On extensions of the one-time-pad

Bhupendra Singh[1], G. Athithan[1] and
Rajesh Pillai[2]

[1] Centre for Artificial Intelligence and Robotic
DRDO, Bangalore, INDIA
bhupendra@cair.drdo.in
ga.drdochair@cair.drdo.in
[2]Scientific Analysis Group, Delhi, INDIA
rpillai@sag.drdo.in

March 5, 2021

Abstract

The one-time-pad (OTP) is a classical yet the strongest cipher. Although the OTP offers perfect secrecy and is quantum-safe, it has cryptographic as well as operational weaknesses. Cryptographically its encryption is malleable. Operationally a key used more than once by mistake can lead to successful breaking of the OTP. Hence, there is a need for extensions of OTP to address these two weaknesses simultaneously keeping all the strength of OTP intact. To address this need, we propose two extensions of OTP. In the process we also prove a relation between block ciphers and Latin rectangles.

Keywords: Latin square, Malleable encryption, One-time-pad, Prefect secrecy, Quantum-safe

1 Introduction

In symmetric key cryptography we have two types of encryption algorithms [1]. The first type of algorithms are computationally secure in the presence of a bounded adversary [2]. Examples are stream ciphers [3] such as RC4 [4], Grain, Salsa, Trivium [5] and block ciphers such as DES, FEAL, IDEA, SAFER, RC5 [1] and AES [6]. The second type of algorithms are perfectly secure in the presence of an unbounded adversary. Examples are the one-time-pad (OTP) [2] and Latin square-based ciphers [7]. The OTP can also be seen as a particular case of the latter. The OTP is an encryption technique that is perfectly secure. That is, breaking the OTP by an unbounded adversary is not possible [2]. But, the OTP has a cryptographic weakness in being a malleable encryption method. The OTP also has an operational weakness with respect to accidental reuse of keys. **Accidental reuse of keys is an operational level problem in OTP, though the likelihood may be low. It can happen in a few scenarios that are known to experienced users of OTP.** Literature shows the OTP is very useful for government organizations [8] and many variants of the OTP are exist. In [9], a study of modified Vernam cipher using $2's$ complement operation for addition for encryption operation is reported. Random keys are generated using genetic algorithm in this cipher. The weak XOR operation of the OTP is replaced with more complex operation for strengthening the Vernam

cipher [10]. The hybridization of the two ciphers is also proposed [11]. The quantum variants of the OTP are also discussed in [12].

In this paper, we present two new extensions of the OTP that are based on non-malleable encryption and provide limited protection against the accidental reuse of the encryption keys. In the process of creating these extensions, we prove a relation between block ciphers and Latin rectangles. When the key size and the block size of a block cipher are equal it can be equivalent to a Latin square under certain conditions.

In Section 1, introduction and motivation for the work reported in this paper are given. In Section 2, usefulness of the OTP and its properties are discussed with the help of probability theory. This is followed by a brief argument that OTP is quantum-safe encryption. Next, some weaknesses of the OTP are highlighted. In Section 3, definitions of Latin rectangle and Latin square are given and these concepts are explained with the help of a few examples. Use of Latin square in encryption and decryption is discussed next with an example. This is followed by the representation of the OTP as a Latin square, and overall properties of Latin square-based ciphers are summarised. In this section we prove an important theorem that shows the relation between Latin rectangles and block ciphers. A particular case of the theorem on the relation between Latin squares and block ciphers is also discussed. In Section 4, two extensions of the OTP are presented. The first extension uses confusion and diffusion operators, while the second extension uses a block cipher in place of XOR function. Properties of both the extensions are highlighted in this section. Section 5 concludes the paper.

2 One-time-pad

The one-time-pad (OTP) is a simple encryption method, proposed originally in 1882 by Frank Miller [13, 14]. It is a perfectly secure encryption methods. It is a method where one XORs a plain message (M) with a key (K) [2]. When the key is random and not used more than once, the OTP is unbreakable. It is relatively easy to implement. Shannon has proved that this method provides perfect secrecy under certain conditions [15]. Any key once used should be discarded and never used again. It is easy to see why this condition is necessary to ensure that the method is unbreakable. Let us say that a single key K is used for encrypting two different plain messages M_1 and M_2 at two different times. That is, $C_1{=}M_1 \oplus K$ and $C_2{=}M_2 \oplus K$. If XOR of C_1 and C_2 is carried out, the common key K cancels itself giving rise to $M_1 \oplus M_2$. What is left would be just the result of XOR of the two plain messages. Using frequency statistics and a dictionary of the common language of the messages, an adversary would be able to factor the XOR and figure out both the messages in reasonable time frames today. As a key should be used only once, this method is called the *one-time-pad*. Incidentally the word *pad* is some historical baggage, coming from the manner of producing and distributing paper pads containing large amounts of key needed for using the OTP method. How the random keys were generated and distributed is another story.

The operational issue with OTP is to generate long random keys and to distribute them securely. However, there are many scenarios, where the secure delivery of the key is not an issue. Governments and large corporations can use their resources to send keys. The required length of these keys is often quoted as an issue. This issue is easily addressed by the large capacities of the hard disks today. For a meaningful rate of 10KB per second requirement for highly classified messages, a 4TB hard disk can keep keys for more than 10 years.

The OTP has been used for very critical communications in history, most notably [16] for the highest level of voice communications between the Allies in World War II. The Washington-Moscow hotline [8] that connected the top leaders of Russia and US used OTP. It is perhaps still used for keeping sensitive US government communications secret. For distribution of keys,

heavily armoured trucks protected by armed guards are used between the Pentagon and remote locations [17].

Now we discuss the perfect secrecy aspect of the OTP. An encryption method is called perfectly secure, when it cannot be broken by brute-force attempt drawing from unlimited computation time. In the context of unbreakability of ciphers, Shannon gave two properties namely confusion and diffusion that a good cryptographic system should have to hinder statistical analysis. The OTP does not have these two properties. Confusion and diffusion are necessary in computationally secure ciphers but not in perfectly secure ciphers [2]. Perfect secrecy of the OTP cannot be studied in computational complexity framework, because we allow computation time to be unlimited. The appropriate framework in which to study perfect secrecy is probability theory. We start with a few notations from [2].

A discrete random variable, say $\mathbf{X}$, is defined by a finite set X and a probability distribution on X. The probability that the random variable $\mathbf{X}$ takes on the value x is denoted as $Pr[\mathbf{X} = x]$. Sometimes we denote this as $Pr[x]$ if the random variable $\mathbf{X}$ is fixed. It must be the case that $Pr[x] \geq 0$ for all $x \in X$, and $\sum Pr[x] = 1$.

Suppose $\mathbf{X}$ and $\mathbf{Y}$ are random variables defined on finite sets X and Y, respectively. The joint probability distribution is $Pr[x, y]$. The conditional probability $Pr[x|y]$ denotes the probability that $\mathbf{X}$ takes on the value x given that $\mathbf{Y}$ takes on the value y. The random variables $\mathbf{X}$ and $\mathbf{Y}$ are said to be independent random variables, if $Pr[x, y]{=}Pr[x]Pr[y]$ for all $x \in X$ and $y \in Y$.

Joint probability and conditional probability are related as $Pr[x, y]{=}Pr[x|y]Pr[y]$. Interchanging x and y, we have that $Pr[x, y]{=}Pr[y, x]{=}Pr[y|x]Pr[x]$.

Random variables $\mathbf{X}$ and $\mathbf{Y}$ are independent if and only if $Pr[x|y]{=}Pr[x]$ for all $x \in X$ and $y \in Y$.

Theorem 1 (Bayes' theorem [2])**.** *If $Pr[x] > 0$, then $Pr[x|y] = \frac{Pr[x]Pr[y|x]}{Pr[y]}$.*

Definition 1. *[2] A cryptosystem is a five-tuple (P, C, K, E, D), where the following conditions are satisfied:*

1. *P is a finite set of possible plaintext*
2. *C is a finite set of possible ciphertext*
3. *K, the keyspace, is a finite set of possible keys*
4. *$E : K \times P \to C$ and $D : K \times C \to P$ are called encryption and decryption functions respectively. These functions satisfy $D(k, E(k, m)) = m, \forall m \in P, k \in K$.*

Theorem 2. *[2] A cryptographic system has perfect secrecy if $Pr[m|c]{=}Pr[m]$ for all $m \in P$ and $c \in C$. That is, the a posteriori probability that the plaintext is m, given that the ciphertext is c, is identical to the a priori probability that the plaintext is m.*

A well-known realization of perfect secrecy is the OTP. It was believed to be an unbreakable cryptosystem. Shannon developed a mathematical proof that OTP offers perfect secrecy.

The OTP is vulnerable to a known-plaintext attack, since key K can be computed as the XOR of the message m and ciphertext $E(K, m)$ i.e., $K{=}E(K, m) \oplus m$. Hence a new key needs to be generated and communicated over a secure channel for every message that is going to be sent. This poses key management challenges as mentioned earlier. Consequently the use of OTP in commercial applications is rather limited.

The OTP is a malleable encryption method. An encryption algorithm is *malleable* if it is possible to transform a ciphertext into another ciphertext which decrypts to a related plaintext.

In OTP, the ciphertext is obtained by taking XOR of plaintext with key i.e., $c = m \oplus k$. An adversary can construct an encryption of $m \oplus m'$ for any m' as $c' = c \oplus m' = m \oplus m' \oplus k$, where m' is modified plaintext. Malleability is an undesirable property of cryptographic algorithms.

Accidental reuse of keys or key-reuse error can happen for many reasons. Some of them are highlighted here. In paper (book) based OTP, this is possible due to operator error. Sender may fail to tear off one or more pages of OTP keys. In this case the sender has used the same page for the encryption of more than one segment of message. This operator error will help the active adversary monitoring the communication channel. Suppose sender sends c_1=$m_1 \oplus k_1$ and c_2=$m_2 \oplus k_1$ i.e., sender is using same OTP keys for the encryption of the two different segments of the messages. Adversary can XOR both the ciphertext and try to get plaintext of these two segments. The receiver who uses the key pages in correct sequence will not be able to get the message from the second segment onward.
Printing system by mistake may produce two or more pages of same OTP keys. In this case the same OTP keys are used to encrypt multiple segments of the messages. Here again the adversary will be able to get plaintext segments of the message.
In the hardware-based OTP also key-reuse error is possible due to faulty TRNG. If the TRNG generates same OTP keys for two different sessions, the adversary will be able to get the plaintext of both the sessions.
Finally, apart from key-reuse error, it is possible for OTP keys to be compromised without the sender and receiver not knowing this development. In this scenario the adversary can easily succeed due to the simple and open nature of the OTP algorithm.

The threat of quantum computing to the security of cipher algorithms is a new development that is attracting a lot of attention among cryptographers. Due to Grover's search algorithm [20], brute force attacks reduce the key space size to its square root in the case of symmetric key stream ciphers and block ciphers. In the OTP encryption, key space is as large as the message space keeping the system immune to such attacks. The brute force attacks on a crypt enabled by a large quantum computer would not provide any additional information for the attacker to select the plain message of interest to him. Hence, the OTP is quantum-safe. The properties of OTP are summed up as perfectly secure, quantum-safe, but malleable with no cover for key-reuse error.

3 Latin rectangles and squares

In this section we briefly discuss the Latin rectangles and squares and their applications in design and security analysis of block ciphers.

Definition 2. *[18] An $r \times n$ Latin rectangle is an $r \times n$, $(r < n)$ array made out of the integers $\{1, 2 \ldots n\}$ such that no integer is repeated in any row or in any column.*

An example of a Latin rectangle for $r = 3$ and $n = 5$ is made out of integers $\{1, 2, 3, 4, 5\}$ and is given in Table 1.

Definition 3. *[18] An $n \times n$ Latin rectangle is called a Latin square of order n.*

An example of 5×5 Latin square constructed with integers $\{1, 2, 3, 4, 5\}$ is given in Table 2.

Lemma 1. *[18] Every Latin rectangle can be extended to a Latin square.*

Table 1: Example of 3×5 Latin rectangle

1	2	3	4	5
2	3	4	5	1
3	4	5	1	2

Table 2: Example of 5×5 Latin square

1	2	3	4	5
2	3	4	5	1
3	4	5	1	2
4	5	1	2	3
5	1	2	3	4

Example of 5×5 Latin square given in Table 2 is extension of 3×5 Latin rectangle given in Table 1.

Theorem 3. *[18] For any positive integer n, there is a Latin square of order n.*

Definition 4. *[18] An $n \times n$ Latin square is said to be in reduced form if both its first row and its first column are in their natural order.*

An example of 5×5 Latin square in reduced form is given in Table 3.

Table 3: Example of 5×5 reduced form Latin square

1	2	3	4	5
2	3	4	5	1
3	4	5	1	2
4	5	1	2	3
5	1	2	3	4

Theorem 4. *[19] The total number of Latin squares of order n ($LS(n)$), $n > 2$, is given by $LS(n)=n!(n-1)!T(n)$, where $T(n)$ denotes the number of reduced form Latin squares of order n.*

The number of reduced form Latin squares, $T(n)$ is given in Table 4.

Let n be a positive integer. A Latin square of order n is an $n \times n$ matrix $L=(l_{i,j})$, $1 \leq i, j \geq n$ with entries $l_{i,j} \in \{1, 2 \ldots n\}$, such that each element of the set $\{1, 2 \ldots n\}$ appears exactly once in each row and each column of L. A Latin square defines a cipher over the message space $P=\{1, 2, \ldots n\}$ and the key space $K=\{1, 2, \ldots n\}$ for which the encryption of a plaintext $p \in P$ under a key $k \in K$ is defined by $c=C_k(p)=l_{k,p}$. For the decryption of the ciphertext (c) generated by Latin square $L=l_{i,j}$ with key (k) and plaintext (p), we need the inverse of Latin square. From Latin square $L=(r_i, c_j, l_{i,j})$, inverse Latin square is generated as $L^{-1}=(r_i, l_{i,j}, c_j)$ i.e., in inverse Latin square entries of Latin square and column number of Latin square get swapped. Same thing is illustrated with the help of an example.

Table 4: Values of $T(n)$ for reduced form Latin squares of order upto 12

n	T(n)
2	1
3	1
4	4
5	56
6	9048
7	16942080
8	535281401585
9	377597570964258
10	7580721483160132811489280
11	5.36×10^{33}
12	1.62×10^{44}

Latin square $L{=}l_{i,j}$ of order 4 is given in Table 5. In the table, the row above the first row represents the plaintext $p \in \{1, 2, 3, 4\}$ and the column to the left of the first column represents the key $k \in \{1, 2, 3, 4\}$.

Table 5: Example of 4×4 Latin square used for encryption

K/P	1	2	3	4
1	1	4	3	2
2	4	2	1	3
3	2	3	4	1
4	3	1	2	4

The encryption of plaintext $p{=}3$ using key $k{=}2$ is obtained using Latin square given in Table 5, as $L(k, p){=}l_{2,3}{=}1{=}c$. For the decryption of the ciphertext $c{=}1$, one needs the inverse of the Latin square given in Table 5. This inverse Latin square is in Table 6. In Table 6 the top most row represents the ciphertext $c \in \{1, 2, 3, 4\}$ and the left most column represents the key $k \in \{1, 2, 3, 4\}$.

Table 6: Example of 4×4 inverse Latin square used for decryption

K/C	1	2	3	4
1	1	4	3	2
2	3	2	4	1
3	4	1	2	3
4	2	3	1	4

For the decryption one should have knowledge of ciphertext c and the key k used for encryption. In the given example $c{=}1$ and $k{=}2$. For the decryption, $L^{-1}(k, c){=}L^{-1}(2, 1){=}3{=}p$. The original plaintext $p{=}3$ is obtained. Overall block diagram of encryption and decryption using Latin square and its inverse is given in Figure 1. From the method used for the calculation of inverse Latin square, we have the following theorem.

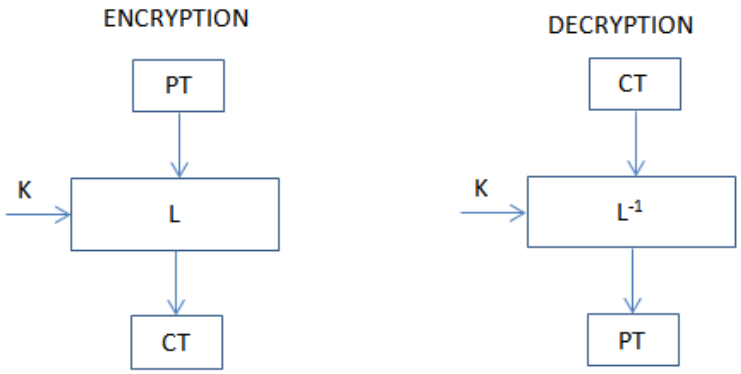

Figure 1: Encryption and decryption using Latin square and inverse Latin square

Theorem 5. *Inverse of a Latin square is another Latin square.*

Theorem 6. *[7] A Latin square $L(k,p)$=$l_{k,p}$=c defines a cipher which achieves perfect secrecy if the key is uniformly distributed, independent from the plaintext, and is used only once.*

3.1 One-time-pad as Latin square

In this section we discuss the OTP as a particular case of Latin square-based cipher. In Table 7 the top most row represents the plaintext, $p \in \{0,1\}$ and the left most column represents the key, $k \in \{0,1\}$. The Latin square corresponding to the encryption of the OTP i.e., $C = K \oplus P$, is given in Table 7.

Table 7: OTP as Latin square for encryption

K/P	0	1
0	0	1
1	1	0

The Latin square corresponding to the decryption of the OTP i.e., $P = K \oplus C$, is given in Table 8. In the table the top most row represents the ciphertext $c \in \{0,1\}$ and the left most column represents the key $k \in \{0,1\}$.

Table 8: OTP as Latin square for decryption

K/C	0	1
0	0	1
1	1	0

For example, suppose plaintext P=$\{0,1,1,0\}$ and key k=$\{1,1,1,0\}$. Ciphertext, C=$K \oplus P$=$\{1,0,0,0\}$. Same ciphertext can also be generated using Table 7 i.e., OTP as Latin square for encryption. For the decryption, P=$K \oplus C$. Same plaintext can also be generated using Table 8 i.e., OTP as Latin square for decryption. For the OTP, encryption and decryption Latin squares are the same.

The OTP is malleable encryption. However, ciphers based on Latin square may or may not be malleable encryption. By choosing a suitable block ciphers that form Latin square, one can have non-malleable encryption. Latin square-based ciphers are not secure with respect to the accidental reuse of the keys. Ciphers based on Latin squares are by the way quantum-safe. The brute force attacks on a ciphertext would not provide any additional information for the attacker to get the plaintext of interest. Hence, the ciphers based on Latin square are quantum-safe encryption algorithm. Overall properties of ciphers based on Latin square are summarized as perfectly secure and quantum-safe but with status of malleability and cover for key-reuse error as unknown.

3.2 Block ciphers as Latin rectangles or Latin Squares

In this section we discuss equivalence between block ciphers and Latin rectangles or Latin squares. More formally we prove a Theorem 7, which shows the relation between block ciphers and Latin rectangles. Block ciphers are important primitives used in symmetric key cryptographic protocols. A block cipher is specified in term of encryption and decryption functions. Encryption function $E_K(P)$=$E(K,P)$ is mapping defined as, $E_K(P) : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$, which takes input key K of bit length k, called the key size of block ciphers, and plaintext P of bit length n, called the block size of block ciphers and returns ciphertext C of bit size n. For each key $K \in \{0,1\}^k$, $E_K(P)$ is required to be bijective mapping from $\{0,1\}^n \rightarrow \{0,1\}^n$. The inverse of $E_K(P)$ is $D_K(C)$ =$D(K,C)$ defined as, $D_K(C) : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$, such that $D_K(E_K(P))$=P, $\forall K \in \{0,1\}^k$ and $\forall P \in \{0,1\}^n$. For each key $K \in \{0,1\}^k$, $E_K(P)$ is a permutation over the set of input blocks. Each key selects one permutation from the set of all $(2^n)!$ permutations. In cryptographic protocols only cryptographically strong block ciphers are used i.e., Block ciphers which withstand crypt analytic attacks. A block cipher like $E_K(P)$=P is identity permutation and is not used in cryptographic applications.

Theorem 7. *Let $E_K(P) : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$, $k \neq n$, be a block cipher with following properties.*

1. *Suppose $E(K_i,P_j)$=$C_{i,j}$ and $E(K_i,P_k)$=$C_{i,k}$, for $i \in \{0,1,\ldots 2^k-1\}$ and for $j,k \in \{0,1,\ldots 2^n-1\}$. If $C_{i,j}$=$C_{i,k}$ then P_j=P_k. That is, if encryption of plaintext P_j and P_k under the key K_i are same then P_j=P_k. All block ciphers have to satisfy this condition otherwise one cannot have proper decryption.*

2. *Suppose $E(K_i,P_j)$=$C_{i,j}$ and $E(K_k,P_j)$=$C_{k,j}$ for $i,k \in \{0,1,\ldots 2^k-1\}$ and for $j \in \{0,1,\ldots 2^n-1\}$. If $C_{i,j}$=$C_{k,j}$ then K_i=K_k. That is, if encryption of any plaintext P_j under the keys K_i and K_k are same then K_i=K_k.*

Then block cipher $E_K(P)$=$E(K,P)$ forms a Latin rectangle.

Proof. Let C=$(C_{i,j})$, for $i \in \{0,1,\ldots 2^k-1\}$ and $j \in \{0,1,\ldots 2^n-1\}$ be a $2^k \times 2^n$ rectangle corresponding to block cipher $E(K,P)$ given below. In rectangle C top most row represent all possible n-bit plaintexts and left most column represents the all possible k-bit keys. The entries of rectangle C=$C_{i,j}$=$E(K_i,P_j)$, for $i \in \{0,1,\ldots 2^k-1\}$ and $j \in \{0,1,\ldots 2^n-1\}$.

K/P	0	1	...	i	...	j	...	k	...	2^n-1
0	$C_{0,0}$	$C_{0,1}$	...	$C_{0,i}$	...	$C_{0,j}$	...	$C_{0,k}$	...	$C_{0,2^n-1}$
1	$C_{1,0}$	$C_{1,1}$	...	$C_{1,i}$	...	$C_{1,j}$	...	$C_{1,k}$	...	$C_{1,2^n-1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	$C_{i,0}$	$C_{i,1}$	...	$C_{i,i}$	...	$C_{i,j}$	...	$C_{i,k}$	...	$C_{i,2^n-1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
j	$C_{j,0}$	$C_{j,1}$	...	$C_{j,i}$	...	$C_{j,j}$	...	$C_{j,k}$	...	$C_{j,2^n-1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
k	$C_{k,0}$	$C_{k,1}$	...	$C_{k,i}$	...	$C_{k,j}$	...	$C_{k,k}$	...	$C_{k,2^n-1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
2^k-1	$C_{2^k-1,0}$	$C_{2^k-1,1}$	...	$C_{2^k-1,i}$	...	$C_{2^k-1,j}$	...	$C_{2^k-1,k}$	...	$C_{2^k-1,2^n-1}$

We want to prove that rectangle $C=(C_{i,j})$ is Latin rectangle. That is, in each row, numbers $\{0,1,\ldots,2^n-1\}$ occurs exactly once, and in each column, numbers $\{0,1,\ldots,2^k-1\}$ occurs exactly once. Suppose in row $i \in \{0,1,\ldots,2^k-1\}$, $\exists\, j,k \in \{0,1,\ldots,2^n-1\}$, $j \neq k$ such that $C_{i,j}=C_{i,k}$, i.e., $E(K_i,P_j)=E(K_i,P_k)$. This is a contradiction to condition 1. Therefore, $P_j=P_k$.

Suppose in column $j \in \{0,1,\ldots,2^n-1\}$, $\exists\, i,k \in \{0,1,\ldots,2^k-1\}$, $i \neq k$ such that $C_{i,j}=C_{k,j}$, i.e., $E(K_i,P_j)=E(K_k,P_j)$. This is a contradiction to condition 2. Therefore, $K_i=K_k$. Hence the rectangle $C=(C_{i,j})$ forms a Latin rectangle.

□

Corollory 1. *Let $E_K(P) : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$, be a block cipher with following properties.*

1. *Suppose $E(K_i,P_j)=C_{i,j}$ and $E(K_i,P_k)=C_{i,k}$, for $i \in \{0,1,\ldots 2^n-1\}$ and for $j,k \in \{0,1,\ldots 2^n-1\}$. If $C_{i,j}=C_{i,k}$ then $P_j=P_k$. That is, encryption of plaintext P_j and P_k under the key K_i are same implies $P_j=P_k$.*

2. *Suppose $E(K_i,P_j)=C_{i,j}$ and $E(K_k,P_j)=C_{k,j}$ for $i,k \in \{0,1,\ldots 2^n-1\}$ and for $j \in \{0,1,\ldots 2^n-1\}$. If $C_{i,j}=C_{k,j}$ then $K_i=K_k$. That is, encryption of plaintext P_j under the key K_i and K_k are same implies $K_i=K_k$.*

Then block cipher $E_K(P)=E(K,P)$ forms a Latin square.

Proof. From Theorem 7, with $k=n$, we get the result.

□

4 Extensions of the OTP

In this section we discuss two extensions of the OTP. The first extension is using invertible confusion and diffusion operators. These types of operators are used in computationally secure block ciphers in conjunction with round keys. The round keys are derived from the main key and are only unknown parameter to adversary. In the case of first extension of OTP, by keeping confusion and diffusion operators secure some advantage is gained as discussed later. In general, the first extension of OTP is a kind of hybrid cipher using perfectly secure cipher and computationally secure ciphers. Confusion operators $C_1, C_2, \ldots, C_n$ may include some unknown quantity like round key. The second extension of the OTP is using a block cipher which forms a Latin square. The proposed extensions keep all the properties of OTP intact and have some more properties that are discussed in the following sections.

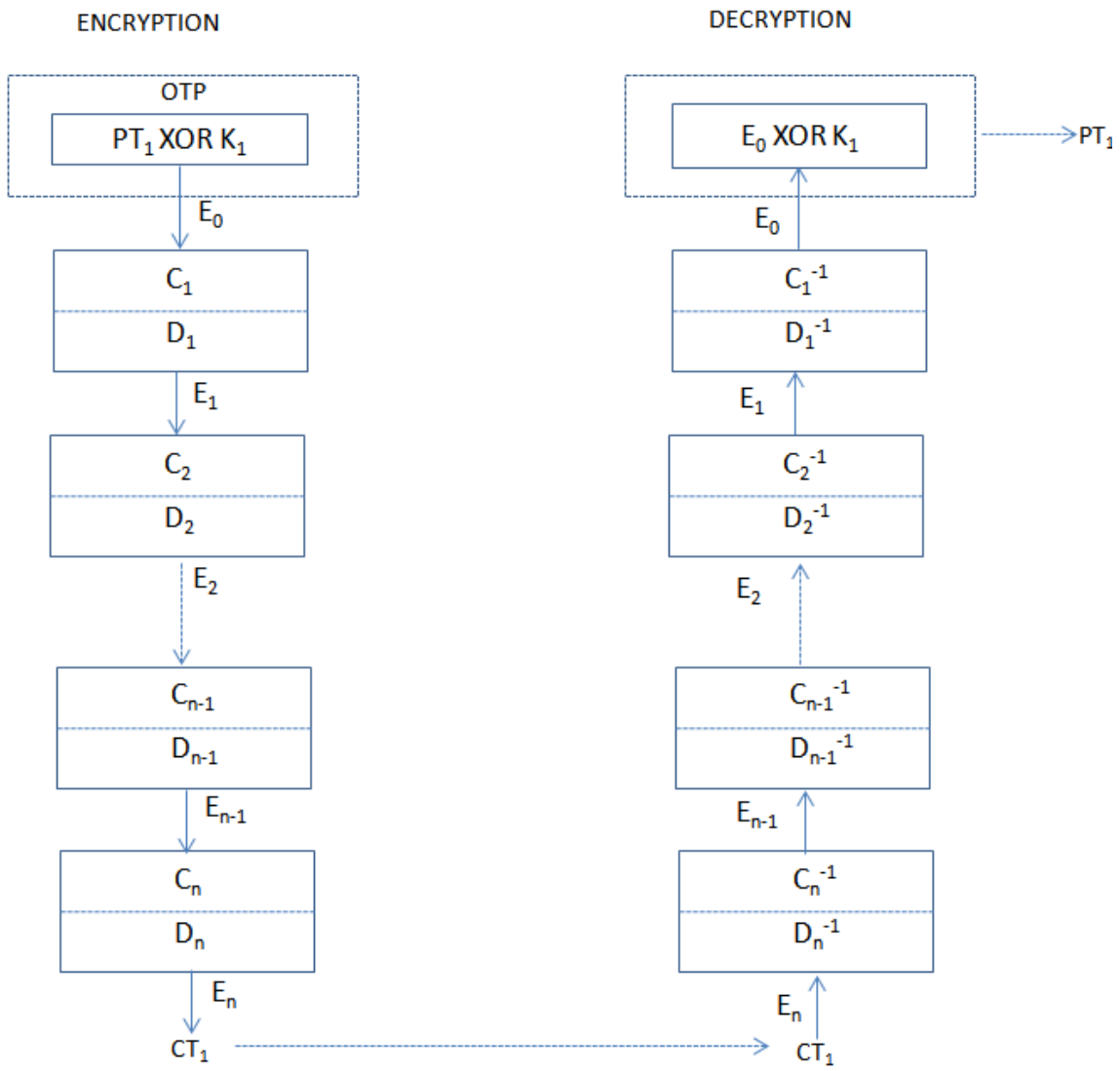

Figure 2: First extension of one-time-pad

4.1 First extension of OTP

In this section we discuss, the first extension of the OTP using confusion and diffusion operators. Confusion operators are $C_1, C_2, \ldots, C_n$. Each confusion operator is invertible and inverse of C_i=C_i^{-1} i.e., $C_iC_i^{-1}$=$C_i^{-1}C_i$=I, for i=$1, 2, \ldots n$. Where I is identity operator. Diffusion operators are $D_1, D_2, \ldots, D_n$. Each diffusion operator is invertible and inverse of D_i=D_i^{-1} i.e., $D_iD_i^{-1}$=$D_i^{-1}D_i$=I, for i=$1, 2, \ldots n$. The confusion and diffusion operators are not commutative i.e., $C_iD_i \neq D_iC_i$ for i=$1, 2, \ldots n$. Overall block diagram of first extension of OTP is depicted in Figure 2.

The encryption operation of the first extension of OTP is as follows.

E_0=$PT_1 \oplus K_1$;
E_1=$D_1(C_1(E_0))$;
E_2=$D_2(C_2(E_1))$;
$\vdots$
E_{n-1}=$D_{n-1}(C_{n-1}(E_{n-2}))$;
E_n=$D_n(C_n(E_{n-1}))$;
CT_1=E_n.

The decryption operation of the first extension of OTP is as follows.
E_n=CT_1;
E_{n-1}=$D_n^{-1}(C_n^{-1}(E_n))$;
E_{n-2}=$D_{n-1}^{-1}(C_{n-1}^{-1}(E_{n-1}))$;
$\vdots$
E_1=$D_2^{-1}(C_2^{-1}(E_2))$;
E_0=$D_1^{-1}(C_1^{-1}(E_1))$;
PT_1=$E_0 \oplus K_1$.

In Figure 2 encryption and decryption operations are discussed for one block of plaintext. Typical block sizes are $n \in \{128, 256\}$ bits. In order to integrate confusion and diffusion operators with the OTP, the message has to be segmented into blocks of size n bits. Padding may be required in the last block. The first extension of the OTP keeps all the strengths of the OTP intact i.e., it is perfectly secure and quantum-safe. By combining OTP with confusion and diffusion operators, first extension of OTP becomes non-malleable encryption. Also with assumption that confusion and diffusion operators are secret i.e., not known to adversary the OTP keys get cover for reuse error. Overall properties of first extension of OTP under the assumption that confusion and diffusion operators are secret are summed up as perfectly secure, quantum-safe, non-malleable with cover for key-reuse error.

4.2 Second extension of one-time-pad

In this section we discuss the second extension of OTP using block ciphers which form Latin squares. The second extension of the OTP can also be seen as a particular case of first extension of the OTP. Most of the computationally secure block ciphers have been using confusion and diffusion operators with round keys. The second extension of OTP also keep all the strengths of OTP intact i.e., it is perfectly secure and quantum-safe. In addition to these properties the second extension of OTP is non-malleable encryption. Non-malleability is due to the design of the block ciphers. One bit change in input of block cipher produce $\frac{n}{2}$ bits change in output. Also with assumption that block cipher used in the second extension of the OTP is secret i.e., not known to adversary the OTP keys get some cover from reuse. Overall block diagram of second extension of OTP is depicted in Figure 3.

From Figure 3 it is clear that the second extension will do proper encryption and decryption. Working with n-bit blocks of message and key sequence and using an operator having Latin square structure will preserve the perfect secrecy aspect of OTP. In the second extension of the OTP, XOR operation of OTP is replaced by a block cipher, which forms a Latin square. Taking n-bit plaintext and n-bit key as inputs it produces an n-bit ciphertext. Also given n-bit ciphertext and n-bit key as inputs it to produces n-bit plaintext. A simple extension to the OTP is the replacement of the single bit XOR algorithm by n-bit block cipher. The message has to be segmented into blocks of size n-bit. Padding may be required in the last block. Then the message and encryption key are segmented into blocks of n bits. Starting from the first block, consecutive key blocks of n bits may be used to encrypt the corresponding plaintext blocks to get the ciphertext blocks. To recover the plaintext from the ciphertext the decryption process has to be reverse as illustrated in Figure 3.

Overall properties of second extension of the OTP under the assumption that design of block cipher is secret are the same as those of the first extension.

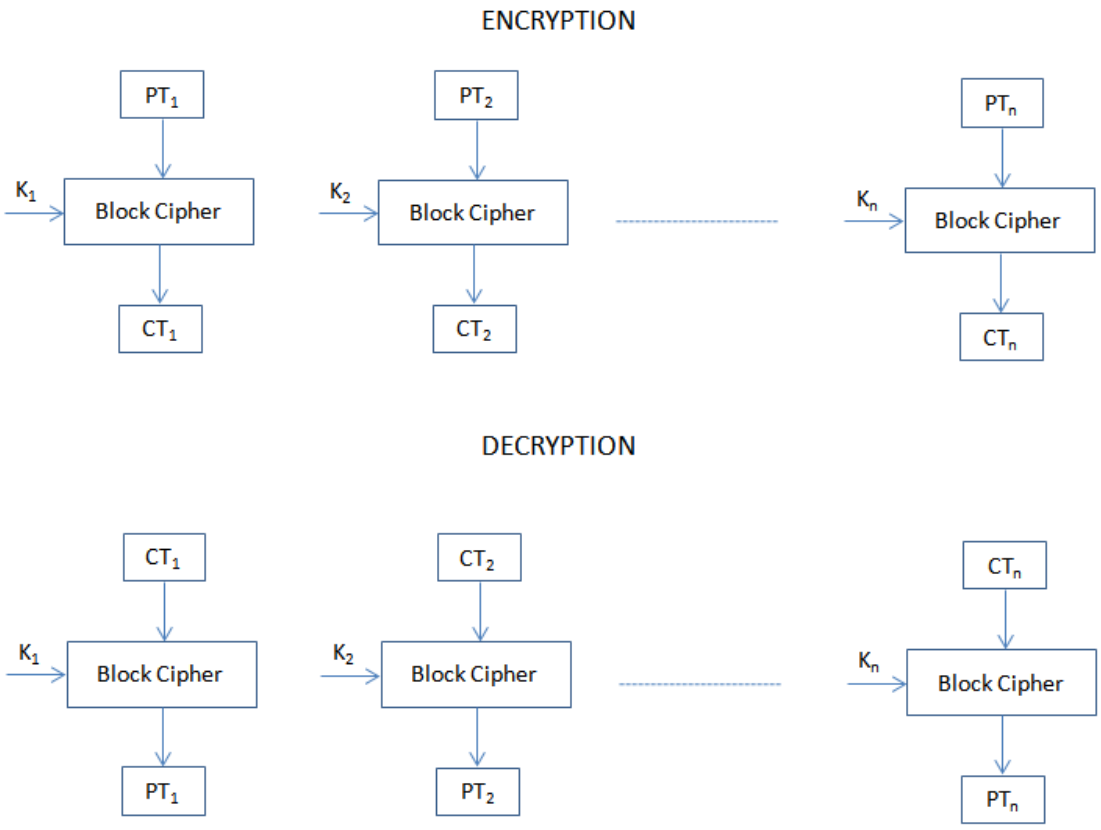

Figure 3: Second extension of one-time-pad

5 CONCLUSION

In this paper we discuss a relation between block ciphers and Latin rectangles based on Theorem 7. A particular case of Theorem 7 is used to derive relation between block ciphers and Latin squares,as Corollary 1. We also propose two extensions of the OTP. The first extension proposes to use invertible, non-commutative confusion and diffusion operators. Under the assumption that confusion and diffusion operators are not known to the adversary, the first extension gives some cover for key-reuse error. It is also non-malleable. The second extension proposes to use a block cipher which forms a Latin square. It shares all the desirable properties of the first extension and is more practical of the two. These two extensions keep all the properties of OTP intact i.e., they are perfectly secure and quantum-safe. At the same time both extensions are non-malleable encryption and provides some cover for accidental reuse of keys. Furthermore, in case of a compromise of segments of OTP keys, both extensions offer some cove against recovery of plaintext.

References

[1] A. Menezes, P. Van Oorschot and S. Vanstone, *Handbook of Applied Cryptography, CRC Press, 1997.*

[2] Stinson, Douglas Robert, and Maura Paterson. *Cryptography: theory and practice. CRC press, 2018.*

[3] R. A. Rupple, *Analysis and Design of Stream Ciphers, Springer-Verlag, 1986.*

[4] Kitsos, P., et al. *Hardware implementation of the RC4 stream cipher. 2003 46th Midwest Symposium on Circuits and Systems. Vol. 3. IEEE, 2003.*

[5] *https://www.ecrypt.eu.org/stream/*, last access on 05 Aug 2020.

[6] Daemen, Joan and Rijmen, Vincent *The design of Rijndael, Springer, 2002.*

[7] Baigneres, Thomas, et al. *A classical introduction to cryptography exercise book. Springer science and business media, 2006.*

[8] Kahn, David. *The Codebreakers: The comprehensive history of secret communication from ancient times to the internet. Simon and Schuster, 1996.*

[9] Dalimunthe, Aulia Rahman, et al. *Vernam Cipher with Complement Method and Optimization Key with Genetic Algorithm. Journal of Physics: Conference Series. Vol. 1235. No. 1. IOP Publishing, 2019.*

[10] Brosas, Deborah G., Ariel M. Sison, and Ruji P. Medina. *Strengthening The Vernam Cipher Algorithm Using Multilevel Encryption Techniques. IJSTR Volume 8, 2019.*

[11] Saraswat, Aditi, et al. *An Extended Hybridization of Vigenere and Caesar cipher techniques for secure communication. Procedia Computer Science 92 (2016): 355-360.*

[12] Leung, Debbie W. *Quantum vernam cipher. arXiv preprint quant-ph/0012077 (2000).*

[13] Miller, Frank. *Telegraphic code to insure privacy and secrecy in the transmission of telegrams. CM Cornwell, 1882.*

[14] Bellovin, Steven M., Frank Miller *Inventor of the one-time pad. Cryptologia 35.3 (2011): 203-222.*

[15] Shannon, Claude E. *Communication theory of secrecy systems. The Bell system technical journal 28.4 (1949): 656-715.*

[16] *https://en.wikipedia.org/wiki/SIGSALY* , last access on 5 Aug 2020.

[17] *http://www.bu.edu/articles/2015/secure-quantum-key-distribution-encryption/*, last access on 5 Aug 2020.

[18] Lidl, Rudolf, and Günter Pilz. *Applied abstract algebra. Springer Science and Business Media, 2012.*

[19] Battey, Matthew, and Abhishek Parakh. *An efficient quasigroup block cipher. Wireless personal communications 73.1 (2013): 63-76.*

[20] Nielsen, Michael A., and Isaac Chuang. *Quantum computation and quantum information. (2002): 558-559.*