An Optimal Fault-Tolerant Routing Algorithm for Double-Loop Networks

Yu-Liang Liu, Yue-Li Wang, and D.J. Guan

Abstract—A weighted double-loop network can be modeled by a directed graph $G(n; h_1, h_2; w_1, w_2)$ with vertex set $Z_n = \{0, 1, \ldots, n-1\}$ and edge set $E = E_1 \cup E_2$, where $E_1 = \{(u, u + h_1) | u \in Z_n\}$, $E_2 = \{(u, u + h_2) | u \in Z_n\}$. Assume that the weight of each edge in $E_1$ is $w_1$ and the weight of each edge in $E_2$ is $w_2$. In this paper, we present an optimal routing algorithm on double-loop networks under the case where there is at most one faulty element. Our algorithm is based on the fact that the shortest path from a vertex to any other vertex in a double-loop network is in the L-shape region.

Index Terms—Double-loop networks, fault-tolerant, optimal message routing.

1 INTRODUCTION

A weighted double-loop network $G(n; h_1, h_2; w_1, w_2)$ is a directed graph with vertex set $Z_n = \{0, 1, \ldots, n-1\}$ and edge set $E = E_1 \cup E_2$, where $n$, $h_1$, and $h_2$ are positive integers and $w_1$, $w_2$ are positive real numbers which are the weights of the edges in $E_1$ and $E_2$, respectively, and $E_1 = \{(u, u + h_1) | u \in Z_n\}$, $E_2 = \{(u, u + h_2) | u \in Z_n\}$ [5], [11]. Note that the addition in the finite group $Z_n$ is defined as the addition in $Z$ and then modulo $n$. In this paper, we present an optimal message routing algorithm on double-loop networks in which there is at most one faulty edge or vertex.

A double-loop network $G(n; h_1, h_2; w_1, w_2)$ is strongly connected if and only if $n$, $h_1$, $h_2$ are relatively prime [7]. We assume that, in this paper, the double-loop network is strongly connected. For example, the graph shown in Fig. 1 is $G(8; 3, 2; 1, 2)$, which is strongly connected. The weight of each dotted edge is 2 and the weight of each solid edge is 1. Note that, in most cases, there are many shortest paths between any two vertices. For the shortest path from vertex 0 to vertex 7 can pass through vertices 2 and 4 or vertices 3 and 5. The lengths of both paths are 5.

In the design and implementation of local area networks, the ring topology has been widely used because of its simplicity, expandability, and regularity. A popular variation of the ring network is the double-loop network. There are a lot of papers which discuss the properties of a double-loop network. The diameter problem on double-loop networks is discussed in [1], [2]. The message routing and fault-tolerant message routing problems on double-loop networks are discussed in [3], [4], [6], [8]. The permutation routing problem on double-loop networks is discussed in [9]. Moreover, the Hamiltonian cycle problem on double-loop networks with exactly one faulty element is solved in [12].

Message routing is a basic function in communication networks. The problem in message routing is to find a route along which messages should be sent. It is called an optimal message routing if every message is sent along a shortest path from its source vertex to its destination vertex. In [8], Guan presented an optimal message routing algorithm for double-loop networks. Moreover, he also gave a fault-tolerant message routing algorithm under the case where there is at most one faulty edge or vertex. However, his algorithm does not guarantee to send a message along a shortest path after a faulty edge or a faulty vertex is detected. We shall propose an optimal message routing algorithm in the case that the network contains at most one faulty element.

The remaining part of this paper is organized as follows: In Section 2, we introduce Guan’s Algorithm and some properties of double-loop networks. In Section 3, we shall propose an optimal message routing algorithm after a faulty edge or a faulty vertex is detected. Section 4 contains our concluding remarks.

2 GUAN’S ALGORITHM

Let $s$ be the first vertex that detects the faulty vertex $f$. If the edge between $s$ and $f$ is in $E_1$ (respectively, $E_2$), then it is called a type-$1$-fault (respectively, type-$2$-fault). In Guan’s algorithm, when a faulty element is first found, the algorithm checks if there exists an alternative shortest path. If there is an alternative shortest path, then the algorithm routes the message in a normal way; otherwise, the algorithm enters into the faulty-mode. The faulty-mode of Guan’s algorithm can be described as in Algorithm A. For simplicity, we only consider type-$1$-faults in Algorithm A. Note that Algorithm A is applied only when the algorithm enters into the faulty-mode and the vertex $u$ that tries to send the message is not the destination $d$ of the message.
Algorithm A

Input: A double-loop network \( G(n; h_1, h_2; w_1, w_2) \), destination vertex \( d \), and faulty vertex \( f \).

Output: An edge \( e \) through which the message is sent by vertex \( u \).

Method:

Step 1. If \( u \) is the vertex that detects the faulty element \( f \), then

\[ e = (u, u + h_2), \text{ otherwise, execute Step 2 and Step 3.} \]

Step 2. Solve the following four equations

\[
egin{align*}
\hat{h}_1 x_0 & \equiv d - u \pmod{n} \\
\hat{h}_2 y_0 & \equiv d - u \pmod{n} \\
\hat{h}_1 x_1 & \equiv f - u \pmod{n} \\
\hat{h}_2 y_1 & \equiv f - u \pmod{n}
\end{align*}
\]

Step 3. If \( 0 < x_0 < x_1 \) or \( y_0 = \infty \) or \( y_0 > y_1 \), then \( e \) is the edge \((u, u + h_1)\); otherwise, \( e \) is the edge \((u, u + h_2)\).

End of Algorithm A

Note that \( x_0 \) (respectively, \( y_0 \)) is the number of edges in the path from \( u \) to \( d \) that only uses edges in \( E_1 \) (respectively, \( E_2 \)), and \( x_1 \) (respectively, \( y_1 \)) is the number of edges in the path from \( u \) to \( f \) that only uses edges in \( E_1 \) (respectively, \( E_2 \)), and if no such path exists, then its value is \( \infty \).

We use an example to show that Algorithm A does not always route a message through a shortest path. Let \( G(14; 4, 3; 1, 2) \) be a double-loop network. Assume that the source, destination, and faulty vertices are vertices 0, 8, and 4, respectively. Initially, in Step 1, since vertex 4 is the faulty vertex \( f \), \( e \) is the edge \((0, 3)\). It means that vertex 0 will send the message to vertex 3. Now, consider \( u = 3 \). It needs to solve the four equations in Step 2. We can obtain \( x_0 = \infty \), \( y_0 = 11 \), \( x_1 = \infty \), and \( y_1 = 5 \). It can be found that \( e \) is the edge \((3, 7)\). That is, vertex 3 will send the message to vertex 7. Therefore, Algorithm A will find the path passing through vertices 0, 3, 7, 10, 13, 2, 5, and 8. The length of this path is 13. However, in this case, a shortest path will pass through vertices 0, 3, 7, 11, 1, 5, and 8 and its length is 8.

Before we present our message routing algorithm, we first introduce some properties of double-loop networks. It is convenient to map \( \mathbb{Z}^2 \) into the vertex set of a double-loop network \( G(n; h_1, h_2; w_1, w_2) \) \([2, 6]\). A point \((i, j)\) in the plane represents the vertex \((h_1 i + h_2 j) \mod n\) in the network. On the other hand, we also say that the coordinate of the vertex \((h_1 i + h_2 j) \mod n\) is \((i, j)\). Notice that if the coordinate of vertex \( u \) is \((i, j)\), then the coordinates of its two adjacent vertices \((u + h_1) \mod n\) and \((u + h_2) \mod n\) are \((i + 1, j)\) and \((i, j + 1)\), respectively. Let \( S \) (respectively, \( T \)) be the smallest positive integer such that \( h_1 S \equiv h_2 \pmod{n} \) (respectively, \( h_1 t \equiv h_2 \pmod{n} \)) for some nonnegative integer \( w_2 s \leq w_1 S \) (respectively, \( w_1 t < w_2 T \)). For simplicity, we consider only a region \( D \), a subset of the entire plane which contains only the points \((i, j)\) for \( 0 \leq i < n / \gcd(n, h_1) \) and \( 0 \leq j < n / \gcd(n, h_2) \). We call it the \( D \)-region of the network. For example, the \( D \)-region of the graph in Fig. 1 is shown in Table 1.

Let \( P \) and \( Q \) be the pair of nonzero integers with minimum \( P \) satisfying \( Ph_1 + Qh_2 \equiv 0 \pmod{n} \) and with minimum \( Pw_1 + Qw_2 \). Cheng and Hwang proved that, in the \( D \)-region, there exists a region \( L \) which contains all the vertices exactly once. This implies that the shortest path from vertex 0 to any other vertex is within the region \( L \) \([2]\). The region \( L \) is \( L \)-shaped and its corner points are \((0, 0)\), \((0, T - 1)\), \((P - 1, T - 1)\), \((P - 1, Q - 1)\), \((S - 1, Q - 1)\), and \((S - 1, 0)\). We call \( P, Q, S, \) and \( T \) the routing parameters of the network. Since the network is vertex transitive, the region \( L \) that contains all the shortest paths can also be defined for any other vertex as the source vertex. We shall use \( L_u \) to denote the region that contains all the shortest paths with source vertex \( u \). For example, in Table 1, the shaded region is \( L_0 \). The values of \( P, Q, S, \) and \( T \) are 2, 1, 4, and 3, respectively. For a double-loop network \( G(n; h_1, h_2; w_1, w_2) \), its routing parameters \( P, Q, S, \) and \( T \) can be computed in \( O(\log n) \) time \([2, 10]\).

Let \( u \) and \( v \) be a pair of adjacent vertices in \( G(n; h_1, h_2; w_1, w_2) \) and the coordinates of \( u \) and \( v \) are \((i, j)\) and \((i, j + 1)\), respectively. By the properties of the \( L \)-region, the vertices in the bottom row of \( L_u \) will appear in the upper side of \( L_v \). Moreover, the vertex at position \((x, y)\) in the bottom row of \( L_u \) will appear at position \((x', y')\) of the upper side of \( L_v \), where \( x' = (x + P) \mod S \) and

\[
y' = \begin{cases} 
  j + Q, & \text{if } (x - i) < S - P \\
  j + T, & \text{otherwise}
\end{cases}
\]

For example, in Table 1, vertex 6 is at position \((2, 0)\) in the bottom row of \( L_0 \). It appears at position \((0, 3)\) which is in the upper side of \( L_2 \).
3 Fault-Tolerant Message Routing Algorithm

In this section, we shall consider the message routing algorithm after a faulty vertex is detected. We assume that there is a type-1-fault. To find a shortest route from the source to the destination, there are several cases to consider. We shall describe them in the following lemmas.

Let the coordinates of respectively. For simplicity, we assume that vertex 0 is the position \( P \). The routing parameters \( s, d, f \) be the source, destination, and faulty vertices, respectively. For simplicity, we assume that vertex 0 is the source vertex. Since the network is vertex transitive, we can handle other vertices as the source vertex in a similar way. Let the coordinates of \( s, d, f \) in \( L_u \) be \((0, 0), (i, 0), \) and \((1, 0)\), respectively, where \( i > 1 \). The \( L \)-region is defined by the routing parameters \( P, Q, S, \) and \( T \) as defined in Section 2. Notice that if the destination vertex is not at position \((i, 0)\), then there is an alternative shortest path from \( s \) to \( d \). Therefore, an optimal message routing path can be found by the normal message routing algorithm by sending the message to \((0, 1)\). In this case, the routing algorithm will not enter into the faulty mode.

**Lemma 1.** If \( Q \geq 2 \), then the length of a shortest path from \( s \) to \( d \) is \( rw_1 + kw_2 \), where \( r = (i + P) \mod S \) and

\[
  k = \begin{cases} 
    Q, & \text{if } i < S - P \\
    T, & \text{if } i \geq S - P.
  \end{cases}
\]

**Proof.** Let \( u \) be the vertex at position \((0, 1)\). Since \( f \) is at position \((1, 0)\), \( s \) has to pass through \( u \) to \( d \), the vertices at positions \((0, 0), (1, 0), \ldots, (S - 1, 0)\) in \( L_u \) will appear at positions \((0, T), (1, T), \ldots, (P - 1, T)(P, Q), (P + 1, Q), \ldots, (S - 1, Q)\) in \( L_u \). Notice that the vertex at position \((x, 0)\) in \( L_u \) will appear at position \((x + P) \mod S, Q\) (respectively, \((x + P) \mod S, T\)) in \( L_u \) if \( x < S - P \) (respectively, \( x \geq S - P \)). The position of \( d \) in \( L_u \) is \((r, k)\). Since \( d \) is within \( L_u \) and \( Q \geq 2 \), a shortest path from \( s \) to \( d \) can pass through the vertices at positions \((0, 1), (1, 1), \ldots, (r, 1), (r, 2), \ldots, (r, k - 1)\). Note that \( f \) is not on the above path since \( f \) is at one of the positions \((0, T), (1, T), \ldots, (P - 1, T), (P, Q), (P + 1, Q), \ldots, (S - 1, Q)\) in \( L_u \), except \((r, k)\). The length of the above shortest path is \( rw_1 + kw_2 \).

**Figs. 2a and 2b** are the illustrations of Lemma 1.

**Lemma 2.** Let \( r = (i + kP) \mod S \), where \( k \) is the smallest positive integer such that \( i + kP \geq S \). If \( Q = 1 \) and \( k \geq 2 \), then \( f \) does not appear in any one of the positions \((0, T + k - 2), (1, T + k - 2), \ldots, (r, T + k - 2)\) of the \( D \)-region.

**Proof.** Since \( k \) is the smallest positive integer such that \( i + kP \geq S \), it is clear that \( i + (k - 1)P < S \). According to this inequality, we can derive \( r < P \) as follows:

\[
  \begin{align*}
    i + (k - 1)P &< S \\
    i + (k - 1)P + P &< S + P \\
    i + kP &< S + P \\
    i + kP - S &< S + P - S \\
    r &< P.
  \end{align*}
\]

The position of \( f \) in row \( T + k - 2 \) of the \( D \)-region is \((1 + (T + k - 2)P, T + k - 2)\). We now prove the lemma by contradiction. Assume that \( f \) is in one of the positions \((0, T + k - 2), (1, T + k - 2), \ldots, (r, T + k - 2)\). It means that \( 1 + (T + k - 2) \leq r \). Thus,

\[
  \begin{align*}
    1 + (T + k - 2)P &\leq r \\
    1 + (T + k - 2)P &< P \\
    (T + k - 3)P &< -1.
  \end{align*}
\]

Since \( P \) and \( T \) are positive numbers, \( k \) must be less than two. It is a contradiction and the lemma follows. \( \square \)

**Lemma 3.** If \( Q = 1 \) and \( i + P \geq S \), then \( f \) is not in any one of the positions \((0, 1), (1, 1), \ldots, (i + P - S, 1)\) of the \( D \)-region.

**Proof.** The position of \( f \) in row 1 of the \( D \)-region is \((1 + P, 1)\). Let \( i + x = S \). Clearly, \( x \) is greater than 0. Thus,
Let \( r = (i + kP) \mod S \), where \( k \) is the smallest positive integer such that \( i + kP \geq S \). If \( Q = 1 \), then the length of the shortest path from \( s \) to \( d \) is \( rw_1 + (T + k - 1)w_2 \).

**Proof.** We consider the following two cases: 1) \( T + k = 2 \) and 2) \( T + k > 2 \). The equation \( T + k = 2 \) holds only when \( T = k = 1 \). In this case, by Lemma 3, there exists a path from \( s \) to \( d \) passing through the vertices in the positions \((0, 1), (1, 1), \ldots, (i + P - S - 1, 1)\) of the \( D \)-region. It is obvious that the above path is the shortest. Its length is \( rw_1 + (T + k - 1)w_2 \).

Now, we consider the case where \( T + k > 2 \). Since the position of \( f \) in row \( i \), \( 0 \leq i < T + k - 1 \), of the \( D \)-region is \((1 + iP, i)\), \( f \) does not appear in any one of the positions \((0, 1), (0, 2), \ldots, (0, T + k - 2)\). By Lemma 2, there exists a path passing through the vertices in the positions
\[
(0, 1), (0, 2), \ldots, (0, T + k - 2), (1, T + k - 2), \ldots, (r, T + k - 2)
\]
of the \( D \)-region (see Fig. 3 for an illustration). Since \( k \) is the smallest positive integer such that \( i + kP \geq S \), the \( x \) coordinate of \( f \) must be less than that of \( d \) in each row \( t \), \( 0 \leq t < T + k - 1 \), of the \( D \)-region. Thus, any shortest path from \( s \) to \( d \) contains at least \( T + k - 1 \) edges in \( E_2 \). Since it is within the \( L \)-region, the above path is a shortest path. Its length is also \( rw_1 + (T + k - 1)w_2 \). \( \square \)

**Lemma 5.** Let \( s \), \( d \), and \( f \) denote the source, destination, and faulty vertices, respectively, and their positions in \( L \), be \((0, 0)\), \((0, j)\), and \((0, 1)\), respectively. The length of the shortest path is \( rw_1 + tw_2 \), where \( r \) and \( t \) are defined by:
\[
r = \begin{cases} P & \text{if } P \geq 2 \text{ and } j < T - Q, \\ S & \text{if } P \geq 2 \text{ and } j \geq T - Q, \\ S + k - 1 & \text{if } P = 1 \end{cases}
\]
\[
t = \begin{cases} (j + Q) \mod T & \text{if } P \geq 2, \\ (j + kQ) \mod T & \text{if } P = 1. \end{cases}
\]
and \( k \) is the smallest positive integer such that \( j + kQ \geq T \).

**Proof.** With similar reasoning as in Lemmas 1 and 4, the lemma follows. \( \square \)

Now, we are at the position to describe our algorithm. We will first handle the case when a faulty vertex is detected. The case when a faulty edge is detected can be handled in a similar way.

Assume that the message is at vertex \( u \). If a faulty vertex is detected and there are no alternative shortest paths, then the routing algorithm enters into the vertex-fault mode. In vertex-fault mode, the message includes the following entries:
\[
(\alpha : \text{type of fault}, d : \text{destination vertex}, \\
f : \text{faulty vertex}, m : \text{message}).
\]

The algorithm also needs to know the four routing parameters \( P, Q, S, \) and \( T \). According to the above lemmas, the routing algorithm for each vertex \( u \) in \( G(n; h_1, h_2; w_1, w_2) \) is described as follows. Note that Algorithm B is called only when the algorithm enters into the faulty-mode and vertex \( u \) wants to determine through which edge to send the message.

**Algorithm B**

**Input:** A double-loop network \( G(n; h_1, h_2; w_1, w_2) \) and its four routing parameters \( P, Q, S, T \), type of fault \( \alpha \), destination vertex \( d \), faulty vertex \( f \) and message \( m \).

**Output:** An edge \( e \) through which the message is sent by vertex \( u \).

**Method:**

**Step 1.** Use \( \alpha \) to determine the faulty type. Steps 2 and 3 process type-1-fault and Steps 4 and 5 process type-2-fault.

**Step 2.** Compute \( x_0, x_1, \) and \( y \) such that
\[
h_{1x0} \equiv d - u (\mod n) \\
h_{1x1} \equiv f - u (\mod n) \\
h_{2y} \equiv d - u (\mod n).
\]

**Step 3.** If one of the following conditions is satisfied:
1. \( u = s \),
2. \( Q \geq 2 \) and \( y < T \),
3. \( Q = 1, T > Q, \) and \( (x_0 < S \text{ or } y < T) \).
4. \( Q = 1 \) and \( (x_0 \geq S) \) and \( x_0 \leq S - P \).

then \( e = (u, u + h_2) \),
else \( e = (u, u + h_1) \).

**Stop.**

**Step 4.** Compute \( y_0, y_1, \) and \( x \) such that
\[
h_{2y0} \equiv d - u (\mod n) \\
h_{2y1} \equiv f - u (\mod n) \\
h_{1x} \equiv d - u (\mod n).
\]
Step 5. If one of the following conditions is satisfied:
1. \( u = s \),
2. \( P \geq 2 \) and \( x < S \),
3. \( P = 1 \), \( S > P \), and \( (y_0 < T \) or \( x < S \)\),
4. \( S = P = 1 \) and \( (y \geq y_1 \) and \( y_0 \leq T - Q \)\),
then \( e = (u, u + h_1) \).
else \( e = (u, u + h_2) \).
Stop.

End of Algorithm B

Note that \( x_0 \) (respectively, \( x_1 \)) is the number of edges in the path from \( u \) to \( d \) (respectively, \( u \) to \( f \)) that only uses edges in \( E_1 \), \( y \) (respectively, \( x \)) is the number of edges in the path from \( u \) to \( d \) (respectively, \( u \) to \( f \)) that only uses edges in \( E_2 \), \( y_0 \) (respectively, \( y_1 \)) is the number of edges in the path from \( u \) to \( d \) (respectively, \( u \) to \( f \)) that only uses edges in \( E_2 \), and, if no solution exists, then its value is \( \infty \).

We explain the meaning of the four conditions in Step 3 of Algorithm B as follows: In Condition 1, if \( u = s \), then \( f \) is \( u \)'s right neighbor in the \( D \)-region. Thus, edge \( (u, u + h_1) \) is the only edge which can be used. Condition 2 routes the message along the path described in the proof of Lemma 1. Conditions 3 and 4 route the message along the path described in the proof of Lemma 4.

We use the graph \( G(14; 4, 3, 1.2) \) as an example to illustrate Algorithm B. The four routing parameters \( P, Q, S, \) and \( T \) of an \( L \)-region are 2, 2, 5, and 4, respectively. Let \( s, d, \) and \( f \) be vertices 0, 8, and 4, respectively. Initially, \( u = s = 0 \) and \( e \) is the edge \( (0, 3) \). Then, \( u = 3 \). In Step 2, \( x_0, x_1, \) and \( y \) are equal to \( \infty, \infty, \) and \( 11 \) respectively. In Step 3, none of the four conditions is satisfied. Thus, \( e \) is the edge \( (3, 7) \). In a similar way, we can find the edges \( (7, 11), (11, 1), \) and \( (1.5) \) when \( u = 7, 11, \) and 1, respectively. Now, we consider \( u = 5 \). In Step 2, \( x_0, x_1, \) and \( y \) are equal to \( \infty, \infty, \) and \( 1 \), respectively. In Step 3, Condition 2 is satisfied. Thus, the output edge is \( (5, 8) \). And, the message is sent to its destination vertex 8.

Now, we consider the edge-fault mode. Let \( e \) be the faulty edge which is between vertices \( s \) and \( v \), where \( s \) is the first vertex finding the faulty edge. This faulty mode can be handled almost the same as the vertex-fault mode. Since no faulty vertex exists, the destination vertex can be any vertex in a double-loop network. We only need to replace the faulty vertex \( f \) in Algorithm B by \( v \). It will work correctly for the edge-fault mode.

By Lemmas 1, 4, and 5 the following theorem follows directly.

Theorem 1. Algorithm B finds a shortest path for the source vertex to the destination vertex after a faulty vertex or a faulty edge is detected.

4 Concluding Remarks

In this paper, an optimal fault-tolerant message routing algorithm is presented. Assume that there is at most one faulty element in the network and the faulty element is not known in advance. The algorithm provides a mechanism to send messages along a shortest path when it detects the faulty element. Double-loop networks have many interesting properties. Although many researchers have devoted their research efforts to it in recent years, many problems still remain open. For example, whether there exist efficient message routing algorithms without routing tables for the undirected version of the double-loop network is still unknown.

References


Yu-Liang Liu received both the BS and MS degrees in information management from the National Taiwan University of Science and Technology, Taiwan, Republic of China, in 1997 and 1999, respectively. He is presently a PhD student in the Department of Information Management, National Taiwan University. His research interests include interconnection networks, graph theory, and computer algorithms.

Yu-Li Wang received the BS and MS degrees from Tamkang University, Taiwan, Republic of China, in 1975 and 1979, respectively, and the PhD degree from National Tsing Hua University, Taiwan, in 1988, all in computer science. He joined the National Taiwan University of Science and Technology in 1990. He served as the chairman of the Department of Information Management, National Taiwan University of Science and Technology from 1988 to 1999. Before joining the National Taiwan University of Science and Technology, he worked at the Advanced Technology Center, Industrial Technology Research Institute. His research interests include graph theory, computational geometry, and the design and analysis of computer algorithms.
D.J. Guan received a diploma in mechanical engineering from the Taipei Institute of Technology in 1971, a BS degree in computer science from Tamkang University in 1979, an MS degree in electrical engineering from National Taiwan University in 1983, and an MS and a PhD degree in computer science from Purdue University in 1986 and 1989, respectively. He has been a faculty member in the Department of Applied Mathematics, National Sun Yat Sen University, since August 1989. During his tenure at the National Sun Yat Sen University, Dr. Guan has served as the director of the Information Science Research Center and the chairman of the Department of Applied Mathematics. He is now a professor in the Department of Computer Science, National Sun Yat Sen University. He is a member of the Chinese Cryptology and Information Security Association. His current research interests include algorithms, combinatorics, and cryptography.

IEEE Computer Society publications cited in this article can be found in our Digital Library at http://computer.org/publications/dlib.