computer sciences and mathematics forum

MDPI

Proceeding Paper

Using Reconfigurable Multi-Valued Logic Operators to Build a New Encryption Technology [†]

Hongjian Wang [1], Shan Ouyang [2], Xunlei Chen [2] and Yi Jin [2,*]

[1] School of Computer Science and Technology, Donghua University, Shanghai 201620, China; hongjian.wang@dhu.edu.cn
[2] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
[*] Correspondence: yijin@shu.edu.cn

[†] Presented at the 2023 Summit of the International Society for the Study of Information (IS4SI 2023), Beijing, China, 14–16 August 2023.

Abstract: Current encryption technologies mostly rely on complex algorithms or difficult mathematical problems to improve security. Therefore, it is difficult for these encryption technologies to possess both high security and high efficiency, which are two properties that people desire. Trying to solve this dilemma, we built a new encryption technology, called *configurable encryption technology* (CET), based on the typical structure of *reconfigurable quaternary logic operator* (RQLO) that was invented in 2018. We designed the CET as a block cipher for symmetric encryption, where we use four 32-quit RQLO typical structures as the encryptor, decryptor, and two *key derivation operators*. Taking advantage of the reconfigurability of the RQLO typical structure, the CET can automatically reconfigure the keys and symbol substitution rules of the encryptor and decryptor after each encryption operation. We found that a chip containing about 70,000 transistors and 500 MB of nonvolatile memory could provide all the CET devices and *generalized keys* needed for any user's lifetime, to implement a practical *one-time pad* encryption technology. We also developed a strategy to solve the current key distribution problem with prestored *generalized key source data* and on-site *appointment codes*. The CET is expected to provide a theoretical basis and core technology for using the RQLO to build a new cryptographic system with high security, fast encryption/decryption speed, and low manufacturing cost.

Keywords: reconfigurable quaternary logic operators; one-time pad; symmetric encryption; real-time communication encryption; digital file encryption/decryption

check for updates

Citation: Wang, H.; Ouyang, S.; Chen, X.; Jin, Y. Using Reconfigurable Multi-Valued Logic Operators to Build a New Encryption Technology. *Comput. Sci. Math. Forum* **2023**, *8*, 99. https://doi.org/10.3390/cmsf2023008099

Academic Editor: Chin-Tser Huang

Published: 10 April 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Digital information has become an important asset in modern society, and it is the main battlefield to ensure national security. Therefore, many countries are focusing on the establishment of digital information encryption technology with a high security level, real-time speed, compact equipment, easy operation, and many applications.

Information encryption technologies, such as AES [1,2] and SM4 [3], rely on complex symbol substitution processes and long digital keys to ensure the security of ciphertexts. But, the difficulty of distributing and memorizing keys limits the number of the keys that can actually be used. The same key is inevitably used multiple times, which leads to the potential risk of being cracked. Currently, the main method of changing keys involves relying on public key technology to distribute keys. However, by repeatedly using a public/private key pair, the risk of the private key being cracked increases rapidly, and the computational effort to update the public/private key pair is large. As a result, the BB84 quantum key distribution protocol, which is still in the experimental research stage, has received a lot of attention. In order to make up for the shortcomings of digital information encryption technology, on the one hand, people have strengthened network antiattack technologies such as identity authentication, electronic signatures, and access address tracking. On the other hand, people have avoided files from being opened in illegal environment by reading the current network segment, current processor device number, and other environmental information. Moreover, people invented digest generation technology, which can identify information without transmitting the original message, and the homomorphic encryption technology [4], which can perform certain operations on encrypted data without first having to decrypt the data.

Looking at the current information security technology, we can see that it is based on high computing power and high network bandwidth, which are also assets on which attackers rely. Therefore, computer and network capabilities have become the key factors in the success of both attackers and defenders, and attackers often have an advantage in these areas over normal users. To reverse this passive situation, it is necessary to establish a new encryption technology that is convenient and low-cost and can effectively protect digital files and streaming data.

At the end of 2018, following the *reconfigurable ternary* (i.e., three-valued) *optical processor* structure [5,6], a typical structure of reconfigurable *multi-valued logic* (MVL) electronic operator was invented by Jin et al. [7,8], opening the way for new technology for encrypting and decrypting digital files and streaming data information [9]. After more than four years of research, a new encryption technology, which uses the 32-quit *reconfigurable quaternary (i.e., four-valued) logic operator* (RQLO) to realize a practical *one-time pad* (OTP) method, has been theoretically and experimentally verified. This paper gives a brief introduction to this new encryption technology—*configurable encryption technology* (CET).

The novelty of CET is the use of the hardware of reconfigurable MVL operators (RQLO typical structures, to be specific) for encryption, where we can leverage the reconfigurability of RQLO to generate massive encryptor/decryptor pairs for achieving high-security and efficient encryption technology. There are currently two main types of MVL applications in the field of cryptography: one involves using MVL operations to construct basic arithmetic operators and then implementing existing encryption algorithms [10,11]; the other involves using several simple MVL operators to perform complex symbol array transformation, realizing encryption/decryption [12,13] or generating pseudorandom numbers [14,15]. We have not found any scheme for directly encrypting/decrypting digital information using MVL operators, as what we achieve with the CET discussed in this paper. We think the reason for this is that before 2018, when the reconfigurable MVL electronic processor has not been invented, it was difficult to construct massive different MVL operators. But now, using the typical structure of reconfigurable MVL operator and the corresponding *reconfiguration instructions* (RIs), inexhaustible encryptor/decryptor pairs for almost any user lifetime can be provided in a low-cost encryption device.

2. Configurable Encryption Technology (CET)

CET mainly consists of three parts, namely, (1) massive encryptor/decryptor pairs based on the 32-quit RQLO typical structure; (2) generalized key (GK) and generalized key source data (GKSD); (3) GKSD pre-storage and appointment code (AC) negotiation.

In view of the fact that current computers mostly use 64-bit processors, we use 64-bit data as the basic block of plaintext to maximize the use of computer hardware resources, where two-bit data represent one quit quaternary data. So, a 32-quit RQLO is required to perform quaternary logic operations on 64-bit data. In this study, we used the example of encrypting/decrypting 64-bit plaintext/ciphertext blocks with 32-quit RQLO to discuss CET's basic idea in detail.

2.1. Massive Encryptor/Decryptor Pairs Based on RQLO Typical Structure

According to the definition of quaternary logic operations [7,8], Table 1 shows a truth table representing the quaternary logic operation rules for the RQLO: it has two input variables A and B, and the operation result C array has 16 elements, i.e., C_{hk} (h, k = 0, 1, 2, 3). So, C has 4^{16} different results; that is, there are nearly 4.3 billion different 1-quit quaternary logic operators. Suppose that a quaternary logic operator is encryptor F, where the input variable B is key K, the input variable A is plaintext M, and the operation result C_{hk} is ciphertext S. Then, the encryption process can be written as S = F(M, K). As for decryption, the corresponding decryptor F^{-1} is also a quaternary logic operator, where the input variable input variable B is still the key K, while the input variable A is the ciphertext S, and the operation result C_{hk} is the decrypted plaintext M. So, the decryption process can be written as M = F^{-1}(S, K). Here, F and F^{-1} are a pair of quaternary logic operators. The value ranges of A, B and C_{hk} are all {00, 01, 10, 11}, indicating that 1 quit of M, S, and K can be implemented by 2 bits.

Table 1. Truth table of quaternary logic operation.

A \ B	**B_0(00)**	**B_1(01)**	**B_2(10)**	**B_3(11)**
A_0 (00)	C_{00}	C_{01}	C_{02}	C_{03}
A_1 (01)	C_{10}	C_{11}	C_{12}	C_{13}
A_2 (10)	C_{20}	C_{21}	C_{22}	C_{23}
A_3 (11)	C_{30}	C_{31}	C_{32}	C_{33}

If any column in the truth table has two identical result elements, i.e., $C_{hk} = C_{hg}$ and $k \neq g$ (h, k, g = 0, 1, 2, 3), then the corresponding quaternary logic operator cannot be used as an encryptor F because C_{hk} is decrypted as two possible values. Therefore, there are $(4!)^4$ (i.e., 331,776) different 1-quit encryptors in total, among the nearly 4.3 billion different 1-quit quaternary logic operators.

Since CET uses 64-bit data as the basic plaintext block, 2 bits form 1-quit data. So, encryptor F can be viewed as either 32 individual 1-quit encryptors or 1 32-quit encryptor. If each quit of F is required to be different from the others, a constraint relationship is established among all quits of F. Thus, each quit of F can no longer be considered independently but rather as a whole. (To enhance the correlation between quits and the integrity of the 32-quit encryptor, various confusion and diffusion techniques from modern block ciphers can be superimposed on CET.) The total number of F is calculated as a 32 permutation of 331,776:

$$N_F = 331{,}776!/(331{,}776-32)! = 331{,}776!/331{,}744! > 4.64 \times 10^{176}. \tag{1}$$

If a person uses 1000 encryptors per second, they will use 3.1536×10^{12} encryptors in 100 years. So, it can be easily concluded that the probability that the same F will be used twice within a person's lifetime is almost zero. This lays the foundation for the realization of OTP encryption technology.

However, constructing single-use 10^{12} pairs of encryptor/decryptor hardware costs too much, and the finished product is too large and too heavy to be afforded by an individual. So, no one could have enough encryptors for a lifetime of use, until the typical structure of reconfigurable MVL operators based on the *decrease-radix design principle* [16] was invented in 2018 [7]. Based on this invention, we designed a RQLO typical structure for CET, as shown in Figure 1. It can be estimated that about 17,600 transistors are needed to construct a 32-quit RQLO typical structure, which consists of 32 1-quit RQLO typical structures.

In the schematic diagram of a 1-quit RQLO typical structure shown in Figure 1, 2-bit data represent 1-quit data, where the ith quit of the quaternary input data a is given by a pair of binary data ($a_i^1 a_i^0$): a_i^1 is the high-bit value, and a_i^0 is the low-bit value. Similarly, the ith quit of the quaternary input data b is also given by a pair of binary data ($b_i^1 b_i^0$); the ith quit of the quaternary output data c is given by a pair of binary data ($c_i^1 c_i^0$). There are four identical subcircuits in the main circuit in Figure 1, marked by four red dotted-line boxes, P_i0, P_i1, P_i2, and P_i3; the subcircuits have the same structure as shown in P_i0.

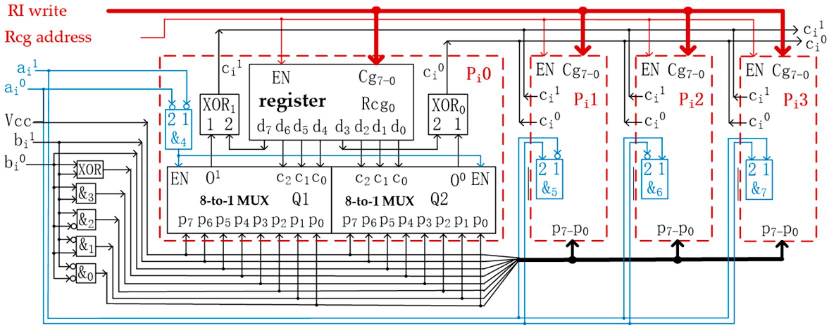

Figure 1. Schematic diagram of 1-quit RQLO typical structure.

For CET, the most important part in the RQLO typical structure is the reconfiguration registers Rcg0, Rcg1, Rcg2, and Rcg3 (note that only Rcg0 is drawn in Figure 1), and the values written in these four registers determine the logic operation function of the typical structure. Rcg0, Rcg1, Rcg2, and Rcg3 are combined into a 32-bit register RGi. RGi is called the ith reconfiguration register, and the RI written to RGi is called the ith RI. Thus, the 32-quit RQLO typical structure has 32 RGi (i = 0, 1, . . ., 31), which are collectively called the reconfiguration register RG of the typical structure. RG has 1024 bits, and the corresponding RI for the 32-quit RQLO typical structure has 128 bytes.

By writing the 128-byte RI to each RG of two 32-quit RQLO typical structures, these two structures become two symbol substitution operators. If these two operators are suitable as encryptors and their operation rules are exactly opposite, they become an encryptor/decryptor pair. Therefore, N different encryptor/decryptor pairs need 2N × 128 bytes of RIs plus two 32-quit RQLO typical structures to be prepared.

Based on the reconfigurability of RQLO typical structure, the new encryption technology can automatically reconfigure the keys and symbol substitution rules of the encryptor and decryptor after each encryption operation. Users can obtain different encryptors/decryptors simply by writing different RIs into the reconfiguration register of the typical structure.

2.2. *Generalized Key and Generalized Key Source Data*

The CET inherits not only the keys of modern cryptography but also the key derivation strategy and initial key. It also uses the RQLO typical structure as the *key derivation operator* (KDO). We denote the initial key by K_0, the ith derived key by K_i, where K_i is derived from K_{i-1} and M_{i-1} via KDO. For the 64-bit plaintext block, K_0 and K_i are also 64-bit data, and the corresponding KDO is still a 32-quit RQLO typical structure.

For a given plaintext, different encryptors F generate different ciphertexts, which is the same as the role of key K. Moreover, different initial keys K_0 and KDOs generate different derived keys K_i. Here, K_0 and all K_i form K. Therefore, F, K_0, and KDO jointly determine the ciphertext, and, in this paper, the factors affecting the ciphertext are collectively referred to as a *generalized key* (GK). Since the operation rules of F, F^{-1}, and KDO are all determined by their RIs, the GK can be represented by a K_0 and three RIs. Given that it takes 8 bytes to save a K_0, while 128 × 3 bytes to save the RIs of an encryptor/decryptor pair and a KDO, we call the 392 bytes the *generalized key source data* (GKSD), which determine a specific GK.

After a simple calculation, it can be concluded that 1.0512×10^5 encryptor/decryptor pairs, 1.0512×10^5 KDOs, and 3×10^7 initial keys together can provide enough GKs, which ensure that a user does not have to use the same GK twice for at least 100 years (if the user consumes 1000 GKs per second on average). These GKs require less than 300 MB of memory. In addition, no more than 200 MB of memory is needed to store the encryption and decryption programs. Thus, the hardware for constructing a GK device that is inexhaustible in a human lifespan is four 32-quit RQLO typical structures constructed with about 70,000 transistors (approximately 0.0016 mm^2 of circuit area is required with 14 nm IC technology), plus 500 MB of nonvolatile memory. In this paper, we call this device *configurable encryption chip* (CEC), and we describe the CEC's internal structure in Section 3. By simply writing completely a different GKSD for every CEC, each user will not use the same GK as the others.

2.3. Generalized Key Source Data Prestorage and Appointment Codes

Although the RQLO typical structure and initial key together can provide a large number of GKs, it is necessary to randomly select, with uniform distribution probability, all the used GKs to make the CET practical OTP technology. We created a strategy of storing the randomly selected GKSD in advance for each CEC. The GKSD is written to the CEC by a special fully automatic machine without human intervention, and it uses physical methods to generate true random numbers with uniform distribution probability. The prestored GKSD cannot be read from the CEC.

Since each GKSD written in each CEC is unique, no two CECs have the same GKSD. Thus, a ciphertext generated by a particular CEC can only be decrypted by that CEC. This is an important feature of CET. Also, we use a strategy to ensure that there are no equivalent GKSDs written in each CEC, where two equivalent GKs are different but transform the same plaintext into the same ciphertext. Therefore, as long as each GK is used only once, no equivalent GK will be used twice within 100 years, achieving a practical OTP effect.

Another core aspect of CET is ensuring that both the encryption and decryption parties use the paired GK each time. So, both parties must hold the exact same CEC and agree on which GK to use. We designed the *appointment code* (AC) to reach this agreement. The most straightforward AC is the serial number of the prestored GKSD. Since only the encryption and decryption parties hold the same GKSD, which is different from that of the other CECs, a third party who cracks the AC cannot obtain the correct GKSD. The AC in different CECs stands for different GKSDs, and it does not need to be kept secret.

We can either use "date + time" to generate an AC, or generate it cumulatively. If we want to randomly select a GK from a prestored GKSD, a random number can be generated by counting the number of particular binary strings in the current plaintext, and then the AC can be derived from the random number.

The CET application includes two main branches: real-time stream data encryption/decryption for communication systems and digital file encryption/decryption for storage systems [9]. For the real-time encryption/decryption scenario of communication systems, both parties negotiate the AC to be used for the next encrypted communication at the end of current encrypted communication, and the AC is checked or modified again before the start of the next encrypted communication. This not only prepares for the next encrypted communication but also protects the information security of the current encrypted communication, because even if the current encrypted communication has been eavesdropped by a third party, no one can decrypt the eavesdropped information since no one can obtain the flushed GK. For the digital file encryption/decryption scenario, the AC is stored at the beginning of the ciphertext file. Although anyone can obtain the AC from the ciphertext file, the correct GK for decryption can only be obtained from the paired GKSD stored on the legal user's CEC.

It can be seen that an important role of the AC is activating a predistributed GK using its prestored GKSD. With the help of the prestored GKSD and AC, key distribution can be eliminated, making the encryption system simpler and more secure.

3. Basic Structure of Configurable Encryption Chip

To implement the CET, we designed a CEC structure, as shown in Figure 2. It contains read-only memory, an encryption component, a decryption component, an AC mechanism, and a controller (which is not depicted in Figure 2). The four 32-quit RQLO typical structures are denoted by F, F^{-1}, and two KDOs, respectively.

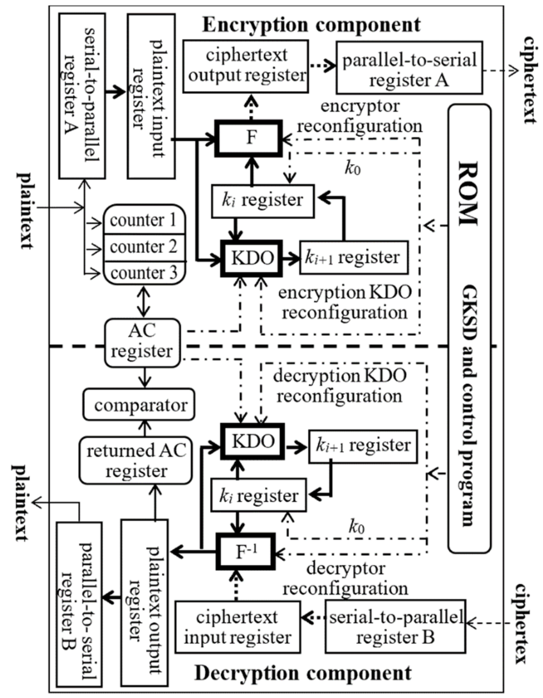
Encryption component
ciphertext output register
parallel-to-serial register A
ciphertext
serial-to-parallel register A
plaintext input register
F
encryptor reconfiguration
k0
ROM
GKSD and control program
plaintext
ki register
counter 1
counter 2
counter 3
KDO
ki+1 register
encryption KDO reconfiguration
AC register
decryption KDO reconfiguration
comparator
returned AC register
KDO
ki+1 register
ki register
k0
F-1
decryptor reconfiguration
plaintext
parallel-to-serial register B
plaintext output register
ciphertext input register
serial-to-parallel register B
ciphertex
Decryption component

Figure 2. Schematic diagram of internal CEC structure.

In the encryption component, F and KDO share the plaintext input register and the key register K_i. The encryptor F generates the current block of the ciphertext from the current values of these two registers and sends it to the ciphertext output register, while KDO generates the derived key K_{i+1} for encrypting the next plaintext block from the current value of the two input registers and sends it to the K_{i+1} register.

In the decryption component, F^{-1} generates the current block of plaintext from the key register K_i and the ciphertext input register and sends it to the plaintext output register. KDO generates the derived key K_{i+1} for decrypting the next ciphertext block from the K_i register and the plaintext output register and sends it to the K_{i+1} register.

As discussed in Section 2.2, the read-only memory needs 500 MB to store the GKSD (composed of 3×10^7 initial keys, the RIs for 1.0512×10^5 encryptor/decryptor pairs, and the RIs for 1.0512×10^5 KDOs) and the control program. The AC is generated by three counters (Counter 1, Counter 2, and Counter 3), where each count the occurrences of three particular binary strings in the input plaintext. When the CEC receives a communication completion command, it connects the values of the three counters to form the AC, which is a random number needed for the next encryption communication.

3.1. AC Negotiation Process with CEC

In a real-time encrypted communication application scenario, the process of negotiating AC between the communication parties is roughly as follows:

(1) Party A, who requests to end the communication, sends AC to the other party B.
(2) B saves the received AC and returns it to sender A.
(3) A receives the returned AC and compares it with the saved AC via a comparator (see Figure 2). If the returned AC and the saved AC are the same, a signal of "AC negotiation success" is given.
(4) If the returned AC fails to match the saved AC, A sends the saved AC again.
(5) If the returned AC keeps failing to match the saved AC 8 times in a row, a signal of "AC negotiation failure" is given.

4. Experiment

For verifying the effectiveness of the CET, we implemented an FPGA-based experimental system to encrypt and decrypt real-time video streaming. In this system, we selected 32 bits data as the plaintext block and implemented 16-quit RQLOs on FPGA to encrypt, decrypt, and derive keys. The system captured 1080P-resolution video streaming with a camera in real time. Meanwhile, the encryption and decryption modules, which took 150 LUTs and 128 FFs in total in terms of resource utilization, were capable of processing 6.21 Gbit/s streaming data, showing that they had real-time processing capability. From the implementation timing summary in Vivado 2021.2 software, the path delay from the plaintext input to the ciphertext output was 4.793 ns. The maximum amount of data that can be processed by a 16-quit RQLO was $32 \div (4.793 \times 10^{-9})$ bit/s $\approx 6.676 \times 10^9$ bit/s > 6.21 Gbit/s. The experimental system worked properly if we set the clock period to 5 ns, which is a frequency of 200 MHz.

The initial key K_0, the encryptor F, and the KDO together form the generalized key. Thus, the size of the key space is equal to the multiplication of the space sizes of the three generalized key components. In our experimental system, the block size had 16 quits (32 bits), so the K_0 space size = $2^{32} \approx 4.29 \times 10^9$, the encryptor space size = $(4!)^{(4\times16)} \approx 2.15 \times 10^{88}$, and the KDO space size = $4^{(4\times4\times16)} \approx 1.34 \times 10^{154}$. Thus, the key space was $4.29 \times 10^9 \times 2.15 \times 10^{88} \times 1.34 \times 10^{154} \approx 1.23 \times 10^{251}$. This huge key space can resist brute-force attacks and provide conditions for practical OTP technology based on RQLOs.

5. Conclusions

The reconfigurable MVL operators created in 2018 provided the basis for building CET that is more secure, easier to use, and lower in cost compared with other encryption methods based on complex algorithms or difficult mathematical problems. In this paper, we discussed the theory and implementation of CET with a practical OTP effect using 32-quit RQLO typical structures. The important CET results can be summarized as the following three advantages: Firstly, it is secure with a practical OTP security level. Secondly, its encryption/decryption speed is fast, and it can handle the real-time encryption tasks of modern communication systems. Thirdly, it solves the current key distribution problem with prestored GKSD and AC. Using CET, we are currently developing a USB device for digital file encryption and a network communication encryption/decryption device inserted between a computer and a network device.

Author Contributions: Conceptualization, H.W. and Y.J.; methodology, H.W. and Y.J.; software, H.W., S.O. and X.C.; validation, H.W., S.O., X.C. and Y.J.; formal analysis, H.W., S.O. and X.C.; investigation, H.W. and Y.J.; resources, H.W., S.O. and Y.J.; data curation, H.W., S.O. and X.C.; writing—original draft preparation, H.W.; writing—review and editing, H.W., S.O., X.C. and Y.J.; visualization, H.W.; supervision, Y.J.; project administration, H.W. and Y.J.; funding acquisition, H.W. and Y.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the "Shanghai Pujiang Program" from Shanghai Municipal Human Resources and Social Security Bureau under grant No. 21PJD001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Blahut, R.E. *Cryptography and Secure Communication*; Cambridge University Press: Cambridge, UK, 2014.
2. Schneier, B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
3. Zhang, B.; Jin, C. Practical security against linear cryptanalysis for SMS4-like ciphers with SP round function. *Sci. China Inf. Sci.* **2012**, *55*, 2161–2170. [CrossRef]

4. Acar, A.; Aksu, H.; Uluagac, A.S.; Conti, M. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv. (Csur)* **2018**, *51*, 1–35. [CrossRef]
5. Jin, Y.; Wang, H.; Ouyang, S.; Zhou, Y.; Shen, Y.; Peng, J.; Principles, X.L. Structures, and implementation of reconfigurable ternary optical processors. *Sci. China Inf. Sci.* **2011**, *54*, 2236–2246. [CrossRef]
6. Wang, H.; Song, K. Simulative method for the optical processor reconfiguration on a dynamically reconfigurable optical platform. *Appl. Opt.* **2012**, *51*, 167–175. [CrossRef] [PubMed]
7. Jin, Y.; Ouyang, S.; Pan, Z.; Wang, Y.; Shen, Y.; Peng, J.; Zhou, S.; Liu, Y.; Chen, X. Many-Bit, Groupable, Reconfigurable Multi-Valued, Electronic Operator and Its Construction Method. China Invention Patent 201811567284.7, 20 December 2018; The Patent Cooperation Treaty (PCT/CN2019/070318), 4 January 2019. (In Chinese)
8. Wang, H.; Wu, Y.; Ouyang, S.; Chen, X.; Shen, Y.; Jin, Y. The design and implementation of reconfigurable quaternary logic processor. In Proceedings of the 22nd International Conference on Parallel and Distributed Computing, Applications and Technologies, Guangzhou, China, 17–19 December 2021; pp. 142–149.
9. Wang, H.; Jin, Y.; Jin, S.; Wang, Y. An Encryption and Decryption Method, Device and Communication System Thereof. China Invention Patent 202110801365.4, 15 July 2021; The Patent Cooperation Treaty (PCT/CN2021/110681), 4 August 2021. (In Chinese)
10. Baba, Y.; Miyamoto, A.; Homma, N.; Aoki, T. Multiple-valued constant-power adder for cryptographic processors. In Proceedings of the 39th International Symposium on Multiple-Valued Logic, Naha, Japan, 21–23 May 2009; pp. 239–244.
11. Homma, N.; Saito, K.; Aoki, T. Formal design of multiple-valued arithmetic algorithms over Galois fields and its application to cryptographic processor. In Proceedings of the 2012 IEEE 42nd International Symposium on Multiple-Valued Logic, Victoria, BC, Canada, 14–16 May 2012; pp. 110–115.
12. Sokolov, A.; Zhdanov, O. Prospects for the application of many-valued logic functions in cryptography. In Proceedings of the International Conference on Computer Science, Engineering and Education Applications (ICCSEEA 2018), Kiev, Ukraine, 18–20 January 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 331–339.
13. Zakaria, S.B.; Navi, K. Image encryption and decryption using exclusive-or based on ternary value logic. *Comput. Electr. Eng.* **2022**, *101*, 108021. [CrossRef]
14. Ali, M.A.; Emran, A.; Habib, M.A.; Nadim, M.; Kusaka, T.; Nogami, Y. Pseudo random ternary sequence and its autocorrelation property over finite field. *Int. J. Comput. Netw. Inf. Secur.* **2017**, *9*, 54–63. [CrossRef]
15. Bykovsky, A.Y. A multiple-valued logic for implementing a random oracle and the position-based cryptography. *J. Russ. Laser Res.* **2019**, *40*, 173–183. [CrossRef]
16. Yan, J.; Jin, Y.; Zuo, K. Decrease-radix design principle for carrying/borrowing free multi-valued and application in ternary optical computer. *Sci. China Ser. F Inf. Sci.* **2008**, *51*, 1415–1426. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.