

Received June 7, 2020, accepted June 30, 2020, date of publication July 6, 2020, date of current version July 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3007335

Provably Secure Encryption Schemes With Zero Setup and Linear Speed by Using Rubik's Cubes

PING PAN¹, YUN PAN², ZHEN WANG³, AND LICHENG WANG^{1,3}, (Member, IEEE)

¹School of Mathematics and Computer Science, Shaanxi University of Technology (SNUT), Hanzhong 723000, China

²State Key Laboratory of Media Convergence and Communication, Communication University of China (CUC), Beijing 100024, China

³State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (BUPT), Beijing 100876, China

Corresponding author: Yun Pan (pany@cuc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61972050, in part by the Shaanxi University of Technology (SNUT) Doctoral Research Foundation under Grant SLGQD13-24, and in part by the Open Foundation of the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, under Grant SKLNST-2020-2-16.


ABSTRACT Recently, new paradigms for designing modern cryptographic schemes were proposed based on Rubik's rotations. However, most of them lack rigorous provable security reductions. Enlightened by this interesting progress, we present a novel method for designing encryption schemes by using Rubik's groups. Different from most naive designs of permutation ciphers based on Rubik's cubes, our proposals are probabilistic encryption schemes that combine some of the newest cryptographic primitives with modern coding theory. More specifically, under the intractability assumption of the conjugacy decision problem over Rubik's groups, the proposed schemes have provable security reductions (in the random oracle model). Furthermore, the proposed schemes have two remarkable performance advantages: zero setup and linear encryption/decryption speed. In addition, the processes of encoding/encryption and decryption/decoding are demonstrated graphically.

INDEX TERMS Rubik's cube, encryption, provable security, zero setup, linear speed.

I. INTRODUCTION

As an ancient, heuristic and classic cryptographic method, the permutation cipher is not new to us. The Rubik's cube, perhaps one of the best-selling iconic puzzle tools, is also well known to us, even in our childhood. Recently, very interesting progress has been made by researchers trying to bridge these subjects: many cryptographic schemes, such as Cayley hash functions [17], [18], key agreement protocols [16], image encryption schemes [5], [12], digital watermarking schemes [26], and zero-knowledge protocols [21], were proposed based on Rubik's groups. However, most of them lack provable security reductions, and some even lay their security basis on a *taken-for-granted* hardness assumption: recovering of a Rubik's cube with random configuration (RRC for short). Today, we know that the RRC problem over a $3 \times 3 \times 3$ Rubik's cube is so easy that it can be solved within 20 steps [19].

Therefore, it is interesting to design new Rubik's cryptographic schemes by following the paradigm of

The associate editor coordinating the review of this manuscript and approving it for publication was Junggab Son .

modern cryptography: provable security reductions under well-established models and falsifiable intractability assumptions. In this paper, we present two encryption schemes by using Rubik's groups. Compared to the existing permutation cipher based on Rubik's cubes, our proposals have the following essential differences:

- Our proposals couple two mainstream methods for designing modern cryptography: permutation and substitution. This technique contribution comes from an easy but seemingly neglected, method for mapping messages as arrows with four different directions and then embedding them onto the 54 facets of the Rubik's cube.
- Our proposals are probabilistic encryption schemes that, under the intractability assumption of the conjugacy decision problem over Rubik's groups, have rigorous provable security reductions (in the random oracle model).
- Last but not less significant, our proposal has two remarkable performance advantages: zero-setup and linear encryption/decryption speed.

In addition, the processes of encoding/encryption and decryption/decoding are demonstrated graphically, and all related source codes are open accessible.

The rest of the contents are organized as follows. Necessary preliminaries, including group theoretical aspects about Rubik's groups and intractability assumptions, are given in Section II. Our main contributions, including an encoding/decoding method, two encryption schemes, and security reductions, are presented in Section III. Performance evaluations and tests are provided in Section IV, and concluding remarks are given in Section V.

II. RUBIK'S GROUPS AND INTRACTABILITY ASSUMPTIONS

Let us take a $3 \times 3 \times 3$ Rubik's cube as an example. This Rubik's cube consists of 54 small facets, numbered from 1 to 54, located in 6 faces, labelled U, L, F, R, D and B, representing the upper face, left face, front face, right face, down face and back face, respectively (Figure 1). Each face can rotate 90° clockwise or anti-clockwise each time.¹ Each rotation incurs a new rearrangement of the 54 facets in the 6 faces, and we call this a *configuration* (see Figure 2 for an example).

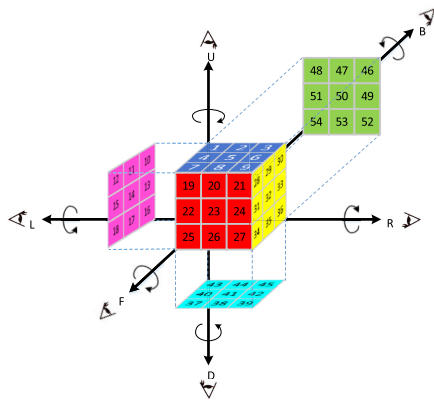


FIGURE 1. Facet numbers and the original configuration.

Clearly, all the possible configurations of the Rubik's cube consist of a subgroup, denoted \mathfrak{R} , of the symmetric group \mathcal{S}_{48} considering that the six center facets remain, although rotated, at the center of the corresponding faces. In fact, the order of \mathfrak{R} is [9] (cf. Page 93)

$$|\mathfrak{R}| = 2^{27} \cdot 3^{14} \cdot 5^3 \cdot 7^2 \cdot 11 \approx 4.3 \times 10^{19} \approx 2^{65}.$$

However, in this paper, we associate a finite generated group to the Rubik's cube in the following way (cf. Page 92 of [9] for more details):

$$\mathfrak{R} = \left\langle U, L, F, R, D, B \left| \begin{array}{l} U^4 = L^4 = F^4 = \\ R^4 = D^4 = B^4 = 1 \end{array} \right. \right\rangle, \quad (1)$$

¹To decide whether a rotation is clockwise or anti-clockwise, one should pay attention to the view angles, which are marked as eye icons near by the six face letters.

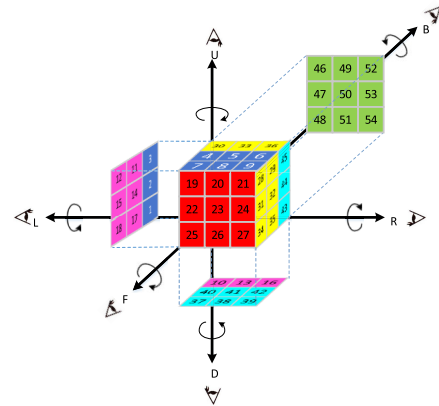


FIGURE 2. Configuration after rotating the face B.

where 1 is the identity that indicates an empty rotation or, equivalently, doing nothing. That is, \mathfrak{R} takes each face rotation as a generator, and the relations among the generators come from the fact that 4 continuous rotations of a face 90° clockwise result in the original configuration of the Rubik's cube. Moreover, the reverse of each face rotation (i.e., anti-clockwise 90° rotation), is equivalent to rotating the same face 3 times continuously, and for convenience, we use U', L', F', R', D', B' to indicate the reverse rotations of the corresponding faces. Apparently, \mathfrak{R} is a non-Abelian group, and thus, the conjugator search problem, defined below, over \mathfrak{R} is nontrivial.

Definition 1 (Conjugacy Decision Problem, CDP): Given a group G and two elements $x, y \in G$, decide whether x and y are conjugate to each other (i.e., $\exists z \in G$ such that $x = z^{-1}yz$).

Definition 2 (Conjugator Search Problem, CSP): Given a group G and two elements $x, y \in G$ that are conjugate, find $z \in G$ such that $x = z^{-1}yz$.

Apparently, for any Abelian group G , both the CDP and CSP are trivial, since every pair of two elements $(x, y) \in G$ are conjugated to each other and every element $z \in G$ can be viewed as a conjugator of the pair (x, y) . However, for a non-Abelian group G , this problem is nontrivial. In fact, in the generic group model, the CDP is unsolvable [15]. On the one hand, we know that for the permutation group, the CDP has no polynomial time methods [20] (cf. Page 53). On the other hand, during the past two decades, based on the intractability assumption of the CSP and CDP over braid groups, several cryptographic schemes were proposed by using braid groups [1], [6], [10], [11], [22]–[25]. Considering that the Rubik's group is a subgroup of the permutation group \mathcal{S}_{48} , which is in turn a subgroup of the braid group \mathcal{B}_{48} [7], this progress gives us the confidence to lay the security of our new encryption scheme on the intractability assumptions of the conjugacy problems, including CDP and CSP, over the Rubik's groups.

III. OUR PROPOSALS

The general architecture of our proposal is given below in Figure 3, of which the encryption (resp. decryption)

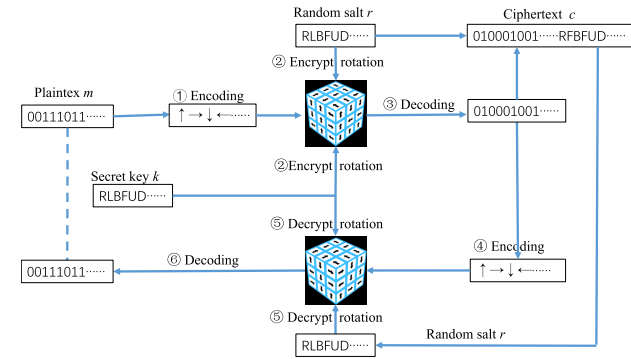


FIGURE 3. General architecture of the proposal.

algorithm consists of steps ①, ②, and ③ (resp. ④, ⑤, and ⑥), respectively. The details of these steps are given in the subsequent subsections.

A. ENCODING/DECODING METHODS AND VISUALIZATION

With the purpose of encrypting messages by using the Rubik's cube, we first need to design a method for encoding messages on the Rubik's cube. Different from the traditional method of describing message letters directly on the Rubik's facets, we introduce the following encoding and decoding method:

- **Encode.** We can use two bits to indicate four arrows with different directions, i.e. $\uparrow, \rightarrow, \downarrow$ and \leftarrow . Now without loss of generality, we assume that each message is a 108-bit string² and each message can be described on the 54 Rubik's facets as a string of arrows. That is, to encode a message $m = (m_1m_2 \cdots m_{108})_2$ on a $3 \times 3 \times 3$ Rubik's cube, we use the following steps:
 - Let $f_i = \alpha(m_{2i-1}m_{2i})$ for $i = 1, \dots, 54$, where $\alpha(\cdot)$ maps a 2-bit string to an arrow, i.e., $\alpha(00) = \uparrow, \alpha(01) = \rightarrow, \alpha(10) = \downarrow$ and $\alpha(11) = \leftarrow$, while f_i indicates the arrow assigned to the i -th facet.
 - Assign the 54 facets (i.e. f_1, \dots, f_{54}) to the six faces of the Rubik's cube as if it were the original configuration³.
- **Decode.** The reverse process of encoding: Given a configuration, not necessarily the original one, of a $3 \times 3 \times 3$ Rubik's cube with each facet assigned an arrow outputs a 108-bit string $m = (m_1m_2 \cdots m_{108})_2$ as follows:
 - Regard the configuration as if it were original and then number the 54 facets on the six faces from 1 to 54 (ref. Figure 1.(a)). Then, assign f_i as the arrow on the i -th facet for $i = 1, \dots, 54$.
 - Let $m_{2i-1}m_{2i} = \alpha^{-1}(f_i)$ for $i = 1, \dots, 54$, where $\alpha^{-1}(\cdot)$ indicates the reverse process for transform-

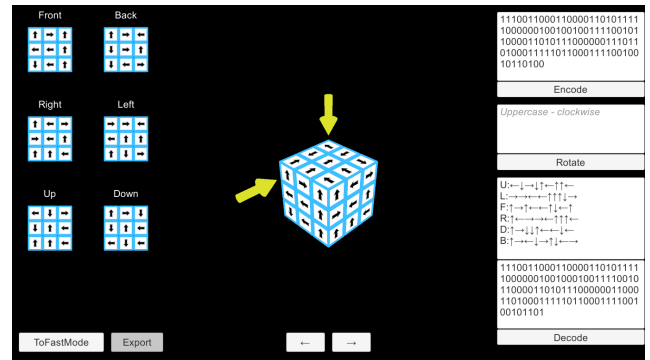
ing an arrow into a 2-bit string, i.e., $\alpha^{-1}(\uparrow) = 00, \alpha^{-1}(\rightarrow) = 01, \alpha^{-1}(\downarrow) = 10$ and $\alpha^{-1}(\leftarrow) = 11$.

For further illustrations, the message

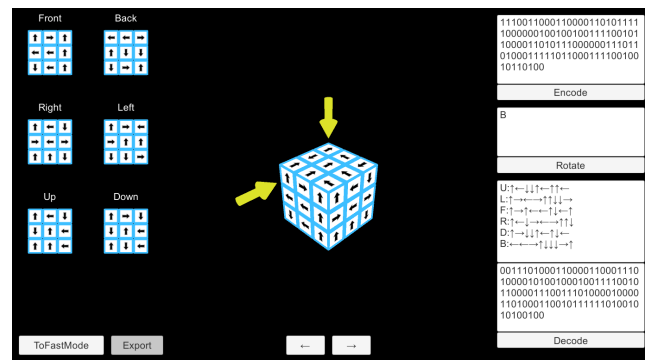
$$m = 111001100011000011010111110000001001001001111001011000011010111000000111011010001111101100011110010010110100 \quad (2)$$

can be assigned as the arrows depicted in Figure 2. (a), while after rotating face B, the original configuration of Figure 2. (a) becomes a new configuration depicted in Figure 2. (b), which corresponds to the following message:

$$m' = 0011101000110000110001110100001010010001001111001011000011100111010000100011010001111101001010100100. \quad (3)$$



(a) Arrow filling over original configuration



(b) Arrow updating after rotating face B

FIGURE 4. Encoding/decoding with arrows.

Remark 1: To further enhance the density of message embedding, the following encoding patterns might be useful:

- Assign more directions, for example, 360° , to an arrow.
- Arrange more arrows on a facet. For example, take a facet as the face of a watch with three hands – the second hand, minute hand and hour hand. Then, a combination of three hands at different angles is used to express a message.

²Padding necessary 0 in the left if not.

³This means not rotating the Rubik's cube to the original configuration.

B. ENCRYPTION/DECRYPTION ALGORITHMS

Our secret key encryption scheme, denoted \mathfrak{S}_1 , consists of the following three steps:

- **Setup**. Over a $3 \times 3 \times 3$ Rubik's cube, the message space and the ciphertext space are set as

$$\mathcal{M} = \{0, 1\}^{108}, \text{ and } \mathcal{C} = \mathcal{M} \times \mathfrak{R} \quad (4)$$

respectively.

- **KeyGen**. A random rotating sequence $k \in \mathfrak{R}$ with the *proper*⁴ word length can be used as a secret key.
- **Encrypt**. Upon input of a secret key $k \in \mathfrak{R}$ and a 108-bit message m , perform the following steps:
 - Choose a random rotation sequence $r \in \mathfrak{R}$;
 - Encode the message m to the 54 facets of the Rubik's cube;
 - Perform rotation k' (i.e. the reverse rotation of k);
 - Perform rotation r ;
 - Perform rotation k ;
 - Decode the arrows on the 54 facets of the Rubik's cube to a 108-bit string m' ;
 - Output $c = (m', r)$.
- **Decrypt**. Upon input of a secret key $k \in \mathfrak{R}$ and a ciphertext $c = (m', r)$, perform the following steps:
 - Check whether m' is a 108-bit string: If not, return \perp , which indicates that c is an invalid ciphertext; otherwise, continue;
 - Check whether r is a valid rotating sequence: If not, return \perp ; otherwise, continue;
 - Encode m' to the 54 facets of the Rubik's cube;
 - Perform rotation k' ;
 - Perform rotation r' ;
 - Perform rotation k ;
 - Decode the arrows on the 54 facets of the Rubik's cube to a 108-bit message m ;
 - Output m .

Remark 2: Note that scheme \mathfrak{S}_1 does not work in the following two cases:

- k or r lies in the center of \mathfrak{R} . However, considering that the center of \mathfrak{R} is negligibly small compared to the order of the Rubik's group, we do not need to worry about these cases. In fact, the center of \mathfrak{R} consists of only 2 elements [9](cf. Page 99).
- k and r commutes, i.e., $kr = rk$. On the one hand, this can be easily tested for during the encryption process and avoided by choosing another properly random rotation sequence r . On the other hand, if both k and r are chosen to be sufficiently random and sufficiently long, this case occurs only with a negligible probability.

Theorem 1 (Correctness.): If the above encryption scheme \mathfrak{S}_1 is correct, i.e., for a given secret key $k \in \mathfrak{R}$ and any

message $m \in \mathcal{M}$, we have

$$\text{Decrypt}(k, \text{Encrypt}(k, m)) = m.$$

Proof: To see the correctness, it is enough to notice that after the encryption process, the ciphertext $c = (m', r)$ consists of a random rotation r and a transformed message m' that is encoded in the configuration $k' \cdot r \cdot k$. Thus, after the decryption process, we obtain the confirmation

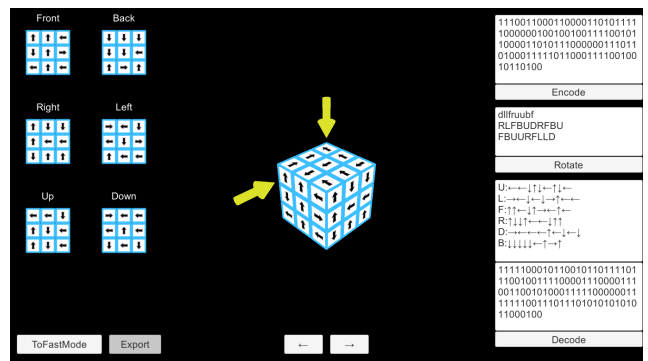
$$(k' \cdot r \cdot k) \cdot (k' \cdot r \cdot k) = 1$$

which is just the original configuration for encoding message m during the encryption process. \square

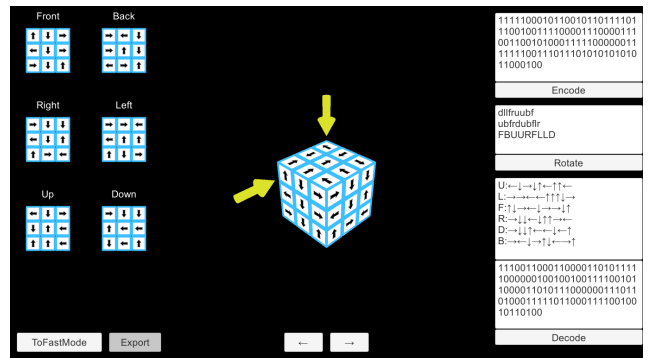
Example 1: Supposing that the secret key is $k = \text{FBUURFLLD}$ and $r = \text{RLFBUDRFBU}$, the message given by (2) is encrypted as

$$c = (1111110010111100000111101110010011111011011000101110000111001011100110110110001011100001110010111001101110011010101001000100, \text{RLFBUDRFBU}). \quad (5)$$

See Figure 5.(a). The decryption view is also given in Figure 5. (b).



(a) encrypting



(b) decrypting

FIGURE 5. Encryption and decryption views.

Remark 3: Note that although the message is always encoded in the original configuration, God's algorithm and other Rubik's solvers do not threaten the security of the above encryption scheme. In fact, without knowing the secret key k , we can view the encryption algorithm as a black box that

⁴Here, the adjective word *proper* has two aspects. First, the sequence should be long enough to resist guessing attacks. Second, we should take into consideration the so-called equivalent key problem. That is, two words in \mathfrak{R} with different word lengths might express the same value according to \mathfrak{R} 's generating relations.

takes as input a message $m \in \mathcal{M}$ and outputs a ciphertext $c = (m', r) \in \mathcal{M} \times \mathfrak{R}$, where r is selected at random and is thus totally independent of k . That is, the ciphertext does not output the final configuration $k' \cdot r \cdot k$. Thus, no one, except the decryption, knows which configuration is used for calling Rubik's solvers. A more rigorous security proof is presented in the following theorem.

Theorem 2 (IND-CPA): The above encryption scheme \mathfrak{S}_1 is indistinguishable against a chosen plaintext attack (IND-CPA), assuming that the CDP problem is intractable over the Rubik's group \mathfrak{R} . More specifically, if there is a probabilistic polynomial time adversary \mathcal{A} that can, within time t , break the IND-CPA security of this scheme with a non-negligible advantage ϵ , then there is a probabilistic polynomial time algorithm \mathcal{B} that can, within time t' , solve the CDP problem over \mathfrak{R} with a non-negligible advantage ϵ' such that $t' \approx t + t_{\text{enc}}$ and $\epsilon' = \epsilon$, where t_{enc} indicates the time for performing one encryption.

Proof: Suppose that \mathcal{B} 's CDP challenge instance is given by $(x, y) \in \mathfrak{R}$. Without loss of generality, assume that the word length of x is less than that of y . Then, upon receiving two equal-length challenge messages m_0 and m_1 from adversary \mathcal{A} , \mathcal{B} performs the following steps:

- Choose a random bit $b \in \{0, 1\}$;
- Encode the message m_b in the 54 facets of the Rubik's cube;
- Perform rotation y ;
- Decode the arrows on the 54 facets of the Rubik's cube to a 108-bit string m' ;
- Output the challenge ciphertext $c^* = (m', x)$.

Now, it is \mathcal{A} 's duty to output a bit b' , i.e., a guess for b , based on the decision of which of the two challenge messages m_0 and m_1 is concealed in c^* . Upon receiving b' from \mathcal{A} , \mathcal{B} judges whether $b' = b$: If so, \mathcal{B} outputs 1, which indicates that the given CDP challenge is an Yes -instance; otherwise, \mathcal{B} randomly outputs 1 or 0.

To proceed, let us determine \mathcal{B} 's advantages and running time for solving the CDP challenge. Apparently, only if (x, y) is a conjugate pair, i.e., $y = z'xz$ for some $z \in \mathfrak{R}$, c^* is a valid ciphertext of the message m_b . In this case, \mathcal{A} 's advantage in breaking the IND-CPA security of the scheme (i.e. guessing correctly $b' = b$) is equally transferred to \mathcal{B} 's advantage in making correct decisions regarding whether x and y are conjugated to each other. On the other hand, whenever \mathcal{A} 's guess is incorrect, \mathcal{B} makes a random decision by outputting 1 or 0 randomly. In this case, \mathcal{B} has no advantage in solving the given CDP challenge. This suggests that $\epsilon' = \epsilon$. In addition to calling \mathcal{A} , \mathcal{B} 's extra running time for the above reduction process is similar to performing one-time encryption, considering that the rotation y is similar to rotating z' , x and z one-by-one, for some possible and unknown z such that $y = z'xz$. Therefore, $t' \approx t + t_{\text{enc}}$. \square

Remark 4: Enlightened by the so-called FO technique [8], the security of the above encryption scheme can be further

improved. The enhanced scheme, denoted \mathfrak{S}_2 , consists of the following core points:

- Employ a cryptographic hash function $H : \mathcal{M} \times \mathcal{R} \rightarrow \mathcal{M}$ and redefine the ciphertext space as $\mathcal{C} = \mathcal{M}^2 \times \mathcal{R}$;
- **Encrypt2.** Upon input of a secret key k and a message m , perform the following steps:
 - Choose a random rotation sequence $r \in \mathfrak{R}$;
 - Compute $h = H(m, r)$;
 - $(m', r) \leftarrow \text{Encrypt}_k(m; r)$, i.e., encrypt m with secret key k and random rotation r ;
 - $(h', r) \leftarrow \text{Encrypt}_k(h; r)$, i.e., encrypt h with secret key k and random rotation r ;
 - Output $c = (m', h', r)$.
- **Decrypt2.** Upon input of a secret key k and a ciphertext $c = (m', h', r)$, perform the following steps:
 - $m \leftarrow \text{Decrypt}_k(m', r)$;
 - $h \leftarrow \text{Decrypt}_k(h', r)$;
 - Check whether $h = H(m, r)$ holds: If not, abort; otherwise, continue;
 - Output m .

- Now, the security of \mathfrak{S}_2 is given by the following theorem:

Theorem 3 (IND-CCA2): The above enhanced encryption scheme \mathfrak{S}_2 is indistinguishable against adaptively chosen ciphertext attack (IND-CCA2), assuming that the hash function H is a random oracle, and the CDP problem is intractable over the Rubik's group \mathfrak{R} .

Proof: This is a corollary of Theorem 12 of [8] and our Theorem 1 established above. \square

- Compared to the standard diagram given in [8], a slight variation in \mathfrak{S}_2 is to let the verification code $h = H(m, r)$ be encrypted as another ciphertext component h' instead of directly outputting h explicitly. This is crucial for the security of \mathfrak{S}_2 since otherwise an adversary \mathcal{A} in the IND-CPA or IND-CCA2 game can always output a correct decision by checking whether $h = H(m_b, r)$ holds, where m_b ($b = 0, 1$) is the challenge message chosen by \mathcal{A} itself.

IV. PERFORMANCE EVALUATION

Let us proceed to evaluate the performance of our proposal based on the asymptotic complexity, as well as the running time.

Asymptotically, the performance of the above schemes is determined mainly by the word length of the involved rotations. Without loss of generality, we assume that both the secret key rotation sequence and the random rotation sequence used for encryption should be sufficiently long to resist brute force attack. Now, suppose that we want to ensure λ -bit entropy in the involved random rotations by using random rotations that are as long as ℓ basic rotations.

Then, we have⁵

$$12^\ell \geq 2^\lambda. \tag{6}$$

Thus, it is enough to set $\ell \approx 0.28\lambda$. That is, a random rotation sequence that consists of 28 random basic rotations contains 100-bit entropy. Now, the asymptotical performance with respect to the system security parameter λ is summarized in Table 1. This suggests that our proposal has the following two remarkable merits:

- **Zero setup.** In the Setup algorithm, we just need to reach agreement on the definitions of the message space \mathcal{M} , the ciphertext space \mathcal{C} , and the involved hash function H for Scheme \mathfrak{S}_2 . That is, for real implementation, we do not need to perform any computations in this step, and the hash function H can be instantiated with any secure cryptographic hashes, such as SHA256 and SM3.
- **Linear encryption/decryption speed.** In practice, we can implement the basic rotations within the time complexity $\mathcal{O}(1)$ by using a precomputed table. Thus, it is easy to see that, for an 108-bit message, both the encryption and the decryption can be finished linearly with the word length of the involved random rotations. For long messages, we can divide them into several blocks and then encrypting/decrypting them one-by-one. Recall those cryptosystems with provable security reductions, such as RSA-based ones (with modulus n) and ECC-based ones (with modulus $q = p^m$), the best computational complexities of encryption/decryption algorithms that we can expect are $\mathcal{O}(\log^2 n \log \log n)$ and $\mathcal{O}(\log^2 q \log \log q)$, respectively.⁶ That is, they are quadratic or even higher with respect to the length of the modulus – the system security parameters. This comparison says that in a theoretical perspective, the proposed schemes are considerably fast.

TABLE 1. Asymptotic performance.

Algorithms	Schemes	Core operations	Complexity
Encode	$\mathfrak{S}_1, \mathfrak{S}_2$	108 bits \Rightarrow 54 arrows	$\mathcal{O}(1)$
Decode	$\mathfrak{S}_1, \mathfrak{S}_2$	54 arrows \Rightarrow 108 bits	$\mathcal{O}(1)$
Setup	$\mathfrak{S}_1, \mathfrak{S}_2$	define \mathcal{M}, \mathcal{C} and \mathcal{H}	0
KeyGen	$\mathfrak{S}_1, \mathfrak{S}_2$	pick ℓ random basic rotations	$\mathcal{O}(\lambda)$
Encrypt	\mathfrak{S}_1	3ℓ rotations	$\mathcal{O}(\lambda)$
	\mathfrak{S}_2	6ℓ rotations	$\mathcal{O}(\lambda)$
Decrypt	\mathfrak{S}_1	3ℓ rotations	$\mathcal{O}(\lambda)$
	\mathfrak{S}_2	6ℓ rotations	$\mathcal{O}(\lambda)$

In practical, to test the real running time of our proposal, we implement the basic rotations by using the following software/hardware environments:

- OS: Windows 10, Visual Studio 2017, Microsoft Visual C++ (Compiler: cl)
- CPU: Intel Core I&-6700K, 8 Cores, 4.0

⁵Here, the base is 12 instead of 6 since the random rotation sequence can be viewed as a sentence defined over an alphabet with size 12.

⁶Since we need to perform at least one time exponential operation modulo n or q .

Over 10^6 random tests, the average running time for each basic rotation is approximately 0.015 microseconds (or equivalently, 15 nanoseconds). Thus, with the suggested parameter settings, i.e., $\ell = 28$ for ensuring 100-bit entropy in involved random rotations, the main workload of encryption/decryption of our scheme \mathfrak{S}_1 (resp. \mathfrak{S}_2) can be finished within 1.26 (resp. 2.52) microseconds.

The comparisons between our proposal and other 19 typical cryptosystems — divided into 7 categories according to whether and what hardness assumptions are based on — are listed in Table 2. Note that since the message block sizes of different cryptosystems are always different, a fair comparison way is to use the amortized 1-bit encryption/decryption speed. That is, if the claimed encryption/decryption speed is x milliseconds for messages with block size L bits, then the amortized 1-bit encryption/decryption speed is $x \times 1000/L$ microseconds.¹⁰ In particular, for image encryption, we view the message block size as the total bit-length of the image. For instance, Lian *et al.*'s chaos-based image encryption scheme can encrypt the Lena image of size $3 \times 512 \times 512$ in 349 ms [2], then the amortized 1-bit encryption speed is $\frac{349 \times 1000}{3 \times 512 \times 512} \approx 0.444 \mu s$. From this table, we can see that the amortized 1-bit encryption/decryption speed of our proposal is considerably fast. More specifically, the performance advantages of our proposal can be summarized as follows:

- Nearly 3 (resp. 2) times faster than in encryption than the symmetric scheme DES/CRT (resp. AES/CBC-256), of which assembly language routines were used for speed optimization [4].
- Over 3 times faster in encryption than the scheme RSA-2048, which is even implemented with a small encryption exponential $e = 17$ [4] (and thus suffering from the so-called small exponential attack).
- Nearly 1000 times faster in encryption than the ECC type scheme ECIES over the finite field $GF(p)$ with a 256-bit prime p .
- Over 10 times faster in encryption than the lattice-based schemes such as NTRU-743 [3] and LAC-128 [13].
- Significantly faster in encryption than other chaos-based encryption schemes and Rubik-based schemes:
 - For chaos-based image encryption, since the decryption does not required to exactly recover every bit, instead of using provable security models such as CPA and CCA, we always discuss its security in a heuristic manner by using the metrics such as the number of pixels changing rate (NPCR), the unified average changing intensity (UACI) and so on, the encryption can be made even fast. Even so, our provable CPA-secure scheme \mathfrak{S}_1 is still 3, 10, and 30 times faster than the schemes Faraja (2013), Wong (2008) and Liao (2005), respectively.

¹⁰Informally, 1-bit amortized speed is anti-proportional to the so-called throughput: 1-bit amortized speed v (in microseconds) means nothing but the throughput $1/v$ Mbps.

TABLE 2. Comparisons on amortized 1-bit encryption/decryption speeds.

Hardness Assumptions	Schemes & Parameters	1-bit Amortized Speeds (μs)		Provable Security	Memo & References
		Encryption	Decryption		
-	DES/CRT	0.071	0.071	No, CCA	Crypto++ Benchmark [Dai09] ¹
	AES/OFB-128	0.101	0.101	No, CCA	
	AES/CBC-256	0.039	0.039	No, CCA	
IFP	RSA-2048 (with $e = 17$)	0.078	2.969	Insecure	
ECDLP	ECIES over GF(p) 256	22.070	15.547	Yes, CCA	
	ECIES over GF(2^n) 233	90.858	52.146	Yes, CCA	
Lattice RLWE	LAC-128	0.263	0.103	Yes, CPA	NIST PQC Round 2 [LLJ ⁺ 19] ²
	LAC-192	0.565	0.366	Yes, CPA	
	LAC-256	0.80	0.372	Yes, CPA	
	NTRU-443	0.318	0.413	Yes, CCA	NIST PQC Round 1 [CHWZ17] ³
	NTRU-743	0.207	0.334	Yes, CCA	
	NTRU-1024	88.158	151.316	Yes, CCA	
Chaos-based	Lian (2005)	0.444	0.455	No, $\left(\begin{matrix} \text{NPCR}\% = 99.59 \\ \text{UACI}\% = 33.42 \end{matrix} \right)$	Image encryption
	Wong (2008)	0.122	0.133	No, $\left(\begin{matrix} \text{NPCR}\% = 99.61 \\ \text{UACI}\% = 33.43 \end{matrix} \right)$	
	Faraja (2013)	0.031	0.032	No, $\left(\begin{matrix} \text{NPCR}\% = 99.61 \\ \text{UACI}\% = 33.46 \end{matrix} \right)$	Based on evaluation in [AFV14]
	Liao (2010)	3.017	1.406	Yes, CPA	Data encryption
	Li (2018)	7.353	4.267	Yes, CCA	Based on evaluation in [LWW ⁺ 18]
Rubik-based	Loukha (2012)	1.831	1.831	No, $\left(\begin{matrix} \text{NPCR}\% = 99.59 \\ \text{UACI}\% = 28.62 \end{matrix} \right)$	Based on evaluation in [LCB12]
	Dhanda (2018)	\sim DES	\sim DES	No, not claimed	Based on evaluation in [DPTS18]
	Scheme \mathfrak{S}_1	0.012	0.012	Yes, CPA	Our Proposal
	Scheme \mathfrak{S}_2	0.023	0.023	Yes, CCA	

¹ See <https://www.cryptopp.com/benchmarks.html>

² See <https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions>

³ See <https://csrc.nist.gov/Projects/post-quantum-cryptography/Round-1-Submissions>

- For chaos-based data encryption, our provable CCA-secure scheme \mathfrak{S}_2 is nearly 150 and 300 times faster than the schemes Liao (2010) and Li (2018), respectively.
- For Rubik-based image encryption, our scheme \mathfrak{S}_1 is nearly 150 times faster than the scheme Loukha (2012).
- For Rubik-based data encryption, our scheme \mathfrak{S}_2 is over 3 times faster than the scheme Dhanda (2018).

To further illustrate our idea even clearly, the processes of encoding/encryption and decryption/decoding are demonstrated graphically with Unity (Ver: 2019.3.2f1), a well-known real-time 3D development platform. The source codes for these tests and illustrations are available at <https://github.com/flowerlet/RubikEnc>.

V. CONCLUSIONS

It is interesting to couple the mainstream techniques of modern cryptography and the well-known Rubik's cube. Indeed, with proper message mapping methods, Rubik's rotations are not only permutations but also substitutions. Thus, any $n \times n \times n$ ($n \geq 3$) Rubik's cube can be viewed as a compact physical transformation device for modern cryptography and is suitable for the proposed method. The larger n is, the denser the message embedding achieved, although setting n to 3 is sufficiently secure and efficient. Most existing Rubik's cryptographic proposals lack of provable security reductions, while in this paper, we bridge this gap by laying the security of our proposals within falsifiable intractability assumptions: the conjugacy decision problem over Rubik's

groups. In addition, there are two remarkable merits in our proposal: zero setup and linear encryption/decryption speed.

ACKNOWLEDGEMENTS

The authors would like to thank Prof. Jun Shao for providing valuable discussion.

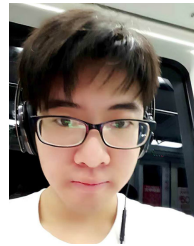
REFERENCES

- [1] I. Anshel, M. Anshel, B. Fisher, and D. Goldfeld, "New key agreement protocols in braid group cryptography," in *Proc. Cryptographers Track RSA Conf.*, in Lecture Notes in Computer Science. Berlin, Germany: Springer, 2001, pp. 13–27.
- [2] S. El Assad, M. Farajallah, and C. Vlădeanu, "Chaos-based block ciphers: An overview," in *Proc. 10th Int. Conf. Commun. (COMM)*, May 2014, pp. 1–4.
- [3] C. Chen, J. Hoffstein, W. Whyte, and Z. Zhang, "NTRUEncrypt: A lattice based encryption algorithm," NIST, Gaithersburg, MD, USA, Tech. Rep., 2017. [Online]. Available: <https://csrc.nist.gov/Projects/post-quantum-cryptography/Round-1-Submissions>
- [4] W. Dai. (2009). *Crypto++ 5.6.0 Benchmarks*. [Online]. Available: <https://www.cryptopp.com/benchmarks.html>
- [5] R. Dhandabani, S. S. Periyasamy, P. Theagarajan, and A. K. Sangaiah, "Six-face cubical key encryption and decryption based on product cipher using hybridisation and Rubik's cubes," *IET Netw.*, vol. 7, no. 5, pp. 313–320, Sep. 2018.
- [6] Y. Ding, H. Tian, and Y. Wang, "An improved signature scheme based on the braid group," *J. Xidian Univ.*, vol. 33, no. 1, pp. 50–61, 2006.
- [7] E. A. Elrifai and H. R. Morton, "Algorithms for positive braids," *Quart. J. Math.*, vol. 45, no. 4, pp. 479–497, 1994.
- [8] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 1666. Berlin, Germany: Springer, 1999, pp. 537–554.
- [9] D. Joyner, *Adventures Group Theory: Rubik's Cube, Merlin's Machine*, 2nd ed. Baltimore, MD, USA: Johns Hopkins Univ. Press, 2008.
- [10] K. H. Ko, S. J. Lee, J. H. Cheon, and J. W. Han, "New public-key cryptosystem using braid groups," in *Advances in Cryptology—Crypto*. Berlin, Germany: Springer, Aug. 2000, pp. 166–183.
- [11] K. Ko, J. Lee, and T. Thomas, "Towards generating secure keys for braid," *Cryptogr., Des., Codes Cryptogr.*, vol. 45, no. 3, pp. 317–333, 2008.

- [12] K. Loukhaoukha, J.-Y. Chouinard, and A. Berdai, "A secure image encryption algorithm based on Rubik's cube principle," *J. Electr. Comput. Eng.*, vol. 2012, pp. 1–13, Oct. 2012.
- [13] X. Lu, Y. Liu, D. Jia, H. Xue, J. He, Z. Zhang, Z. Liu, H. Yang, B. Li, and K. Wang, "LAC: Lattice-based cryptosystems," NIST, Gaithersburg, MD, USA, Tech. Rep., 2019. [Online]. Available: <https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions>
- [14] J. Li, L. Wang, L. Wang, X. Wang, Z. Huang, and J. Li, "Verifiable chebyshev maps-based chaotic encryption schemes with outsourcing computations in the cloud/fog scenarios," *Concurrency Comput., Pract. Exper.*, vol. 31, no. 22, Nov. 2019.
- [15] C. F. Miller, "Decision problems for groups—Survey and reflections," in *Algorithms and Classification in Combinatorial Group Theory*. New York, NY, USA: Springer, 1992.
- [16] S. C. Naik and P. N. Mahalle, "Rubik's cube based private key management in wireless networks," in *Proc. 15th Int. Conf. Adv. Comput. Technol. (ICACT)*, Sep. 2013, pp. 1–6.
- [17] C. Petit and J. Quisquater, "Cayley hash functions," in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds., 2nd ed. Boston, MA, USA: Springer, 2011, pp. 183–184.
- [18] C. Petit and J. Quisquater, "Rubik's for cryptographers," *Notice AMS*, vol. 60, no. 6, pp. 733–739, 2013.
- [19] T. Rokicki, H. Kociemba, M. Davidson, and J. Dethridge, "The diameter of the Rubik's cube group is twenty," *SIAM J. Discrete Math.*, vol. 27, no. 2, pp. 1082–1105, Jan. 2013.
- [20] A. Seress, *Permutation Group Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2002.
- [21] E. Volte, J. Patarin, and V. Nachev, "Zero knowledge with Rubik's cubes and non-abelian groups," in *Cryptography and Network Security (Lecture Notes in Computer Science)*, vol. 8257, M. Abdalla, C. Nita-Rotaru, and R. Dahab, Eds. Paraty, Brazil: Springer, 2013, pp. 74–91.
- [22] L. Wang, Z. Cao, S. Zheng, X. Huang, and Y. Yang, "Transitive signatures from braid groups," in *Proc. IndoCrypt*, Dec. 2007, pp. 183–196.
- [23] L. Wang, Z. Cao, P. Zeng, and X. Li, "One-more matching conjugating problem and security of braid-based signatures," in *Proc. AsiaCCS*, Mar. 2007, pp. 295–301.
- [24] L. Wang, Y. Tian, Y. Pan, and Y. Yang, "New construction of blind signatures from braid groups," *IEEE Access*, vol. 7, pp. 36549–36557, 2019.
- [25] L. Wang, L. Wang, Z. Cao, Y. Yang, and X. Niu, "Conjugate adjoining problem in braid groups and new design of braid-based signatures," *Sci. China Inf. Sci.*, vol. 53, no. 3, pp. 524–536, Mar. 2010.
- [26] E. Yen and L.-H. Lin, "Rubik's cube watermark technology for grayscale images," *Expert Syst. Appl.*, vol. 37, no. 6, pp. 4033–4039, Jun. 2010.



YUN PAN received the B.S. degree from Northwest Normal University, in 1995, the M.S. degree from Liaoning Shihua University, in 2001, and the Ph.D. degree from the China University of Mining and Technology, Beijing, in 2003, all in engineering. She is currently a Professor with the Communication University of China. Her current research interests include network security, blockchain, and the future Internet architecture.



ZHEN WANG received the B.S. degree in engineering from Beijing Information Science and Technology University, in 2015. He is currently pursuing the master's degree with the Beijing University of Posts and Telecommunications. His current research interests include blockchain, cryptography, network communication, and computer vision.



PING PAN received the B.S. degree in mathematics from Northwest Normal University, in 2003, the M.S. degree in mathematics from the Communication University of China, in 2009, and the Ph.D. degree in engineering from the Beijing University of Posts and Telecommunications, in 2013. She is currently an Assistant Professor with the Shaanxi University of Technology. Her current research interests include information security and blockchain.



LICHENG WANG (Member, IEEE) received the B.S. degree from Northwest Normal University, in 1995, the M.S. degree from Nanjing University, in 2001, and the Ph.D. degree from Shanghai Jiao Tong University, in 2007. He is currently a Professor with the Beijing University of Posts and Telecommunications. His current research interests include modern cryptography, blockchain, and quantum computation.

...