




Article

# New Commitment Schemes Based on Conjugacy Problems over Rubik's Groups

Ping Pan <sup>1</sup>, Junzhi Ye <sup>2</sup>, Yun Pan <sup>3,\*</sup>, Lize Gu <sup>4</sup> and Licheng Wang <sup>4</sup>

<sup>1</sup> School of Mathematics and Computer Science, Shaanxi University of Technology (SNUT), 1 East Ring Road, Hanzhong 723000, China; panping@snut.edu.cn

<sup>2</sup> College of Natural Resources and Environment, Northwest A&F University (NWAUFU), No. 3 Taicheng Road, Yangling, Xianyang 712100, China; abeautyleaf@126.com

<sup>3</sup> State Key Laboratory of Media Convergence and Communication, Communication University of China (CUC), 1 Dingfuzhuang East Street, Beijing 100024, China

<sup>4</sup> State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (BUPT), 10 West Tucheng Road, Beijing 100876, China; glzisc@bupt.edu.cn (L.G.); wanglc@bupt.edu.cn (L.W.)

\* Correspondence: pany@cuc.edu.cn

**Abstract:** Commitment schemes are important tools in cryptography and used as building blocks in many cryptographic protocols. We propose two commitment schemes by using Rubik's groups. Our proposals do not lay the security on the taken-for-granted hardness of the word problem over Rubik's groups. Instead, our first proposal is based on a symmetric encryption algorithm that is secure based on the hardness of the conjugacy search problem over Rubik's groups, while our second proposal is based on the hardness of a newly derived problem—the functional towering conjugacy search problem over Rubik's groups. The former is proved secure in the sense of both computational hiding and binding, while the latter is proved even secure in the sense of perfect hiding and computational binding. Furthermore, the proposed schemes have a remarkable performance advantage: a linear commitment/opening speed. We also evaluate the efficiency of the commitment schemes and show that they are considerably fast.

**Keywords:** Rubik's group; conjugator search problems; commitment



**Citation:** Pan, P.; Ye, J.; Pan, Y.; Gu, L.; Wang, L. New Commitment Schemes Based on Conjugacy Problems over Rubik's Groups. *Information* **2021**, *12*, 294. <https://doi.org/10.3390/info12080294>

Academic Editor: Murilo da Silva Baptista

Received: 31 May 2021  
Accepted: 21 July 2021  
Published: 24 July 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 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

The commitment scheme is one of the most important cryptographic primitives and has been widely used as an essential building block in many cryptographic protocols, such as the verifiable secret sharing scheme [1] and zero-knowledge protocols [2]. Meanwhile, the commitment scheme itself can be used in many privacy-preserving scenarios such as e-voting [3], e-auction [4], GIS-supported location services [5], etc. Over the last two decades, many commitment schemes have been developed. To the best of our knowledge, the security of almost all explicit constructions for commitment schemes is based on number-theoretic hardness assumptions, such as the IFP-based scheme due to Goldreich [6] and the DLP-based scheme due to Pedersen [1], which imposes heavy computational burdens. Recently, the Rubik's cube, as a mechanical puzzle game tool, has been used by researchers to construct many cryptographic schemes, such as Cayley hash functions [7], key agreement protocols [8], and encryption schemes [9]. However, some of them establish their security on a taken-for-granted hardness assumption: the recovery of a Rubik's cube with a random configuration. Today, computer programs can solve Rubik's cubes instantaneously, and even human champions can solve them within 20 steps under a  $3 \times 3 \times 3$  configuration. Therefore, designing new Rubik's cryptographic protocols by using falsifiable intractability assumptions is of interest.

This paper proposes two new fast commitment schemes that use Rubik's cube rotation operations. To the best of our knowledge, one of them is the first commitment scheme

based on a symmetric encryption algorithm from a non-Abelian group. The other scheme can be regarded as a non-Abelian variant of the Pedersen commitment scheme, the security of which depends on the intractability assumption of the FT-CSP problem over the Rubik's group. Furthermore, we evaluate the efficiency of the schemes and show that they are more efficient than the Pedersen commitment scheme in terms of the computational cost.

The rest of the paper is organized as follows: in Section 2, we introduce necessary notations and preliminaries, including the relevant concepts of commitment scheme, Rubik's group, and intractability assumptions. In Section 3, we briefly review two building blocks: the encoding/decoding methods and the encryption/decryption algorithms. Two commitment schemes are presented in Section 4. Performance evaluations are provided in Section 5, and concluding remarks are given in Section 6.

## 2. Background

### 2.1. Notations

We adopt the following notations from [10]. The notation  $y = A(x; r)$  indicates an algorithm  $A$  on input  $x$ , and randomness  $r$  outputs  $y$ . We use  $y \leftarrow A(x)$  to represent the process of picking randomness  $r$  at random and setting  $y = A(x; r)$  and write  $y \leftarrow S$  for sampling  $y$  uniformly at random from the set  $S$ . Without loss of generality, all algorithms in this paper take as input a security parameter  $\lambda$  (that is usually written in unary as  $1^\lambda$ ), and sometimes we do not explicitly write  $\lambda$ , but we always need to assume that it is implicitly available, and the larger the  $\lambda$ , the more secure the cryptographic protocols. Finally, given two functions  $f, g : N \rightarrow [0, 1]$ , the notation  $f(\lambda) \approx g(\lambda)$  indicates that  $|f(\lambda) - g(\lambda)| = \mathcal{O}(\lambda^{-c})$  holds for every constant  $c > 0$ , and we say  $f$  is negligible (resp. overwhelming) if  $f(\lambda) \approx 0$  (resp.  $f(\lambda) \approx 1$ ).

### 2.2. Commitment Scheme

A commitment scheme is a two-party scheme between a committer and a receiver and runs in two phases: In the commitment phase, the committer computes a confidential commitment value  $c \leftarrow \text{Com}(s; r)$  and sends  $c$  to the receiver, where  $s$  is the corresponding plaintext commitment, and  $r$  is a random salt used for ensuring the freshness of the commitment; In the opening phase, the committer reveals  $(s; r)$ , and the receiver verifies and accepts the commitment only if  $c = \text{Com}(s; r)$  holds. The commitment scheme is said to be *consistent* if for each run of the above two-phase commitment protocol, the committed value  $s$  is accepted (with overwhelming probability), assuming that both the committer and the receiver are honest. The commitment protocol requires the following two security properties:

- *Hiding (against an adversarial receiver)*. The receiver cannot learn any non-trivial information of  $s$  from  $c$  before the opening phrase.
- *Binding (against an adversarial committer)*. The committer cannot open another commitment value  $s_1 \neq s$  without being detected by the receiver, or equivalently,  $s$  is uniquely bound to  $c$ .

Furthermore, a commitment protocol is said to be *perfect hiding* (resp. *perfect binding*) if no adversarial receiver (resp. committer) can break the hiding (resp. binding) property, while it is said to be *computational hiding* (resp. *computational binding*) if no probabilistic polynomial-time (PPT for short) adversarial receiver (resp. committer) can break the hiding (resp. binding) property with non-negligible probability.

### 2.3. Rubik's Group and the Intractability Assumptions

Let us take a  $3 \times 3 \times 3$  Rubik's cube as an example. The Rubik's cube surface is divided into 54 small facets, numbered from 1 to 54, located on its six faces. The six sides of the cube are called U, L, F, R, D, and B, representing the upper face, left face, front face, right face, down face, and back face, respectively. Each side of a Rubik's cube can be rotated as a whole. A  $90^\circ$  rotation is called a basic operation, denoted in as  $\{U, L, F, R, D, B\}$ . A  $90^\circ$  counterclockwise rotation is basically an inverse operation,

denoted in as  $\{U', L', F', R', D', B'\}$ . The composition of a series of base operations and base inverse operations constitutes a configuration. (See Figure 1).

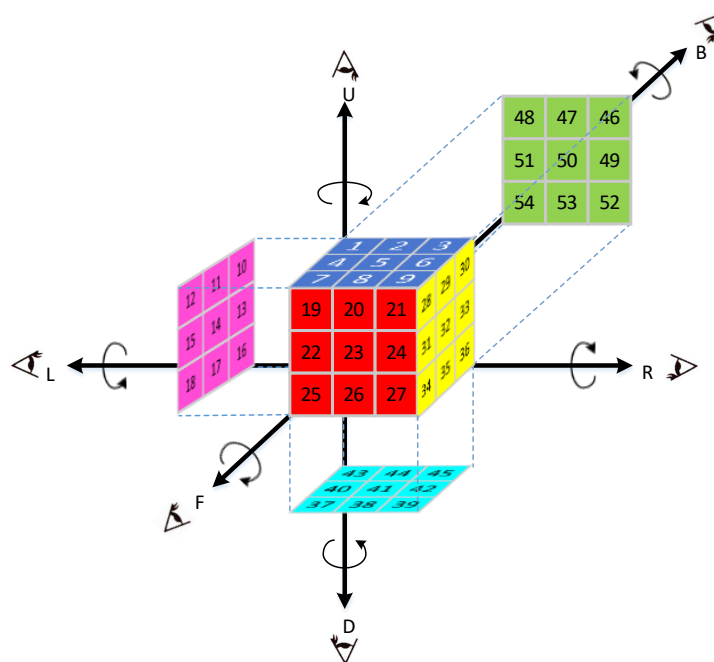


Figure 1. Facet numbers on original configuration [9].

All the possible configurations of a Rubik’s cube comprise a group, denoted by  $\mathfrak{R}$ , with identity 1 that indicates an empty rotation or, equivalently, doing nothing. That is, one finite-generated representation of  $\mathfrak{R}$  is given by

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

According to [9], we know that the cardinality of this group is approximated by  $2^{27} \cdot 3^{14} \cdot 5^3 \cdot 7^2 \cdot 11 \approx 4.3 \times 10^{19}$ . Apparently,  $\mathfrak{R}$  is a non-Abelian group, and thus, the following problems over  $\mathfrak{R}$  are non-trivial.

**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, denoted by  $x \sim y$ , i.e.,  $\exists z \in G$  such that  $x = y^z \triangleq z^{-1}yz$ .*

**Definition 2** (Conjugator Search Problem, CSP). *Given a group  $G$  and two elements  $x, y \in G$  with  $x \sim y$ , find  $z \in G$  such that  $x = y^z$ .*

Now, in this paper, we would like to propose the following problem:

**Definition 3** (Functional Towering Conjugacy Search Problem, FT-CSP). *Given a group  $G$ , a function  $v : G \rightarrow G$ , and three elements  $x, y, z \in G$  with  $y \sim z$ , find  $u \in G$  such that  $z = (y^{v(u)})^{x^u}$ , assuming such  $u \in G$  exists.*

**Remark 1** (Hardness of CDP and CSP). *Note that for any Abelian group  $G$ , all the above problems are trivial: we can answer all CDP instances with YES since every pair  $(x, y) \in G^2$  is conjugated and answers all CSP instances with arbitrary  $z \in G$ , while any FT-CSP instance  $(x, y, z) \in G^3$  with  $z = xy$  admits solutions for arbitrary  $u \in G$ . However, for non-Abelian groups, the above problems are non-trivial. In fact, in the generic group model, the CDP problem is unsolvable [11]. On the one hand, we know that for the permutation group, the CDP problem has*

no corresponding polynomial time method [12]. On the other hand, over the last few years, several cryptographic schemes based on the intractability assumption of the CSP problem and CDP problem over braid groups have been proposed [13,14]. 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  $\mathcal{B}_{48}$ , this progress gives us the confidence to establish the security of our new encryption scheme on the intractability assumptions of the conjugacy problems, including the CDP problem and CSP problem, over the Rubik's groups.

**Remark 2** (Hardness of FT-CSP). As for the FT-CSP problem, we have the following consideration:

- Suppose the conjugator, denoted by  $c$ , of the pair  $(z, y) \in G^2$  is unique. Then, we have  $cv(u) = x^u$ . Now, one possible way that we can conceive is to pick  $u \in G$  at random and check whether  $v(u) = c^{-1}x^u$  holds.
- Suppose the conjugators of the pair  $(z, y) \in G^2$  are not unique. Then, for each conjugator  $c_j$ , we face a similar group equation  $c_jv(u) = x^u$ .

That is, in both the above cases, solving an FT-CSP instance would be observably harder than solving a CSP instance.

Moreover, the hardness of FT-CSP might be tightly related to the choices of function  $v$ . Say, the simplest case is to set  $v(u) = 1$  for  $\forall u \in G$ , then the FT-CSP problem is nothing but the so-called double conjugator search problem (DCSP) that aims to find  $u \in G$  so that  $z = u^{-1}xuyu^{-1}xu$  for given  $(x, y, z) \in G^3$ . On the contrary, a much more complex case might be to set  $v : G \rightarrow G$  as an unpredictable function, say a random oracle that in practice could be instantiated by a collision-resistant hash (CRH for short) function or a pseudo-random function (PRF for short). Then, for solving the FT-CSP problem, we face two obstacles simultaneously: solving the CSP problem, and guessing the output of  $v(u)$  in advance. Thus, in this case, the unpredictability of the function  $v$  suggests that the success probability for solving the given FT-CSP instance is negligible.

Now, let us consider another trick, called after sampling, for defeating an FT-CSP solver  $\mathcal{S}$ : if the function  $v : G \rightarrow G$  is undetermined, say  $v$  is sampled from a CHF or PRF family but kept unrevealed to  $\mathcal{S}$  in advance. Then, after  $\mathcal{S}$  outputs their answer  $u \in G$  for a partially given FT-CSP instance  $(G, *, x, y, z)$ , let us sample  $v$  at random and then check whether  $u$  is a correct solution towards the fully given FT-CSP instance  $(G, v, x, y, z)$ . Clearly, before specifying  $v$ , it is totally undetermined whether  $\mathcal{S}$ 's answer is correct. Therefore, we conclude as follows:

**Claim 1.** The FT-CSP problem, with the after-sampling trick on  $v$ , is intractable even for a solver with unbounded computational power.

This enhanced hardness would play the underlying security basis of our second proposal given later in this paper.

**Remark 3** (Rubik's group vs. Braid group). As aforementioned, the Rubik's group  $\mathfrak{R}$  can be viewed as a special subgroup of the braid group  $\mathcal{B}_{48}$ . However, in this work, we use the term of rotating Rubik's cubes, instead of weaving braids, in describing our proposals, considering the following advantages of doing so:

- Even secure encoding method. For cryptographic applications, the involved group elements should be represented in an unambiguous way, i.e., the so-called canonical form that can be viewed as an encoding method on group elements. For braid-based cryptographic applications, typical canonical forms reveal partial information of the word length of the involved braids, suffering from to the so-called length-based attacks [15]. As for the Rubik's cube given in Figure 1, no matter how many rotations are done, its canonical form is always a permutation in  $\mathcal{S}_{48}$ . Thus, each element in a Rubik's group admits a fixed-length canonical form, and this property makes the length-based attacks useless.
- Even fast implementation. The typical implementation of Rubik-based cryptosystems can be finished approximately in microseconds [9], while the reported braid-based cryptosystems require milliseconds [16].

### 3. Reviewing of Building Blocks: Encoding and Encryption Using Rubik’s Cubes

To proceed, let us review the encoding/decoding methods and encryption/decryption algorithms given in [9] (See Figure 2). They are the building blocks of our proposal given in the next section.

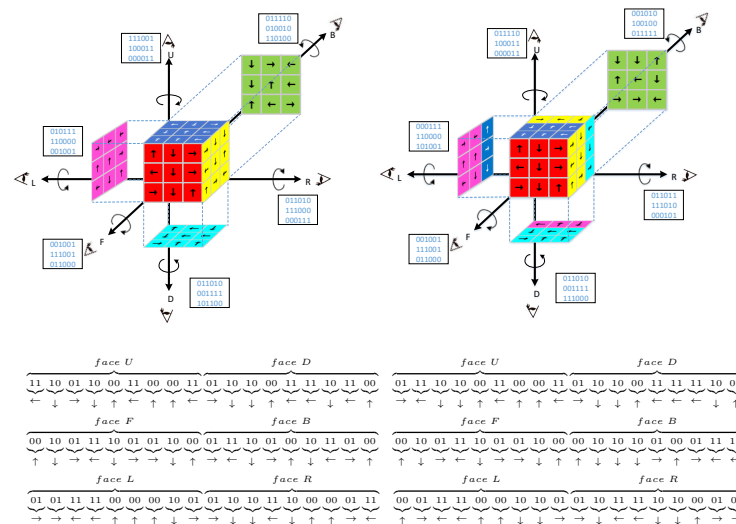


Figure 2. Left: Arrows on original configuration; Right: Arrows updating with rotating face B.

#### 3.1. Encoding/Decoding over Rubik’s Cubes

To commit messages by using the Rubik’s cube, we need to introduce a method for encoding/decoding messages on a Rubik’s cube, which is different from the traditional method of describing message letters directly on the Rubik’s cube facets. The key idea of the encoding/decoding methods is to map two bits to four arrows with different directions as follows [9]:

- Encode. Assume that each message is a 108-bit string and can be divided into 54 pairs. Then, each pair of bits can be translated to one of four arrows. For example, let 00, 01, 10, and 11 be translated to  $\uparrow$ ,  $\rightarrow$ ,  $\downarrow$  and  $\leftarrow$ , respectively. Next, the translated 54 arrows are assigned to the 54 facets one by one. Finally, the 54 facets are assigned to the six faces of Rubik’s cube as if they were the original configuration.
- Decode. The reverse process of encoding: given a configuration with 54 facets assigned with arrows, at first, each arrow is translated back to a 2-bit pair accordingly. Furthermore, then, output a 108-bit string by piecing together all these 2-bit pairs in the 54 facets one by one.

More formal descriptions on the encoding/decoding methods can be found in [9].

#### 3.2. Encryption/Decryption over Rubik’s Cubes

The first encryption scheme in [9] is used as a building block in our first construction of the commitment protocol. The scheme consists of the following four algorithms:

- Setup. Over a  $3 \times 3 \times 3$  Rubik’s cube, let  $\mathcal{M} = \{0, 1\}^{108}$  and  $\mathcal{C} = \mathcal{M} \times \mathfrak{R}$  be the space of messages and ciphertext, respectively.
- KeyGen. Randomly generate a secret key  $k \in \mathfrak{R}$  as a random rotating sequence with the proper word length.
- Encrypt. Input a secret key  $k$  and message  $m$ , then perform the following:
  - Choose a random rotation sequence  $r$ ;
  - 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 a ciphertext  $c = (m^*, r)$ .
- Decrypt. Input the secret key  $k$  and the ciphertext  $c = (m^*, r)$ , and then perform the following:
  - 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 a message  $m$ .

In [9], the above encryption scheme is proved to be indistinguishable against chosen plaintext attack (IND-CPA for short), assuming that the CDP problem is intractable over the Rubik’s group  $\mathfrak{R}$ .

#### 4. Our Proposals: The Commitment Schemes Using Rubik’s Cubes

Now, let us propose two commitment schemes, one of which is based on an enhanced version of the above symmetric encryption algorithm, and the other of which can be viewed as a non-Abelian analog of the Pedersen commitment scheme [1].

##### 4.1. Commitment Scheme Based on the CDP Problem

Over a  $3 \times 3 \times 3$  Rubik’s cube  $\mathfrak{R}$ , let  $\mathcal{M} = \{0, 1\}^{108}$  be the space of messages to commit to and a cryptographic hash function  $H : \mathcal{M} \times \mathfrak{R} \rightarrow \mathcal{M}$ . Then, the first commitment scheme, denoted by  $\mathcal{C}_1$ , involves the following two phases:

- Commitment phase. The committer commits to a message  $s \in \mathcal{M}$  as follows:
  - Choose a random rotation sequence  $r$ ;
  - Compute  $h = H(s, r)$ ;
  - Randomly choose a secret key  $k$ , i.e., a random rotating sequence with the proper word length;
  - $(s^*, r) \leftarrow \text{Encrypt}_k(s; r)$ , i.e., encrypt  $s$  with secret key  $k$  and random rotation  $r$ ;
  - $(h^*, r) \leftarrow \text{Encrypt}_k(h; r)$ , i.e., encrypt  $h$  with  $k$  and  $r$ ;
  - Send the commitment value  $c = (s^*, h^*)$  to the receiver.
- Opening phase. To open a commitment  $c = (s^*, h^*)$  to the receiver, the committer sends  $(k, r)$  to the receiver directly. Furthermore, upon receiving  $(k, r)$  sent by the committer, the receiver performs the following steps:
  - $s \leftarrow \text{Decrypt}_k(s^*, r)$ ;
  - $h \leftarrow \text{Decrypt}_k(h^*, r)$ ;
  - Check whether  $h = H(s, r)$  holds: if not, reject the commitment and return  $\perp$ ; otherwise, accept the commitment and return  $s$ .

CONSISTENCY. Note that the commitment process consists of two encryption algorithms; thus, the commitment value  $c = (s^*, h^*)$  is encoded in the two configurations  $(k' \cdot r \cdot k)_s$  and  $(k' \cdot r \cdot k)_h$ . The opening process consists of two decryption algorithms. Therefore, we obtain the confirmation as, respectively,

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

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

which are just the original configurations for encoding message  $s$  and hash value  $h$  during the commitment process.

SECURITY. Our proposal is based on an enhanced version of the aforementioned probabilistic encryption algorithm. Note that the commitment process is nothing but two encryption processes, the security of which are based on the intractable assumption of the CDP problem over the Rubik’s group, and that the commitment value  $c = (s^*, h^*)$  is two ciphertext components in  $(s^*, r)$  and  $(h^*, r)$  where another ciphertext component  $r$  is a public random. The receiver would like to learn any non-trivial information of  $s$  from  $c$  before the opening phrase; then, the receiver has to operate a decryption process. Therefore, to break the hiding property of the commitment is nothing but solving the CDP problem that is *kept unrevealed* to the receiver. Thus, no receiver can learn information from the commitment about the message  $s$ . This suggests that the above commitment scheme  $\mathcal{C}_1$  is *computational hiding*.

Now, let us consider the binding property. Suppose that an adversarial committer  $\mathcal{A}$  wants to open a commitment  $Com(s; r)$  with another value, say  $Com(s_1; r_1)$ , without being detected. That is,

$$Com(s, r) = c = Com(s_1, r_1) \quad \text{and} \quad (s, r) \neq (s_1, r_1).$$

Or equivalently,

$$\begin{aligned} (c_s, r) &\leftarrow \text{Encrypt}_k(s; r), \\ (c_h, r) &\leftarrow \text{Encrypt}_k(h; r), \end{aligned}$$

where  $c = (c_s, c_h)$ .

Additionally,

$$\begin{aligned} (c_s, r_1) &\leftarrow \text{Encrypt}_k(s_1; r_1), \\ (c_h, r_1) &\leftarrow \text{Encrypt}_k(h_1; r_1), \end{aligned}$$

where  $c = (c_s, c_h)$ .

According to Theorem 3 in [9], if the committer  $\mathcal{A}$  extracts a valid message  $s$  from the ciphertext  $c$  or the commitment  $c$ , then the committer can solve the CDP problem over the Rubik’s group that is *kept unrevealed* to the committer. Therefore, the commitment scheme  $\mathcal{C}_1$  is *computational binding*.

#### 4.2. Commitment Scheme Based on the FT-CSP Problem

Since the rotation sequence is composed of 12 basic operations  $\{U, L, F, R, D, B, U', L', F', R', D', B'\}$  and is well-known to be suitable for the transformation between binary and 12-adic numbers, a bit string message can be converted into a rotation sequence accordingly. Now, for a  $3 \times 3 \times 3$  Rubik’s cube  $\mathfrak{R}$ , let  $g, h \in \mathfrak{R}$  be two random public rotation sequences. Let  $\mathcal{M} = \{0, 1\}^{108}$  be the space of messages to be committed. Then, our second commitment scheme, denoted by  $\mathcal{C}_2$ , involves the following two phases:

- **Commitment phase.** The committer commits to a message  $s \in \mathcal{M}$  as follows:
  - (1) Encode the message  $s$  to the 54 facets of the Rubik’s cube;
  - (2) Convert  $s$  into a rotation sequence  $t$ ;
  - (3) Choose a random rotation sequence  $r$ ;
  - (4) Perform the rotation  $t'g'tr'hr't'gt$ ;
  - (5) Decode the arrows on the 54 facets of the Rubik’s cube to a 108-bit string  $c$ ;
  - (6) Send commitment value  $c$  to the receiver.
- **Opening phase.** To open a commitment  $c \in \mathcal{M}$  to the receiver, the committer sends  $(s, r)$  to the receiver directly. Furthermore, upon receiving  $(s, r)$  sent by the committer, the receiver performs the following steps:
  - (1)~(4) Same as (1)~(4) in the commitment phase;
  - (5) Decode the arrows on the 54 facets of the Rubik’s cube to a 108-bit string  $c^*$ ;
  - (6) Check whether  $c = c^*$  holds: if not, reject the commitment and return  $\perp$ ; otherwise, accept the commitment and return  $s$ .

**CONSISTENCY.** The consistency of the commitment protocol is straightforward, since the main body of the opening phase is nothing but a repeat of the corresponding steps of the commitment phase.

**SECURITY.** The key point is that the rotation sequence specified in step (4) is nothing but a functional towering conjugate action  $z = (h^r)^{g^t}$ . For mapping this action to an FT-CSP instance, we need to regard  $g, h, t$  as  $x, y, u$ , respectively, where  $r$ , picked at random, could be looked at as  $r = v(u)$ , i.e., a one-time output of a PRF function  $v$  that is independent of  $u$ . Therefore, breaking the hiding property of the commitment is nothing but solving the FT-CSP instance  $(\mathfrak{R}, v, g, h, z)$  where  $v : \mathfrak{R} \rightarrow \mathfrak{R}$  is a PRF function that is *kept unrevealed* to the receiver. This suggests that the above commitment scheme  $\mathcal{C}_2$  is *perfect hiding* according to **Claim 1**.

Now, let us consider the binding property. Suppose an adversarial committer  $\mathcal{A}$  wants to open a commitment  $Com(s; r)$  with another value, say  $Com(s_1; r_1)$ , without being detected. That is,

$$Com(s, r) = c = Com(s_1, r_1) \quad \text{and} \quad (s, r) \neq (s_1, r_1).$$

Or equivalently,

$$(h^r)^{g^t} = z = (h^{r_1})^{g^{t_1}} \quad \text{and} \quad (t, r) \neq (t_1, r_1)$$

where  $t_1$  should be a valid encoding of  $s_1$ . Unlike the adversarial receiver who faces an FT-CSP instance with the after-sampling trick on  $v$ , the committer  $\mathcal{A}$  has the freedom to choose  $t, t_1, r, r_1$ . To proceed, let us consider the following four cases:

- (1) Given  $t, r, t_1$ , finding  $r_1$  is to solve the CSP instance  $(\mathfrak{R}, x, y)$  with the setting  $x = ((h^r)^{g^t})^{g^{t_1}}$  and  $y = h$ .
- (2) Given  $t, r, r_1$ , finding  $t_1$  is to solve the FT-CSP instance  $(\mathfrak{R}, v, x, y, z)$  with the setting  $x = g, y = h, z = (h^r)^{g^t}$  and  $v(u) = r$  for  $\forall u \in \mathfrak{R}$ .
- (3) Given  $t_1, r_1, t$ , finding  $r$  is similar to case (1).
- (4) Given  $t_1, r_1, r$ , finding  $t$  is similar to case (2).

At this point, we can safely conclude that the commitment protocol  $\mathcal{C}_2$  is *computational binding*, assuming that the FT-CSP problem over the Rubik’s group  $\mathfrak{R}$  is intractable.

### 5. Performance Evaluation

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

The asymptotic performance with respect to the system security parameter  $\lambda$  is summarized in Table 1. Our commitment scheme  $\mathcal{C}_1$  is based on the symmetric encryption algorithms from the Rubik’s group, which have linear encryption/decryption speeds. Regarding the commitment scheme  $\mathcal{C}_2$ , we note that the commitment value is the result of performing a series of rotations  $t'g'tr'hrt'gt$ . Therefore, our proposals have a remarkable performance advantage: a linear commitment/opening speed.

According to [9], we find that 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.,  $\ell = 28$  to ensure 100-bit entropy in the involved random rotations, the main workload of commitment/opening of our commitment scheme  $\mathcal{C}_1$  (resp.  $\mathcal{C}_2$ ) can be completed within 2.52 (resp. 3.78) microseconds.

Recalling the Pedersen commitment scheme based on the discrete logarithm problem over the finite field  $\mathbb{F}_q$ , we find that the best computational complexities of the commitment/opening phases are  $\mathcal{O}(\log^2 q \log \log q)$ , where  $q$  is the length of the modulus. Therefore, our two commitment schemes are considerably fast.



**Table 1.** Asymptotic performance.

Algorithms	Schemes	Core Operations	Complexity
Encode	$\mathcal{C}_1, \mathcal{C}_2$	108 bits $\Rightarrow$ 54 arrows	$\mathcal{O}(1)$
Decode	$\mathcal{C}_1, \mathcal{C}_2$	54 arrows $\Rightarrow$ 108 bits	$\mathcal{O}(1)$
Conversion	$\mathcal{C}_1, \mathcal{C}_2$	pick $\ell$ random basic rotations	$\mathcal{O}(\lambda)$
Setup	$\mathcal{C}_1, \mathcal{C}_2$	define $\mathcal{M}$	0
	$\mathcal{C}_1, \mathcal{C}_2$	define $\mathcal{M}, g, h$	0
Commitment	$\mathcal{C}_1$	$6\ell$ rotations	$\mathcal{O}(\lambda)$
	$\mathcal{C}_2$	$9\ell$ rotations	$\mathcal{O}(\lambda)$
Verification	$\mathcal{C}_1$	$6\ell$ rotations	$\mathcal{O}(\lambda)$
	$\mathcal{C}_2$	$9\ell$ rotations	$\mathcal{O}(\lambda)$

## 6. Conclusions

We propose new commitment schemes that achieve secure computational/perfect hiding and computational binding assuming the difficulty of the CDP problem or the FT-CSP problem over Rubik's group, respectively. To the best of our knowledge, one of them is the first commitment scheme based on the symmetric encryption algorithm over a Rubik's group. The other is regarded as a non-Abelian variant of the Pedersen commitment scheme. Furthermore, we evaluate the efficiency of the schemes. Our proposals are highly efficient in terms of the computational cost and have a linear commitment/opening speed.

**Author Contributions:** Conceptualization, P.P. and J.Y.; methodology, P.P., Y.P., and L.G.; validation, P.P., L.G., and L.W.; formal analysis, P.P. and L.W.; writing—original draft preparation, P.P.; writing—review and editing, all; visualization, J.Y.; supervision and project administration, Y.P.; funding acquisition, P.P., L.G., and L.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (NSFC) (Grant NO. 61972050), the SNUT Doctoral Research Foundation (Grant No. SLGQD13-24), and the Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (SKLNST-2020-2-16).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** We thank the anonymous reviewers for giving us valuable suggestions.

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

## References

- Pedersen, T.P. Non-interactive and information theoretic secure verifiable secret sharing. In *Proceedings on Advances in Cryptology—CRYPTO, LNCS 576*; Springer: Berlin, Germany, 1992; pp. 129–140.
- Goldreich, O.; Krawczyk, H. On the composition of zero-knowledge proof systems. *SIAM J. Comput.* **1996**, *25*, 169–192. [[CrossRef](#)]
- Schoenmakers, B. A simple publicly verifiable secret sharing scheme and its application to electronic voting. In *Annual International Cryptology Conference*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 148–164.
- Dreier, J.; Dumas, J.G.; Jonker, H.; Lafourcade, P. Verifiability in e-Auction Protocols & Brandt's Protocol Revisited. In *Proceedings of the 1st Workshop on Hot Issues in Security Principles and Trust (HOTSPOT'13)*, Rome, Italy, 16 March 2013.
- Liu, L.; Kong, X.; Li, G.; Gao, L. Location of public service facilities based on GIS. In *Proceedings of the 19th International Conference on Geoinformatics*, Shanghai, China, 24–26 June 2011; pp. 1–4. [[CrossRef](#)]
- Goldwasser, S.; Micali, S.; Rivest, R.L. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM J. Comput.* **1988**, *17*, 281–308. [[CrossRef](#)]
- Petit, C.; Quisquater, J.-J. Rubik's for cryptographers. *Not. AMS* **2013**, *60*, 733–739. [[CrossRef](#)]

8. Naik, S.C.; Mahalle, P.N. Rubik's cube based private key management in wireless networks. In Proceedings of the 2013 15th International Conference on Advanced Computing Technologies (ICACT), Rajampet, India, 10–11 August 2013; pp. 1–6.
9. Pan, P.; Pan, Y.; Wang, Z.; Wang, L. Provably Secure Encryption Schemes With Zero Setup and Linear Speed by Using Rubik's Cubes. *IEEE Access* **2020**, *8*, 122251–122258. [[CrossRef](#)]
10. Chaidos, P.; Groth, J. Making Sigma-Protocols Non-interactive Without Random Oracles. In *Public Key Cryptography*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 650–670.
11. Miller, C.F., III. *Decision Problems for Groups—Survey and Reflections; Algorithms and Classification in Combinatorial Group Theory*; Springer: Berlin/Heidelberg, Germany, 1992.
12. Seress, A. *Permutation Group Algorithms*; Cambridge University Press: Cambridge, UK, 2002
13. Wang, L.; Wang, L.; Cao, Z.; Yang, Y.; Niu, X. Conjugate adjoining problem in braid groups and new design of braid-based signatures. *Sci. China Inf. Sci.* **2010**, *53*, 524–536. [[CrossRef](#)]
14. Wang, L.; Tian, Y.; Pan, Y.; Yang, Y. New construction of blind signatures from braid groups. *IEEE Access* **2019**, *7*, 36549–36557. [[CrossRef](#)]
15. Myasnikov, A.G.; Ushakov, A. Random subgroups and analysis of the length-based and quotient attacks. *J. Math. Cryptol.* **2008**, *1*, 29–61. [[CrossRef](#)]
16. Cha, J.C.; Ko, K.H.; Lee, S.; Han, J.W.; Cheon, J.H. An Efficient Implementation of Braid Groups. In *International Conference on the Theory and Application of Cryptology and Information Security*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 144–156.