems by potentially providing more efficient solutions to the Integer Factorization Problem.

3. Pollard Rho Algorithm

The Pollard Rho algorithm is a probabilistic algorithm designed to factorize composite numbers, particularly those that are large and have no special algebraic structure [11]. Developed by John Pollard in 1975, the algorithm employs a random walk strategy based on Floyd's cycle-finding algorithm, also known as the Tortoise and the Hare algorithm [12]. The key idea is to iteratively generate a sequence of values modulo the target composite number, and by identifying a non-trivial cycle in this sequence, the algorithm can reveal a non-trivial factor of the composite number. Pollard Rho is particularly effective for factoring numbers with small prime factors, making it a valuable tool in integer factorization, a field crucial for the security of various cryptographic systems. While Pollard Rho is not guaranteed to find a factor quickly in all cases, its average-case time complexity is relatively favorable, and it remains an essential component in the arsenal of algorithms used to address the Integer Factorization Problem.

4. Trapdoor Function

The foundation of any cryptosystem lies in the implementation of a specialized mathematical trapdoor function, a pivotal component that serves to safeguard confidential information from unauthorized access. The ingenious design of these trapdoor functions ensures that while they facilitate the seamless exchange of data among authorized parties who possess the secret key, they present a formidable challenge to any unauthorized entities attempting to decipher the information. The essence of a trapdoor function lies in its asymmetry—it allows for an uncomplicated calculation in one direction yet renders the reverse computation exceptionally complex and practically insurmountable within a reasonable timeframe, unless certain confidential information, known as the private key, is available.

Analogously, the concept of a trapdoor function can be likened to the modern lock and key mechanism in cryptography. Without possessing the precise key, akin to the private key in the mathematical realm, an individual is incapable of unlocking the encrypted data. In mathematical terms, if we denote the trapdoor function as f, the

calculation of y given $x\ (y = f(x))$ is a straightforward process. However, attempting to compute the inverse, determining x from $y\ (x = f^{-1}(y))$, becomes an exceedingly intricate task unless equipped with the specific key, denoted as k. This intricate dance of mathematical operations ensures that the security of the cryptosystem remains intact, allowing authorized users to wield the power of the key for decryption while thwarting the efforts of unauthorized entities without the requisite key.

5. Proposed Mathematical Function

The trapdoor function to discuss in this paper has the following form

$$f(x,p) = x^p p^x \tag{1}$$

where x is a positive integer and p, is a prime number. This function is used to encrypt, and its inverse is used to decrypt since the prime number is known. The following can be taken into consideration where p is prime and x represents an ASCII value:

1. **One Way Function:** This function exhibits characteristics of a trapdoor function, particularly due to the involvement of exponentiation with a prime p. The difficulty of factoring large composite numbers, a key property in trapdoor functions, is analogous to the challenge posed by the prime p in the exponentiation process.
2. **Key Generation:** In a cryptographic scenario, key generation would involve selecting a prime p and potentially an ASCII value x to form the public and private keys. The choice of p becomes crucial for the security of the system, as the strength of the function is directly related to the properties of the prime.
3. **Security Considerations:** The security of the function relies on the difficulty of computing x^p and p^x separately. If either operation could be efficiently reversed, the function's security would be compromised. The use of a prime p adds complexity, as factoring the result back into x and p would require finding the prime factors of the result.
4. **Cryptographic Applications:** Depending on the specific cryptographic context, this function could potentially be used in protocols such as digital signatures or encryption schemes. The properties of the function, including the involvement of a prime and the exponentiation operations, align with certain cryptographic requirements.
5. **Potential Weaknesses:** The choice of p as a prime is crucial for the security of the function. If p is small or lacks certain mathematical properties, the system may be vulnerable to attacks such as brute force or integer factorization.
6. **Efficiency Considerations:** The computational efficiency of this function depends on the efficiency of exponentiation algorithms, especially when dealing with large values of x or p. Techniques like modular exponentiation may be employed to optimize the calculations.
7. **Modifications and Enhancements:** As discussed earlier, introducing modulo operations, prime expansions, or other enhancements can potentially strengthen the function or improve its efficiency, depending on the specific cryptographic requirements.

In summary, while the given function exhibits some properties desirable for cryptographic applications, careful consideration of key generation, security, and efficiency is essential to determine its suitability in a specific cryptographic context. Additionally, ongoing evaluation and potential modifications may be necessary to address emerging cryptographic challenges.

6. Benefits and Drawbacks of the function

Let us analyze the trapdoor function $f(x,p) = x^p p^x$ where p is a large prime and x takes on the values for the printable ASCII characters, which is $[32 - 126]$.

Advantages:

1. **Security with Prime p:** Utilizing a large prime p adds a layer of security to the function, as factoring large primes is computationally difficult. This aligns with the security principles commonly used in certain cryptographic schemes.
2. **Number Theoretic Properties:** The use of a large prime p in conjunction with ASCII values may introduce number-theoretic properties that enhance the overall security of the function.
3. **Resistance to Certain Attacks:** Large prime values make certain attacks, such as those based on factorization, more computationally intensive and time-consuming, thereby enhancing the resistance of the function against such attacks.

4. **Versatility:** The function remains versatile, as it can be applied to a variety of contexts while benefiting from the added security provided by the large prime p.

Disadvantages:

1. **Complexity of Analysis:** The security of the function still requires a thorough mathematical analysis. The complexity of analyzing the function increases when dealing with ASCII values, and it is crucial to ensure that the function is resistant to various cryptographic attacks.
2. **Computational Overhead:** The use of large primes may introduce computational overhead, and the efficiency of the function needs to be carefully considered, especially in practical cryptographic applications where performance is a critical factor.
3. **Key Management:** Securely managing and transmitting the large prime p is a challenge. The use of established cryptographic protocols for key exchange or public-key cryptography may be necessary to address this issue.
4. **Limited Applicability:** While the function is versatile, it may not be suitable for all cryptographic applications. The specific algebraic and mathematical properties required for certain cryptographic schemes may not be guaranteed using large prime p and ASCII values alone.

The mathematical function used for the trapdoor appears as a product. This is a classic example of the integer factorization problem and can be easily broken with the Pollard Rho algorithm.

In conclusion, the combination of a large prime p and ASCII values for the variable x can potentially enhance the security of the trapdoor function. However, a comprehensive analysis, consideration of computational efficiency, and adherence to cryptographic best practices are essential before deploying such a function in practical applications. Consulting with cryptographic experts is recommended to ensure the security and appropriateness of the chosen approach.

7. Lambert W Function

With publications by Lambert and Euler in the late 18th century, the Lambert W-function has a long history [13]. Numerous studies have been conducted on the function and its mathematical properties. Two straightforward approximations, one with a maximum relative error of around 15% and the other with a maximum relative error of about 5%, were provided [14]. The Lambert W function is defined to be the multivalued inverse of the function $f(x) = xe^x$. It has many applications in pure and applied mathematics. For a given real number x in the range $-\frac{1}{e} \leq x < 0$ there are two possible real values of $W(x)$. The positive branch is denoted by $W_0(x)$ since in this branch $W_0 \geq -1$. When $W(x) < -1$, it is denoted by $W_{-1}(x) \leq -1$.

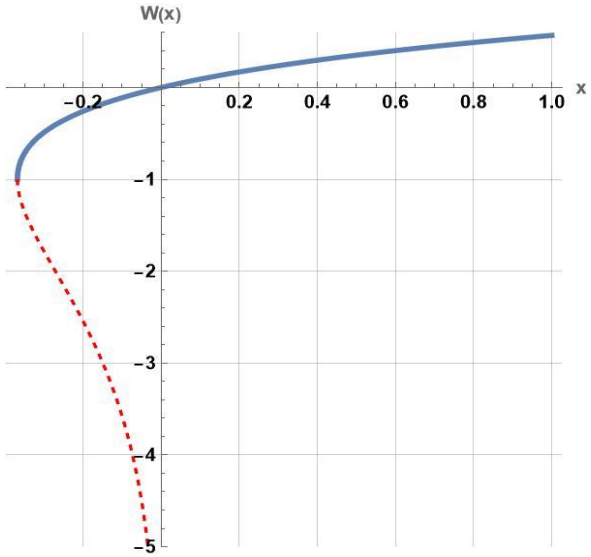

Fig. 1. Two branches of the LambertW Function

8. Calculating the inverse function

Suppose during the encryption procedure in a cryptosystem, we obtain $f(x, p) = c$, which means $x^p p^x = c$, we are required to calculate x given the value of p. We start by taking the p^{th} root of both sides. The steps are shown below.

$$
\begin{aligned}
x^p p^x &= c \\
x p^{\frac{x}{p}} &= c^{\frac{1}{p}} \\
x e^{\ln p^{\frac{x}{p}}} &= c^{\frac{1}{p}} \\
x e^{\frac{x}{p}\ln p} &= c^{\frac{1}{p}} \\
\frac{x}{p}\ln p\, e^{\frac{x}{p}\ln p} &= \frac{1}{p} c^{\frac{1}{p}} \ln p \\
\frac{x}{p}\ln p &= W_0\left(\frac{c^{\frac{1}{p}}\ln p}{p}\right) \\
x &= \frac{p W_0\left(\frac{c^{\frac{1}{p}}\ln p}{p}\right)}{\ln p}
\end{aligned}
$$

Where W_0 is the Lambert W function on the branch $k = 0$ which always generates real numbers.

9. Limitations and Workarounds

The computational efficiency of the proposed trapdoor function is significantly impacted when dealing with large prime values, as the exponent operations involved can escalate the time complexity. This becomes particularly pronounced when the trapdoor function is applied separately to each character in each plaintext, resulting in an exponential rise in time complexity proportional to the number of characters. To address this challenge, a pragmatic solution involves integrating the trapdoor function into block encryption schemes. This approach prevents the primes from reaching unwieldy magnitudes, and diverse permutations of the prime sequence can be allocated to distinct blocks of plaintext. To further expedite the process, advanced computing technologies such as high-power processors, Field-Programmable Gate Arrays (FPGAs), or even quantum computers can be employed. These technologies enable parallel processing, assigning operations in each block to different threads in the processor and facilitating concurrent execution of multiple processes.

Another strategy to mitigate the limitation is to perform the proposed operations in the bitstring domain, where computations are inherently faster and can be executed simultaneously for different blocks. However, it is crucial to note that the encryption function is presented without a modulus, introducing a significant challenge. Computing this function incurs a runtime and space complexity of $O(2^n)$, a massive computational load. Furthermore, computing the inverse of the function necessitates the calculation of the Lambert W-function, for which no exact algorithm is currently known. Approximate algorithms exist and are again very slow [15, 16, 17, 18]. This highlights the intricate nature of the cryptographic processes involved, necessitating careful consideration of both efficiency and security trade-offs in the design and implementation of such trapdoor functions.

10. Potential Improvements

The current trapdoor function can be made useful in the following ways:

1. Introducing a modulo and rewriting equation (1) as

$$f(x,p) \equiv x^p p^x \bmod m \tag{2}$$

If $m = ab$ where a and b are primes, we have an analog to the classic RSA problem.

2. Introducing a prime $q > p$ and rewriting equation (1) as

$$f(x,p) \equiv x^p p^x \bmod q^m \tag{3}$$

This decreases the computational complexity using a fast modular exponentiation algorithm while maintaining an elevated level of security. Using q^m would also involve generating the inverse of p within a Galois Field $\mathrm{GF}(q^m)$.

These improvements not only extend the applicability of the trapdoor function but also enhance its computational efficiency and security, aligning it with contemporary cryptographic standards.

10.1 Similarities with RSA and ElGamal

The improved function introduces a modulo operation with m and exhibits similarities with both RSA and ElGamal cryptographic schemes. Let us discuss its relation to RSA and ElGamal:

1. **Relation to RSA:**
   - The introduced modulo operation and the use of $x^p p^x \bmod m$ in the improved function bear a resemblance to the RSA algorithm. In RSA, the public key operation involves computing $m \equiv x^e \bmod n$, and the private key operation is $x \equiv m^d \bmod n$, where n is the product of two large primes.
   - The commonality lies in the modular arithmetic and the use of exponentiation. The improved function shares the characteristics of RSA in terms of exponentiation and modular reduction, but the specifics of the mathematical operations and the key generation process may differ.
   - Similarities with RSA may make the improved function suitable for scenarios where RSA-like properties are desired, such as in digital signatures or encryption schemes.
2. **Relation to ElGamal:**
   - The improved function also shows similarities with the ElGamal cryptosystem, particularly in its reliance on modular arithmetic and exponentiation. In ElGamal, the public key consists of a large prime p, a primitive root g, and $y \equiv g^x \bmod p$, while the private key is x.
   - The introduction of $x^p p^x \bmod m$ in the improved function aligns with the ElGamal structure, especially with the exponentiation and modulo operations. However, the specific construction and the involvement of p in both the base and exponent differ from the ElGamal setup.
   - The improved function may be considered as a variant that shares certain cryptographic properties with ElGamal, making it relevant for applications where ElGamal-like structures are preferred.
3. **Security Considerations:**
   - Like RSA and ElGamal, the security of the improved function depends on the choice of parameters, especially the prime p and the modulus m. The size of p and m plays a crucial role in resisting attacks such as factorization or discrete logarithms.
   - The improved function should be analyzed for its resistance to common cryptographic attacks and fulfill the necessary security assumptions for its intended use.

In summary, the improved function $f(x,p) \equiv x^p p^x \bmod m$ exhibits relations to both RSA and ElGamal in terms of modular arithmetic and exponentiation. While it draws inspiration from these well-established cryptographic schemes, its specific mathematical structure and operational details distinguish it as a unique cryptographic primitive. A thorough analysis of its security, efficiency, and suitability for various applications is essential in determining its practical significance within cryptographic contexts.

The term "unorthodox" is used to describe something that deviates from established or conventional practices, norms, or standards. In the context of the function $f(x,p) \equiv x^p p^x \bmod m$, several aspects contribute to categorizing it as unorthodox in the field of cryptography:

1. **Novelty of Formulation:** The function introduces a non-traditional combination of operations involving exponentiation with both the base x and the exponent p and vice versa, and their product, namely involving prime p and ASCII value x. This departure from more conventional cryptographic primitives makes the function unconventional.
2. **Unique Mathematical Structure:** The mathematical structure of the function, with its specific combination of exponentiation and modular arithmetic involving both x and p, is distinct from widely used cryptographic primitives. This uniqueness contributes to the characterization of the function as unorthodox.
3. **Use of ASCII Values:** The incorporation of ASCII values in the function, where x represents an ASCII value, is atypical in cryptographic algorithms. Cryptographic functions often deal with numerical values rather than character representations, making this inclusion unconventional.
4. **Modulo Operation with Prime Factors:** The use of a modulo operation involving the product of primes $m = ab$ distinguishes the function. While modulo operations are common in cryptography, the specific use of the product of primes in this context is less conventional.
5. **Limited Prevalence in Established Protocols:** Unorthodox functions are often characterized by their limited prevalence in widely adopted cryptographic protocols and standards. If a function significantly deviates from established algorithms like RSA or ElGamal, which have undergone extensive scrutiny and standardization, it can be considered unorthodox.

It's important to note that labeling a function as unorthodox doesn't necessarily imply insecurity or unsuitability for cryptographic use. However, it does suggest that the function may require additional scrutiny, analysis, and validation before being widely accepted or integrated into cryptographic applications. The field of cryptography typically relies on well-established and widely scrutinized algorithms to ensure security and new proposals are subject to thorough evaluation and peer review.

11. Real-Life Scenario: Secure Digital Signatures in IoT networks

The function $f(x,p) \equiv x^p p^x \bmod m$ could find practical application in scenarios where a unique combination of exponentiation, prime factors, and modular arithmetic is advantageous. One potential real-life scenario is the generation of secure digital signatures for authentication in a distributed network environment.

Consider an Internet of Things (IoT) network where devices need to communicate securely and verify the authenticity of messages. The function $f(x,p) \equiv x^p p^x \bmod m$ could be employed in the following manner:

1. **Key Generation:**
   - Select a large prime p and derive a modulus m such that $m = ab$ where a and b are prime factors. The choice of these primes is crucial for security.
   - Generate public and private keys: p is kept private, and m is public. The public key is (p,m), and the private key is the knowledge of p and m.
2. **Signature Generation:**
   - To sign a message x from an IoT device, the device computes the signature using $f(x,p) \equiv x^p p^x \bmod m$. The private key p is used in the exponentiation, ensuring the uniqueness and security of the signature.
3. **Signature Verification:**
   - Other devices receiving the message can verify the signature by computing the function on their end using the public key (p,m) and checking if the result matches the received signature. If the computed value matches the received signature, the message is considered authentic.

Advantages in this Scenario:

- **Security:** The function relies on the difficulty of factoring m into its prime components, contributing to the security of the digital signature.
- **Unique Exponentiation:** The function combines exponentiation with both x and p, adding a unique element to the signature generation process.
- **Suitability for Resource – Constrained Devices:** The function's mathematical structure allows for flexibility in choosing primes and moduli, potentially accommodating resource constraints in IoT devices.

It is important to note that while this scenario illustrates a potential application, thorough security analysis and validation against established cryptographic standards are necessary before deploying such a function in a real-world context. Additionally, ongoing evaluation and adaptation to emerging cryptographic challenges should be considered.

12. Conclusion

Throughout the past, various classes of functions have been put forth as potential trapdoor functions, revealing a complex landscape where the quest for effective trapdoor functions proved more challenging than initially anticipated. The significance of finding suitable trapdoor functions is paramount in cryptography, as the level of security attained by any encryption algorithm is intricately tied to the properties of the trapdoor function it employs. Notably, the prevailing challenge in contemporary cryptographic algorithms lies in their reliance on three arduous mathematical problems: the integer factorization problem, the discrete logarithm problem [19], and the elliptic curve discrete logarithm problem [20]. The vulnerability of these problems to efficient solutions by quantum computers, particularly through Shor's algorithm, underscores the need for innovative and resilient trapdoor functions to secure the future of cryptographic systems.

Recognizing this imperative, this paper embarks on a humble endeavor to introduce a novel trapdoor function grounded in number theory. While presenting a promising avenue for enhancing cryptographic security, the proposed trapdoor function is not without its limitations, particularly in terms of time complexity when scaled to an industrial level. However, the emergence of quantum computing and deeper insights into the intricacies of calculating modular exponents offer a glimmer of hope for overcoming these limitations. In a quantum computing environment, the proposed algorithm is anticipated to exhibit enhanced efficiency, showcasing the potential for transformative advancements in the realm of cryptographic protocols. As we navigate the dynamic landscape of cryptography, the pursuit of innovative trapdoor functions remains a critical endeavor to fortify the resilience of encryption mechanisms against the evolving landscape of computational capabilities.

References

[1] B. S. Fagin, L. C. Baird, J. W. Humphries and D. L. Schweitzer, "Skepticism and cryptography," Knowledge, Technology & Policy, vol. 20, no. 4, pp. 231-242, 2007.

[2] A. C. Yao, "Theory and application of trapdoor functions," in 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982), IEEE, 1982, pp. 80-91.

[3] E. Ochoa-Jim'enez, L. Rivera-Zamarripa, N. Cruz-Cort'es and F. Rodriguez-Henr'iquez, "Implementation of RSA signatures on GPU and CPU architectures," IEEE Access, vol. 8, pp. 9928-9941, 2020.

[4] L. Qiu, Z. Liu, G. C. CF Pereira and H. Seo, "Implementing RSA for sensor nodes in smart cities," Personal and Ubiquitous Computing, vol. 21, pp. 807-813, 2017.

[5] R. Granger, T. Kleinjung and J. Zumbr"agel, "On the discrete logarithm problem in finite fields of fixed characteristic," Transactions of the American Mathematical Society, vol. 370, no. 5, pp. 3129-3145, 2018.

[6] A. M. Odlyzko, "Discrete logarithms in finite fields and their cryptographic significance," in Workshop on the Theory and Application of of Cryptographic Techniques, 1984.

[7] H. I. Hussein and W. M. Abduallah, "An efficient ElGamal cryptosystem scheme," International Journal of Computers and Applications, vol. 43, no. 10, pp. 1088-1094, 2021.

[8] S. Y. Yan, "Primality testing and integer factorization in public-key cryptography," Advances In Information Security, 2009.

[9] K. Rabah, "Review of Methods for Integer Factorization Applied to Cryptography," Journal of applied Sciences, vol. 6, no. 1, pp. 458-481, 2006.

[10] J. Hoffstein, "Integer Factorization and RSA," in An Introduction to Mathematical Cryptography, Springer, 2008, pp. 1--75.

[11] E. Bach, "Toward a theory of Pollard's rho method," Information and Computation, vol. 90, no. 2, pp. 139-155, 1991.

[12] O. Danvy, "The Tortoise and the Hare Algorithm for Finite Lists, Compositionally," ACM Transactions on Programming Languages and Systems, vol. 45, no. 1, pp. 1-35, 2023.

[13] R. M. Corless, G. H. Gonnet, D. E. Hare, D. J. Jeffrey and D. E. Knuth, "On the LambertW function," Advances in Computational mathematics, vol. 5, no. 1, pp. 329-359, 1996.

[14] J. Boyd, "Global approximations to the principal real-valued branch of the Lambert W-function," Applied Mathematics Letters, vol. 11, no. 6, pp. 27-31, 1998.

[15] F. N. Fritsch, R. Shafer and W. Crowley, "Solution of the transcendental equation wew= x," Communications of the ACM, vol. 16, no. 2, pp. 123-124, 1973.

[16] D. Barry, S. Barry and P. Culligan-Hensley, "Algorithm 743: WAPR--a Fortran routine for calculating real values of the W-function," ACM Transactions on Mathematical Software (TOMS), vol. 21, no. 2, pp. 172-181, 1995.

[17] D. Barry, P. Culligan-Hensley and S. Barry, "Real values of the W-function," ACM Transactions on Mathematical Software (TOMS), vol. 21, no. 2, pp. 161-171, 1995.

[18] N. N. Schraudolph, "A fast, compact approximation of the exponential function," Neural Computation, vol. 11, no. 4, pp. 853-862, 1999.

[19] H. Corrigan-Gibbs and D. Kogan, "The discrete-logarithm problem with preprocessing," in Advances in Cryptology--EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part II 37, 2018.

[20] C. Diem, "On the discrete logarithm problem in elliptic curves," Compositio Mathematica, vol. 147, no. 1, pp. 75-104, 2011.

Authors' Profiles

Awnon Bhowmik is a mathematics graduate turned professional software developer, with a keen interest in data science and cryptography. He has worked in the education industry for 8 years under various designations, including working as a substitute teacher in the NYC Department of Education.

How to cite this paper: Awnon Bhowmik, "An Unorthodox Trapdoor Function", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.10, No.1, pp. 31-38, 2024. DOI: 10.5815/ijmsc.2024.01.04