

# Advanced Data Security and Its Applications in Multimedia for Secure Communication

Lead Guest Editor: Ki-Hyun Jung

Guest Editors: Mehdi Hussain, Rajkumar Soundrapandiyan, and Thai-Son Nguyen





---

# **Advanced Data Security and Its Applications in Multimedia for Secure Communication**

Security and Communication Networks

---

**Advanced Data Security and Its  
Applications in Multimedia for Secure  
Communication**

Lead Guest Editor: Ki-Hyun Jung

Guest Editors: Mehdi Hussain, Rajkumar  
Soundrapandiyan, and Thai-Son Nguyen



---





Copyright © 2020 Hindawi Limited. All rights reserved.

This is a special issue published in "Security and Communication Networks." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

# Chief Editor

Roberto Di Pietro, Saudi Arabia

## Associate Editors

Jiankun Hu , Australia  
Emanuele Maiorana , Italy  
David Megias , Spain  
Zheng Yan , China

## Academic Editors

Saed Saleh Al Rabae , United Arab Emirates  
Shadab Alam, Saudi Arabia  
Goutham Reddy Alavalapati , USA  
Jehad Ali , Republic of Korea  
Jehad Ali, Saint Vincent and the Grenadines  
Benjamin Aziz , United Kingdom  
Taimur Bakhshi , United Kingdom  
Spiridon Bakiras , Qatar  
Musa Balta, Turkey  
Jin Wook Byun , Republic of Korea  
Bruno Carpentieri , Italy  
Luigi Catuogno , Italy  
Ricardo Chaves , Portugal  
Chien-Ming Chen , China  
Tom Chen , United Kingdom  
Stelvio Cimato , Italy  
Vincenzo Conti , Italy  
Luigi Coppolino , Italy  
Salvatore D'Antonio , Italy  
Juhriyansyah Dalle, Indonesia  
Alfredo De Santis, Italy  
Angel M. Del Rey , Spain  
Roberto Di Pietro , France  
Wenxiu Ding , China  
Nicola Dragoni , Denmark  
Wei Feng , China  
Carmen Fernandez-Gago, Spain  
AnMin Fu , China  
Clemente Galdi , Italy  
Dimitrios Geneiatakis , Italy  
Muhammad A. Gondal , Oman  
Francesco Gringoli , Italy  
Biao Han , China  
Jinguang Han , China  
Khizar Hayat, Oman  
Azeem Irshad, Pakistan


M.A. Jabbar , India  
Minho Jo , Republic of Korea  
Arijit Karati , Taiwan  
ASM Kayes , Australia  
Farrukh Aslam Khan , Saudi Arabia  
Fazlullah Khan , Pakistan  
Kiseon Kim , Republic of Korea  
Mehmet Zeki Konyar, Turkey  
Sanjeev Kumar, USA  
Hyun Kwon, Republic of Korea  
Maryline Laurent , France  
Jegatha Deborah Lazarus , India  
Huaizhi Li , USA  
Jiguo Li , China  
Xueqin Liang, Finland  
Zhe Liu, Canada  
Guangchi Liu , USA  
Flavio Lombardi , Italy  
Yang Lu, China  
Vincente Martin, Spain  
Weizhi Meng , Denmark  
Andrea Michienzi , Italy  
Laura Mongioi , Italy  
Raul Monroy , Mexico  
Naghme Moradpoor , United Kingdom  
Leonardo Mostarda , Italy  
Mohamed Nassar , Lebanon  
Qiang Ni, United Kingdom  
Mahmood Niazi , Saudi Arabia  
Vincent O. Nyangaresi, Kenya  
Lu Ou , China  
Hyun-A Park, Republic of Korea  
A. Peinado , Spain  
Gerardo Pelosi , Italy  
Gregorio Martinez Perez , Spain  
Pedro Peris-Lopez , Spain  
Carla Ràfols, Germany  
Francesco Regazzoni, Switzerland  
Abdalhossein Rezai , Iran  
Helena Rifà-Pous , Spain  
Arun Kumar Sangaiah, India  
Nadeem Sarwar, Pakistan  
Neetesh Saxena, United Kingdom  
Savio Sciancalepore , The Netherlands

De Rosal Ignatius Moses Setiadi ,  
Indonesia  
Wenbo Shi, China  
Ghanshyam Singh , South Africa  
Vasco Soares, Portugal  
Salvatore Sorce , Italy  
Abdulhamit Subasi, Saudi Arabia  
Zhiyuan Tan , United Kingdom  
Keke Tang , China  
Je Sen Teh , Australia  
Bohui Wang, China  
Guojun Wang, China  
Jinwei Wang , China  
Qichun Wang , China  
Hu Xiong , China  
Chang Xu , China  
Xuehu Yan , China  
Anjia Yang , China  
Jiachen Yang , China  
Yu Yao , China  
Yinghui Ye, China  
Kuo-Hui Yeh , Taiwan  
Yong Yu , China  
Xiaohui Yuan , USA  
Sherali Zeadally, USA  
Leo Y. Zhang, Australia  
Tao Zhang, China  
Youwen Zhu , China  
Zhengyu Zhu , China

# Contents




---

## **Protecting Metadata of Access Indicator and Region of Interests for Image Files**

JeongYeon Kim 

Research Article (10 pages), Article ID 4836109, Volume 2020 (2020)

## **An Improved Bidirectional Shift-Based Reversible Data Hiding Scheme Using Double-Way Prediction Strategy**

Lin Li , Chin-Chen Chang , and Hefeng Chen 

Research Article (17 pages), Article ID 3031506, Volume 2019 (2019)

## **Using XGBoost to Discover Infected Hosts Based on HTTP Traffic**

Weina Niu , Ting Li, Xiaosong Zhang , Teng Hu , Tianyu Jiang, and Heng Wu

Research Article (11 pages), Article ID 2182615, Volume 2019 (2019)

## **VPN Traffic Detection in SSL-Protected Channel**

Muhammad Zain ul Abideen , Shahzad Saleem , and Madiha Ejaz




Research Article (17 pages), Article ID 7924690, Volume 2019 (2019)

## **Outsourcing Hierarchical Threshold Secret Sharing Scheme Based on Reputation**

En Zhang , Jun-Zhe Zhu, Gong-Li Li, Jian Chang, and Yu Li


Research Article (8 pages), Article ID 6989383, Volume 2019 (2019)

## **Generative Reversible Data Hiding by Image-to-Image Translation via GANs**

Zhuo Zhang , Guangyuan Fu, Fuqiang Di , Changlong Li , and Jia Liu



Research Article (10 pages), Article ID 4932782, Volume 2019 (2019)

## **Linear $(t, n)$ Secret Sharing Scheme with Reduced Number of Polynomials**

Kenan Kingsley Phiri and Hyunsung Kim 

Research Article (16 pages), Article ID 5134534, Volume 2019 (2019)

## **Research on Defensive Strategy of Real-Time Price Attack Based on Multiperson Zero-Determinant**

Zhuoqun Xia, Zhenwei Fang, Fengfei Zou, Jin Wang , and Arun Kumar Sangaiah 

Research Article (13 pages), Article ID 6956072, Volume 2019 (2019)

## Research Article

# Protecting Metadata of Access Indicator and Region of Interests for Image Files

**JeongYeon Kim** 

*School of Business, Sangmyung University, 20, Hongjimun 2-gil, Jongno-gu, Seoul 110-743, Republic of Korea*

Correspondence should be addressed to JeongYeon Kim; [jykim@smu.ac.kr](mailto:jykim@smu.ac.kr)

Received 28 April 2019; Accepted 27 December 2019; Published 22 January 2020

Guest Editor: Rajkumar Soundrapandiyan

Copyright © 2020 JeongYeon Kim. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With popularity of social network services, the security and privacy issues over shared contents receive many attentions. Besides, multimedia files have additional concerns of copyright violation or illegal usage to share over communication networks. For image file management, JPEG group develops new image file format to enhance security and privacy features. Adopting a box structure with different application markers, new standards for privacy and security provide a concept of replacement substituting a private part of the original image or metadata with an alternative public data. In this paper, we extend data protection features of new JPEG formats to remote access control as a metadata. By keeping location information of access control data as a metadata in image files, the image owner can allow or deny other's data consumption regardless where the media file is. License issue also can be resolved by applying new access control schemes, and we present how new formats protect commercial image files against unauthorized accesses.

## 1. Introduction

With the increase of data consumption and content creation through the Internet, data security and privacy are one of the major concerns for data sharing. Many technical solutions and frameworks such as network-based protection, access control with authentication, and data encryption have been introduced. However, any of the suggested technologies cannot fully meet all security requirements for various content types and usages. Nowadays, social network services make easier for the Internet users to share multimedia contents rather texting with others. It causes additional security concerns because sensitive personal images or messages shared through social network services could be delivered to strangers [1]. The proliferation of use of digital files brings nonintended release of privacy information, while the owner of the files who posted them on social media only intended to share with limited acquaintances [2]. Image files may contain additional metadata about location, events, and individual relationships between persons in the picture [3], which can be used for secondary hacking.

Moreover, for commercial stocks, the issues of copyright violation and ownership identification get more industry attentions due to frequent occurrences of data abuse and malicious attacks over communication networks. Usually, the attacks are trying to modify or delete specific information in the document to claim legal ownership or proper authorities for content usage.

Addressing these challenges has been an interesting problem for secure communication over open networks. Most multimedia content providers are actively engaged on the preservation of the integrity of their own metadata, by which the Intellectual Property Right (IPR) and the access right information is reserved. Encryption for metadata stored in a specific area named EXIF of an image file helps for the issue [4]. It is not standardized, and popular image decoders cannot support it though.

In this paper, we suggest new metadata management methods based on JPEG standards for Privacy and Security (ISO/IEC JTC 1/SC 29/WG 1/Systems Part4). New JPEG format enhances protection schemes for the sensitive part of the image and important metadata. In addition to adopting new media standards, we suggest to keep access control as a



remote data and store its reference as a metadata in the media file. By using remote access control and keeping its information as a metadata in each image file, the image owner can manipulate the access control data to allow or deny other's data consumption regardless of the place the media file is stored at. License issue also can be resolved by applying new access control schemes. A remote system can decide to allow or deny the given user's access to the media file by the license status under the system login ID. Besides, as users copy a media content file into several networked places, we discuss how new approach protect them against unauthorized accesses.

In the following, we give a description on multimedia-related security issues and ongoing efforts to resolve them. After that, we introduce new JPEG formats and metadata management schemes for security. Also we provide examples of media content, adopting suggested image file formats and explanations on data protection schemes.

## 2. Related Studies

Security should be a part of data management strategy reflecting all users' information usage. The NIST Cybersecurity Framework, a well-known framework used by many business areas and organizations, helps organizations to be proactive about information security risks [5]. To protect the value of data assets, the framework suggests to have content repositories and identify data assets' location first. Each asset has been assigned access permissions for each current and potential user. The framework also suggests to keep important data in an encrypted form preventing it from unauthorized users' accesses even in the data leak cases.

However, media content management needs additional schemes and efforts because the general risk management approaches cannot resolve multimedia specific security issues, such as privacy or license issues. Images should be handled differently from text data for the issues.

*2.1. Security of Multimedia Data.* For required additional functionalities of multimedia contents, MPEG standardization group (ISO/IEC JTC1 SC29/WG11) defines a suite of standards for design and implementation of media-handling features, MPEG-M (ISO/IEC 23006). MPEG-M enables easy design and implementation of media-handling value chains with common APIs, protocols, and interfaces for service aggregation mechanisms.

MIPAMS (Multimedia Information Protection and Management System) [6, 7], a service-oriented content management platform, follows a relevant part of the MPEG-M engines. It includes the rights expression language, license, orchestrator, metadata, content protocol, event reporting, content search, security, intellectual property management, and protection engines. The content licensing scenario is a subsystem for specialized content willing to trade, where content files are distributed under copyright with license templates chosen by users.

MIPAMS implements most required functions with the external system including the license management, which

makes the system complicated. Instead of keeping license data in the media file itself, additional system to keep user's license information needs user and content identification.

Recently, there are attempts to apply Blockchain technology to protect media contents [8]. KODAKOne [9] is one of the examples for image rights management platform. KODAKCoin is a kind of cryptocurrencies, which can be used to buy the license of images in KODAKOne, enabling photographers to take more control in image rights management. It is a digital ledger of rights ownership for photographers' works. Photographers upload their images to the platform and the records of license purchases are recorded in the ledger. Referring to public Blockchain records, everyone can check if a certain user has a license for digital assets. It enables to track licensing records and prevent illegal uses. However, Blockchain technology is in an early stage for development having limited capacities and not fully defined details, including general identification methods for users and digital assets to records license transactions in the ledger.

*2.2. JPEG Privacy and Security.* As we have reviewed, privacy and license issues of digital assets are not well addressed and it is an inhibiting factor in the further digital content distribution. To resolve the issues, the JPEG group decided to develop another image file format to enhance security and privacy features [10].

The JPEG format is one of widely used multimedia standards. After the file format of JPEG known as JPEG Interchange Format (JIF), additional standards have evolved. JPEG File Interchange Format (JFIF) and Exchangeable image file format (Exif), both formats use JIF byte layout employing the application markers which is one of the JIF standard's extension points [11]. JFIF uses APP0, and Exif uses APP1.

JPEG XT (ISO/IEC 18477-3 for JPEG-1/JPEG XT) defines a file format to embed boxes in a basic structure of a JPEG-1 image file in order to achieve a common framework for future extensions across JPEG standards. With this box structure, future standards can focus on the definition of boxes and provide compliances across the family of JPEG standards [12].

Adopting these schemes, the JPEG working group suggested a new file format for privacy and security features employing box structure with different application markers. The purpose of the new standards is not suggesting new image coding methods based on better mathematical model, but rather additional boxes for metadata useful for security and privacy in an image file. By using different markers for added boxes, current image decoders just skip new box formats while new image decoders can support new features. Some image codes can be stored in the boxes for metadata, but they can be decoded as the existing image data stored in other areas. Figure 1 shows the compatibility between current JPEG decoders and new decoders.

JPEG privacy and security format suggest concept of replacement as a main method of image privacy protection [13–15]. Replacement means the private part of the image or

metadata can be substituted by given public data. The image owner decides the sensitive area of the image and replaces the original image with the public one. As a result of image replacement, the sensitive data is stored in a created replacement box to substitute data stored in the target area where public data is placed. When image decoder encounters a replacement box in the image file, the associated replacement action should be applied and the resulting file is decoded as usual if the user has a proper authorization for the file. The standards define four types of replacements: box structure in same file, app marker segment, region of interest in image data, and whole file.

Figure 2 shows box replacement, region of interest replacement, and file replacement cases. In a box replacement case, byte offset of the target box counting from the beginning of the file should be provided. In a ROI replacement case, start position and end position of the ROI should be provided. Current standards define the position in the image with vertical offset from the top in pixels and horizontal offset from the left in pixels.

The Privacy and Security standards define the additional method using data encryption to secure data from an unauthorized user. The replacement method has sensitive data in its box format and prevents unauthorized users from accessing it with image decoders, but data should be encrypted to avoid additional accesses.

As described in Figure 3, encrypted data is also stored in a box structure. Encryption related parameters, such as encryption method and initial vector used for key generation, should be stored in the box. The protected content shall be decrypted using provided parameters if the encryption method is supported by the decoder and authorization is granted. Otherwise, the entire protected content shall be ignored.

2.3. *JUMBF (JPEG Universal Metadata Box Format)*. For described protection methods in Section 2.2, new standards guide to keep security-related data as metadata and to use JUMBF (JPEG Universal Metadata Box Format), new box format, for the metadata [16]. The JUMBF box contains exactly one JUMBF description box and one or more content Boxes. According to the definitions of the box structure, a box contains other boxes in it and is called a super box. JUMBF is a kind of super box and also can be nested, but in its first place, there is always a description box.

The type of content Boxes is implied by the JUMBF TYPE field in the JUMBF description box. In the standards, there are several predefined JUMBF types with exactly one content box. Table 1 has the currently defined JUMBF TYPE with one content box in the JUMBF super box. A XML or JSON-type JUMBF box has just one content box, where XML or JSON format text data is containing. Also code stream-type JUMBF contains just one content box having image code stream data, while UUID-type JUMBF contains a box containing box identification data to be referred to from inside or outside of the image file. If JUMBF has more than one content box, it should be defined in related specifications.

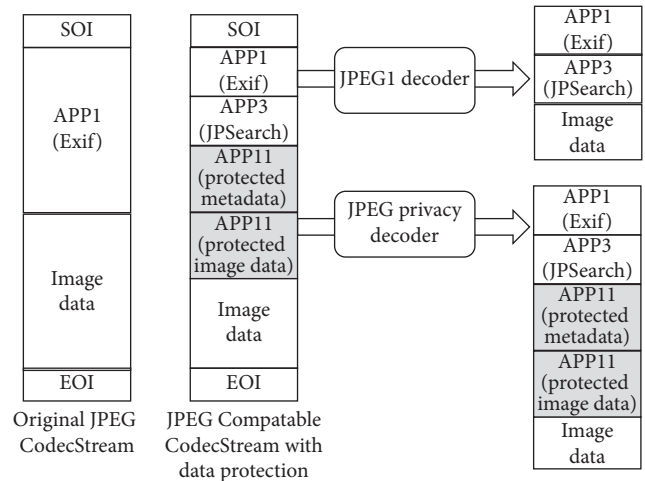


FIGURE 1: JPEG privacy and security requirements.

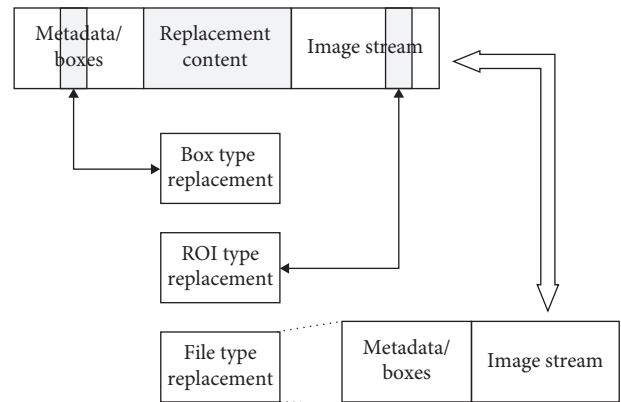


FIGURE 2: Box structure for replacement methods.

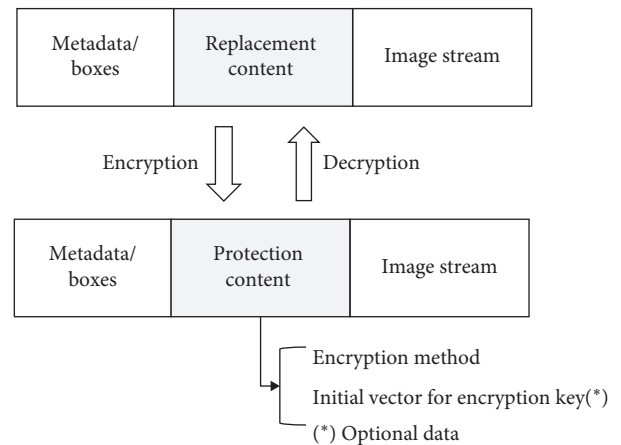


FIGURE 3: Box structure for protection methods.

Replacement and Protect methods also keep the related data in image files as metadata using predefined super boxes, called replacement-type JUMBF and protection-type JUMBF. As described in Figure 4, the replacement-type and protection-type JUMBF boxes have a JUMBF description box, their own description box, and data boxes. Data for

TABLE 1: JUMBF content types.

JUMBF type	Meaning
Code stream content type	Exactly one codestream box.
XML content type	Exactly one XML box.
JSON content type	Exactly one JSON box.
UUID content type	Exactly one UUID box.
Other content types	Other content types defined in other specifications

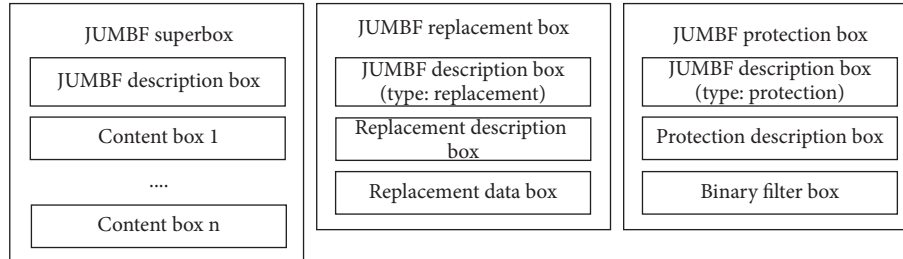


FIGURE 4: JUMBF superbox (replacement/protection box).

replacement such as target area offsets or data for encryption such as encryption method with an initial vector should be placed in the own description box, while real image data for original images or encrypted data is stored in each data box.

Besides, the protection JUMBF box has several labels referring to another JUMBF boxes just in case more data are required. Figure 5 shows additional labels in the JUMBF protection box.

ENC label is an optional label referencing to a JUMBF Box used to keep additional data encryption-related parameters. There are many encryption methods and some of them may need more parameters. The encoder creates an additional box to store them and record its label in the ENC label space. Image decoders find the parameters by following the label. The AR label is also an optional label referencing to a JUMBF Box having access policy rules of the data in the Binary Data box. Besides, the Initial Vector label is an optional space for an initial value used to start encryption key related process.

### 3. Image Protection

With conversions of current JPEG image files into new format, sensitive image data such as Personal Identifiable Information (PII) can be hidden by overlapping the related image area with another images. Metadata in images also can be handled as important data by using protection JUMBF boxes.

In this section, we check resolve image license issues applying access control information protected as metadata. Also we take sample images from a movie titled “Hana Restaurant” and explain how to create access control data within a new image format.

*3.1. Metadata for Access Control.* To admit user’s access to certain data, an application program will check user authentication and verify user authorization for the given local

resource. Usually access permissions of the resource are generated by a policy or rules which depend on the group the user belongs to. Access rules may be expressed using eXtensible Access Control Markup Language (XACML).

New image format reserves a space of the AR (Access Rule) label in the protection JUMBF box linking to another JUMBF box keeping the actual XACML data. In our examples, we will provide location information of the XACML instead of the data to allow image owners to manage it wherever the actual content file is.

Figure 6 gives an example for using the XML JUMBF box linked to a Protection Box. With XACML in the XML box, the system can provide policy-based access control to image files. The following is an example of XACML policy for date-based access control. It permits any user’s view action to an image file named “Sample.jpg” before the end of year 2019. The policy or policy sets defined in XACML 3.0 may have remote references Algorithm 1.

Also, encrypted data can be decrypted by user authorization. Otherwise, the image owner can assign a password. The ENC Label in Figure 5 links to another JUMBF box having parameters for data decryption. The following is an example using the JSON box for the encryption parameters Algorithm 2.

If the encryption method in the linked JUMBF box has password-based encryption, the system will generate a secret key based on Password-based Key Derived Function (PBKDF) [17] to use for data encryption and decryption. The users who provide proper password can decrypt images.

Also access control can be performed with user authorization. The user should provide additional login information to the external image management system, where it keeps image identification and each user’s authorization of the image. The system allows or denies a given user’s access to images through policies. XACML has a request and response form for resource accesses or a reference to remote policy to get user’s current authorization.

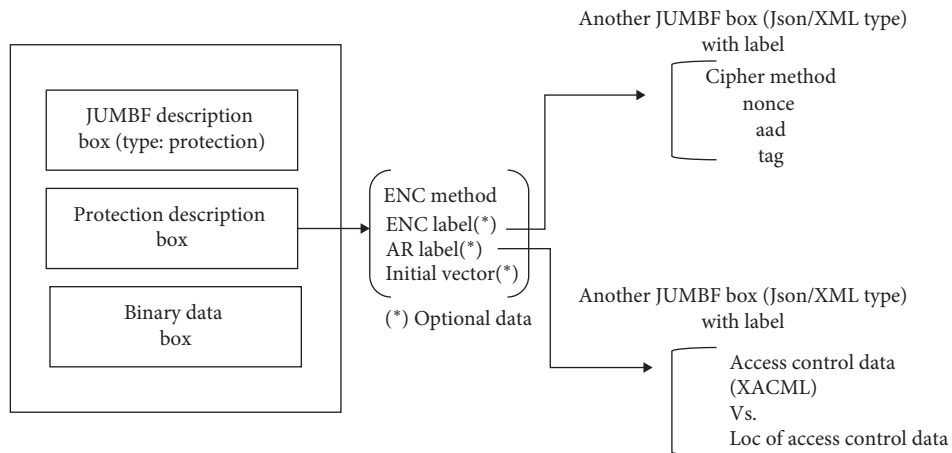


FIGURE 5: Protection JUMBF box for encrypted document.

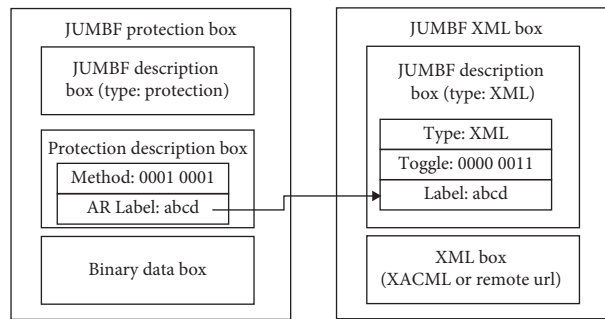


FIGURE 6: Connection to XML box for access control.

```

<Policy xmlns = "urn:oasis:names:tc:XACML:3.0:core:schema:wd-17"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
PolicyId = "urn:isdcm:policyid:1"
RuleCombiningAlgId = "urn:oasis:names:tc:XACML:1.0:rule-combining-algorithm:first-applicable"
Version = "1.0"
xsi:schemaLocation = "urn:oasis:names:tc:XACML:3.0:core:schema:wd-17
http://docs.oasis-open.org/XACML/3.0/XACML-core-v3-schema-wd-17.xsd">
<Description> Desert.jpg </Description>
<Rule Effect = "Permit" RuleId = "urn:oasis:names:tc:XACML:2.0:ejemplo:Desert">
<Description> Any user can view urn:mimage:Desert.jpg before the end of the year 2019 </Description>
.....
<!--resource-->
<Match MatchId = "urn:oasis:names:tc:XACML:1.0:function:regexp-string-match">
<AttributeValue
DataType = "http://www.w3.org/2001/XMLSchema#string"> urn:mimage:Sample.jpg </AttributeValue>
<AttributeDesignator
AttributeId = "urn:oasis:names:tc:XACML:1.0:resource:resource-id"
Category = "urn:oasis:names:tc:XACML:3.0:attribute-category:resource"
DataType = "http://www.w3.org/2001/XMLSchema#string" MustBePresent = "false"/>
</Match>
.....
<!--action-->
<Match MatchId = "urn:oasis:names:tc:XACML:1.0:function:string-equal">
<AttributeValue
DataType = "http://www.w3.org/2001/XMLSchema#string">
View </AttributeValue>
    
```

```

<AttributeDesignator
AttributeId = "urn:oasis:names:tc:XACML:1.0:action:action-id"
Category = "urn:oasis:names:tc:XACML:3.0:attribute-category:action"
DataType = "http://www.w3.org/2001/XMLSchema#string"
MustBePresent = "false"/>
</Match>
.....
<Condition>
<Apply FunctionId = "urn:oasis:names:tc:XACML:1.0:function:date-less-than-or-equal">
<Apply FunctionId = "urn:oasis:names:tc:XACML:1.0:function:date-one-and-only">
<AttributeDesignator AttributeId = "accessDate"
Category = "urn:oasis:names:tc:XACML:3.0:date"
DataType = "http://www.w3.org/2001/XMLSchema#date" MustBePresent = "false"/> </Apply>
<AttributeValue DataType = "http://www.w3.org/2001/XMLSchema#date"> 2019-12-31 </AttributeValue>
</Apply>
</Condition>
.....
</policy>

```

ALGORITHM 1

```

{
  "jpeg_security": {
    "type": "protection",
    "cipher": {
      "method": "AES256-GCM",
      "nonce": "BdZbHABY/sytDTUB",
      "aad": "ZmFzb28uY29t",
      "tag": "1dsCuZ5XuanojwM/p6EoCA == "
    }
  }
}

```

ALGORITHM 2

3.2. *Metadata for Licensing.* With a new image file format, we can manage the license information as a part of access control data or as another policy. It is much easier to manage license issues by access control compared to managing them by metadata, especially for several image copies in distributed environments. If license data is stored as metadata and there are several same files in networks, the system has to update each license metadata as user's authorization changes. However, keeping it as an access policy could be useful to identify the location of the same content files in networks.

In our examples, we integrate license data into access control data and keep the reference of access control policy in a XML JUMBF box.

3.3. *Examples.* Image format conversion process to generate new protected image has a sequence summarized in Figure 7. In this example, converting sample images, we assume that the image owner selects replacement methods and afterward data encryption methods.

Starting from the original image, we proceed through the following steps:

- (1) Select partial areas of the original image and alternative public images to substitute the selected areas. The alternative images should have exactly the same size and dimensions as the selected partial areas of the original image.
- (2) Create alternative images open to public by replacing selected partial areas of the original image with given alternative images. The original image data is stored in a replacement JUMBF box.
- (3) If needed, the replacement JUMBF box can be encrypted and stored as a protection JUMBF box. The protection JUMBF box may have labels for additional JUMBF box, where encryption parameters or access control policy or rules are stored.

For the sample image, we identify facial images as sensitive areas, and they are replaced with a smile image. In our Windows program shown in Figure 8, the user provides input

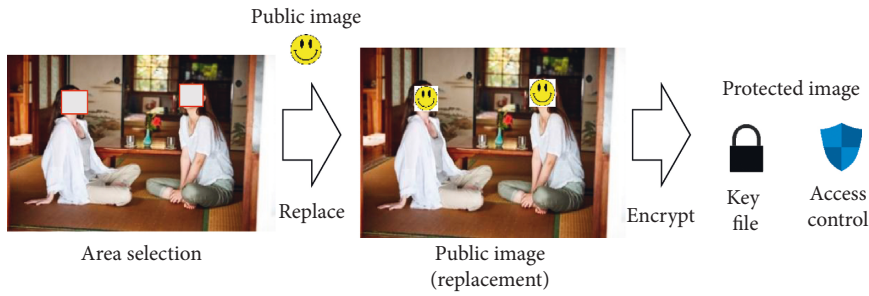


FIGURE 7: Protection JUMBF box for encrypted document (public images from Korean movie titled “Hana Restaurant” 2018).

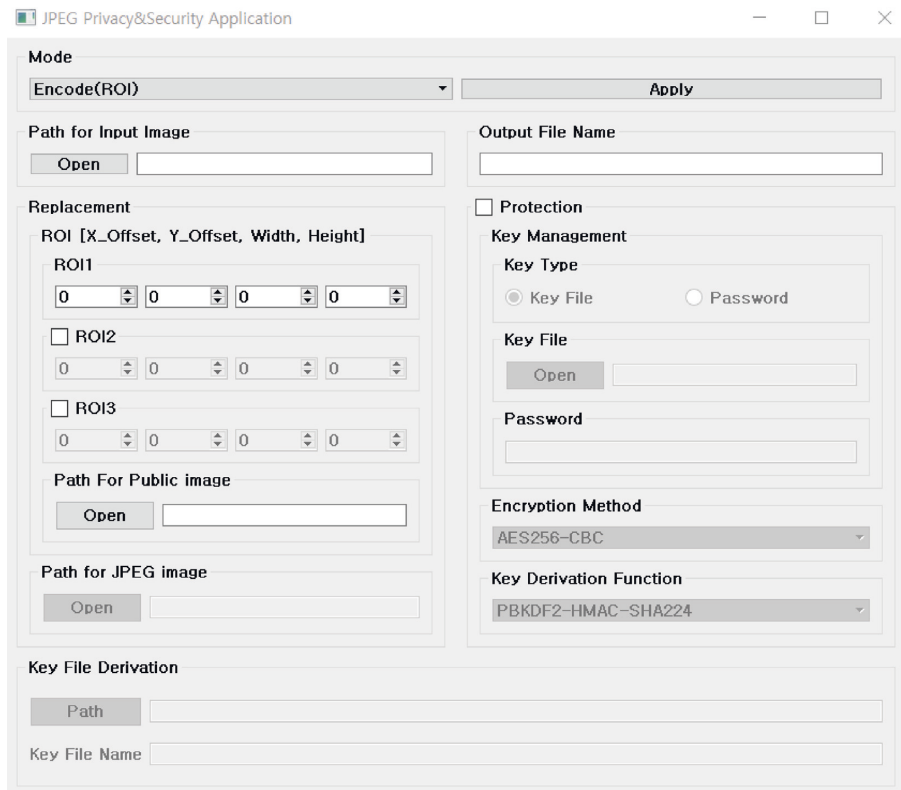


FIGURE 8: Application for replacement and protection JUMBF box.

file path, output file path, alternative image path, and ROI offsets to be replaced. After the file conversion, area’s offsets and partial images are stored in the replacement JUMBF box, as shown in Figure 9.

After the creation of replacement JUMBF boxes, it can be encrypted. Assigned access control data and encryption related data can be stored in separate JUMBF boxes, and the protection JUMBF box keep their labels in the specific area. In this example, new image file has a XML JUMBF box having the location of access rule information, as shown in Figure 10.

As we explained in Section 3.2, the user will be a member of the licensed user group if the user has a license and the group has an authority to use the content.

Decoding the converted image files proceeds as follows:

- (1) During sequential file scanning, a protection JUMBF box is decrypted for users who have the appropriate rights for the file. Otherwise, skip the box.
- (2) If the decrypted content is a replacement JUMBF box, original images are merged at the position determined by offsets  $x$  and  $y$ .
- (3) The resulting image file will be decoded as usual.

With described processes, users with proper rights can access to the original image and others access to alternative security images. We applied replacement and encryption methods with a password to public 44 images of a movie titled “Hana Restaurant” available from a Korean portal site. Figure 11 shows the selected results. Login-based authorization needs the additional management system, and we

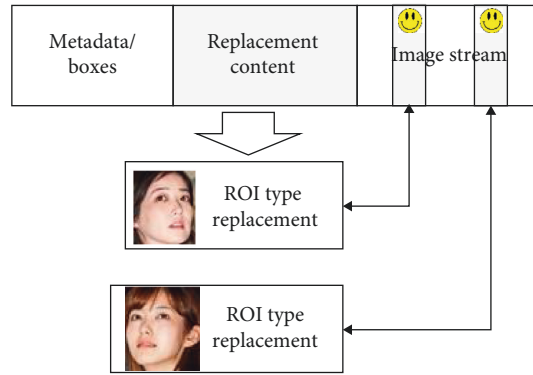


FIGURE 9: Replacement JUMBF box for ROI (public images from Korean movie titled “Hana Restaurant” 2018).

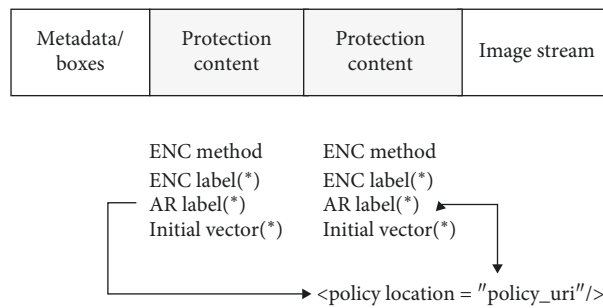


FIGURE 10: Protection JUMBF box for encrypted document.



FIGURE 11: Authorized view with password vs. public view (public images from Korean movie titled “Hana Restaurant” 2018).

encrypted the replacement boxes with the password-based method to simplify the access control.

Image encoding and decoding time varies depending on images’ count of replacements and the size for encryption data. For images with 2 replacements and encryptions, the

average encoding time of replacement is 0.35 seconds while password-based encryption takes 82% more time. Figure 12 shows the time ratio of encoding and decoding time between the original image, images with replacement boxes, images with password-based encryption boxes, and images with

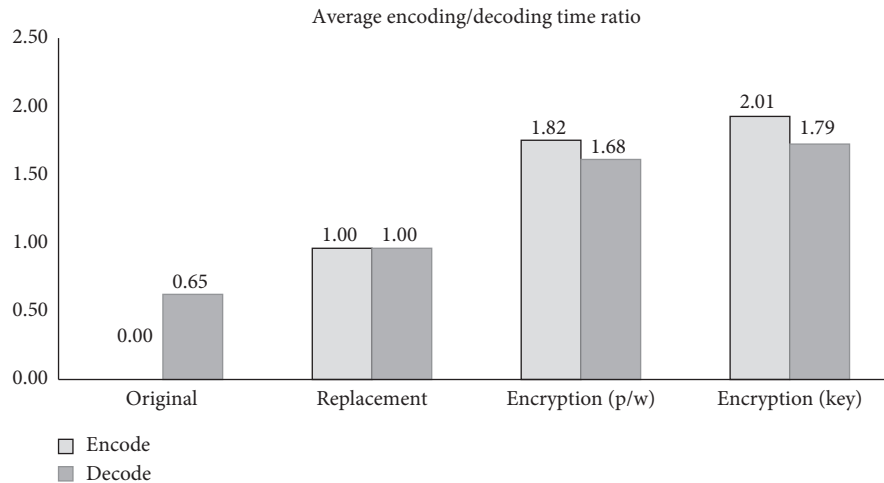


FIGURE 12: Performance of encoding and decoding.

encryption based on AES256-CBC algorithm with the initial vector.

#### 4. Conclusion

We reviewed image protection methods for region of interests and various metadata suggested in JPEG privacy and security standards. Suggested methods provide a mechanism to hide sensitive image parts of personal identifiable areas by overlapping other images. Compared to other JPEG standards for secure images, Secure JPEG 2000 (JPSEC), new standards are focusing on metadata and using JPEG XT box formats instead of suggesting new code stream methods. The JPSEC standards' specification targets protection of the entire code stream or segments of JPEG 2000 coded data and additional syntax to specify associated protection methods. However, new standards allow XML or JSON-type data and even code stream in the metadata areas and provide more flexibility applying additional security features such as XACML policies. Important metadata such as payment or licensing data also can be applied for access control policies.

We provide examples of JPEG image protection and privacy enhancement features using new standards' replacement and protection methods. The personal identifiable information in images can be replaced with public subimages. After deciding image areas to replace with another images, the image owner manages access control data by the following steps:

- (i) Placing reference of access control policy in each image file
- (ii) Placing and encrypting reference information of license policy in each image file if the original image has one
- (iii) Deciding default access allowance for each image file

As a further study, we are planning to review other standards for multimedia data to cooperate with image

metadata. Multimedia blockchain with smart contract functions is a platform to support various multimedia data.

#### Data Availability

No data were used to support this study.

#### Conflicts of Interest

The authors declare that they have no conflicts of interest.

#### Acknowledgments

This research was supported by a 2018 Research Grant from Sangmyung University.

#### References

- [1] M. Blaze, John Ioannidis, and A. Keromytis, "Experience with the keynote trust management system: applications and future directions," p. 1071, Trust Management, Berlin, Germany, 2003.
- [2] T. Jing, Q. Chen, and Y. Wen, "A probabilistic privacy preserving strategy for word-of-mouth social networks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 6031715, 12 pages, 2018.
- [3] Y. Liu, W. Zhang, and N. Yu, "Protecting privacy in shared photos via adversarial examples based stealth," *Security and Communication Networks*, vol. 2017, Article ID 1897438, 15 pages, 2017.
- [4] H. Wijayanto, I. Riadi, and Y. Prayudi, "Encryption EXIF metadata for protection photographic image of copyright piracy," *International Journal of Research in Computer and Communication Technology*, vol. 5, pp. 237–242, 2016.
- [5] J. Guinn, *Why You Should Adopt the NIST Cybersecurity Framework*, PwC, London, UK, 2015.
- [6] S. Llorente, E. Rodriguez, J. Delgado, and V. Torres-Padrosa, "Standards-based architectures for content management," *IEEE MultiMedia*, vol. 20, no. 4, pp. 62–72, 2013.
- [7] J. Delgado, S. Llorente, and E. Rodriguez, "Digital rights and privacy policies management as a service," in *Proceedings of the 2012 IEEE Consumer Communications and Networking Conference (CCNC)*, IEEE, Las Vegas, NV, USA, January 2012.



- [8] D. Bhowmik and F. Tian, "The multimedia blockchain: a distributed and tamper-proof media transaction framework," in *Proceedings of the 2017 22nd International Conference on Digital Signal Processing (DSP)*, IEEE, London, UK, August 2017.
- [9] I. Wenn Digital, *KODAKOne|Image Rights Management Platform*, WENN Digital, Inc., Los Angel, CA, USA, 2008, <https://kodakone.com/>.
- [10] F. Temmermans, T. Ebrahimi, S. Foessel et al., "JPEG privacy and security framework for social networking and glam services," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, p. 68, 2017.
- [11] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. 18–34, 1992.
- [12] T. Richter, "On the standardization of the JPEG XT image compression," in *Proceedings of the 2013 Picture Coding Symposium (PCS)*, IEEE, San Jose, CA, USA, December 2013.
- [13] P. Schelkens, "Image security tools for JPEG standards," in *Proceedings of the 2nd ACM Workshop on Information Hiding and Multimedia Security*, ACM, Salzburg, Austria, June 2014.
- [14] L. Yuan, P. Korshunov, and T. Ebrahimi, "Privacy-preserving photo sharing based on a secure JPEG," in *Proceedings of the 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs)*, IEEE, Hong Kong, China, May 2015.
- [15] L. Yuan, P. Korshunov, and T. Ebrahimi, "Secure JPEG scrambling enabling privacy in photo sharing," in *Proceedings of the 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Ljubljana, Slovenia, May 2015.
- [16] A. Kuzma, F. Temmermans, and T. Richter, "Emerging image metadata standards activities in JPEG," in *Proceedings of the Applications of Digital Image Processing XLI International Society for Optics and Photonics*, vol. 10752, San Diego, CA, USA, August 2018.
- [17] K. Moriarty, B. Kaliski, and A. Rusch, "PKCS# 5: password-based cryptography specification version 2.1," 2017, RFC 8018, <https://www.rfc-editor.org/info/rfc8018>.

## Research Article

# An Improved Bidirectional Shift-Based Reversible Data Hiding Scheme Using Double-Way Prediction Strategy

Lin Li <sup>1,2</sup> Chin-Chen Chang <sup>2,3</sup> and Hefeng Chen <sup>1</sup>

<sup>1</sup>Computer Engineering College, Ji Mei University, Xiamen 361021, China

<sup>2</sup>Department of Information Engineering and Computer Science, Feng Chia University, 100 Wenhwa Road, Seatwen, Taichung 40724, Taiwan

<sup>3</sup>School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

Correspondence should be addressed to Chin-Chen Chang; alan3c@gmail.com

Received 16 April 2019; Revised 16 October 2019; Accepted 12 November 2019; Published 26 December 2019

Guest Editor: Ki-Hyun Jung

Copyright © 2019 Lin Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Reversible data hiding (RDH) is a method that allows a cover image to be completely recovered from its corresponding stego image without distortion after the embedded secret messages have been extracted. Prediction-error expansion (PEE), as a classic RDH scheme, has been studied extensively due to its high quality of stego images. Based on prediction errors, threshold values, and the relative distances between each bin and zero bin, we present a bidirectional shift and double-way prediction strategy to solve the multiple embedding problem. Compared with the original algorithm, this scheme only takes a little more time and reduces the PSNR slightly, but it improves the embedding capacity significantly and allows for reversible data hiding. When both threshold values of  $TH$  and  $TH^*$  are equal to 2, the average  $ER$  value of 108 test images is 1.2 bpp which is ideal for medium data payload. At the same time, the PSNR is above 30 dB, making embedded information visually imperceptible. These data, together with other experimental results, show that the method proposed in this paper has obvious advantages in image quality and embedding capacity.

## 1. Introduction

The concept of steganography was first recorded and used in 1499 by Johannes Trithemius in his *Steganographia*, which was placed on the Index Librorum Prohibitorum in 1609 [1] and removed in 1900 [2], but the practice of secret communication has existed since ancient times. There are stories from Roman times concerning how slaves could hide messages and send the messages to recipients even though they were closely watched. Currently, in the computer age, camouflaged information is still processed to make it unrecognizable to the human eye, but it can possibly be revealed by computer visual recognition. In recent years, with the increasing emphasis on information security, data hiding technology has been used extensively for copyright protection and the authentication of the content of digital multimedia [3–5]. The content of an original image is altered after some data have been embedded in the image, and the

changes are visually imperceptible. But there is an inherent shortcoming of the traditional data hiding scheme, which is that the cover image cannot be restored completely after the receiver extracts the secret information [6–8]. For some specific scenarios, such as military and medical information and the preservation of artwork, even slight distortions in the images are intolerable since it is always essential to recover the original raw data. Therefore, reversible data hiding techniques have been proposed and developed that enable the receiver to extract the secret information, as is done in the traditional scheme, as well as to reconstruct the entire original image without distortion. From the perspective of its application, RDH can be considered a covert communication channel for the transmission of secret information. Thus, it can be seen that its application is quite extensive, such as embedding patients' private information into corresponding medical images and providing lossless authentication watermark for satellite images.

The first reversible data hiding (RDH) scheme was proposed in 1997 by Barton in his patent [9]. Barton's scheme compresses some of the alternative superimposed bits, adds a stream of bits, and embeds them into data blocks. Since then, RDH has attracted the attention of many scholars, and related works have been published frequently. In 2001, an improved algorithm [10] focusing on lossless recovery was proposed by Honsinger et al. One year later, Fridrich et al. [11] developed a high-capacity scheme based on embedding message bits in the status of pixel groups. In 2002, Celik et al. presented a lossless image compression algorithm [12] by using a prediction-based conditional entropy coder. This method uses some feature sets  $S$  of the image to form saved space by lossless compression and then utilizes them to hide information. However, as  $EC$  (embedding capacity) increases, more bit planes must be compressed, resulting in a sharp increase in distortion, which prevents the method from achieving satisfactory performance.

Thus, as can be seen, it has become the focus of scholars to determine how to reduce distortion while considering as much secret storage capacity as possible. Subsequently, expansion-based RDH schemes have been proposed to achieve the desired effectiveness, which is defined by three basic classifications, i.e., histogram shifting (HS), difference expansion (DE), and prediction-error expansion (PEE). The first HS-based reversible data hiding was proposed by Ni et al. [13] in 2003. The scheme is based on the concept of histogram of an image, which normally refers to the histogram of grayscale values of all pixels in the image, which shows the number of pixels at different grayscale values found in that image. For an 8 bit grayscale image, there are 256 possible grayscale values. The histogram will display the distribution of the pixels among those grayscale values between 0 and 255, as shown in Figure 1. For instance, the red mark in Figure 1 shows image "Baboon" has 2759 pixels with a grayscale value of 121. The scheme is implemented by shifting a certain histogram bin, usually the *peak* bin, which has the maximum number of pixels in the image. But HS techniques do not work well when the histogram of an image is equal. Although multiple pairs of peaks and minimums can be used for embedding, the embedding capacity is still insufficient. In 2003, with *peak* and *zero* bins to be chosen from the middle and the edge part of the histogram, respectively, a special case of HS, called the DE-based scheme, was first proposed by Tian [14], and it was based on the difference expansions of pixel pairs. Later, several improved techniques for DE-based embedding [15] were proposed, and they included integer transformation, pixel difference [16, 17], and prediction-error expansion (PEE). PEE is the focus of this paper, which was first proposed by Thodi and Rodriguez [18], and it extended the pixels and predicted differences for embedding data. Compared with the methods based on DE and HS, prediction-error histogram distribution is sharper and more centralized, so PEE has better performance. In addition to this, unlike DE, which only considers the correlation of two neighbouring pixels, and also unlike HS, which only considers the current pixel itself, PEE uses more adjacent pixels to obtain better performance.

Existing PEE-based RDH methods are mainly based on the modification of the one- or two-dimensional prediction-error histogram (PEH). The approach we propose using the PEH to form a method with a bidirectional shift and double-way prediction strategy for RDH and the details associated with its use are covered in the next section.

In this paper, we propose an improved RDH method based on the study [19] with double-way prediction and bidirectional shifting. The difference between the current value and the prediction value is used to form a prediction-error histogram. This histogram will be used to hide secret messages by expanding and shifting, and the embeddable bins are called *peak* (peak point) bins. Through outward shifting, they will be used to insert secret bits into the vacant bins reserved by other unembeddable bins that are also known as outer region bins. This process will be accomplished through two stages of prediction and corresponding processing based on the predicted values. The process is repeated for all pixels of the entire image to produce the stego image. During the extraction procedure, to draw out all of the secret bits and recover the original image, the receiver end can utilize the auxiliary information, which includes the position information of *zero* bins, threshold values, and pixels in the last row and the last column that have been kept intact. First, we calculate the prediction values of pixels in neighbourhood in the last row and the last column in the stego image. And the difference, named the prediction error (PE), between the prediction value and the stego-pixel value can be worked out at the same time. According to the PE, the quantity of the pixel value changes and the hidden information can be determined. Thus, the original pixel value can be worked out. Then pixel by pixel, the prediction value of the currently processed pixel can be determined using the pixels in its neighbourhood that have been recovered before it. Finally, all the pixel values of the original image and secret message are recovered. Therefore, the algorithm proposed in this paper is an RDH scheme. In addition, the experimental results show that the proposed method can achieve higher embedding capacity and better image quality than the previous scheme.

The contributions of this paper are as follows:

- (1) We apply bidirectional histogram shifting to improve the hiding capacity of a secret message. During the shifting procedure, we make two shift directions opposite to each other to reduce distortion.
- (2) We utilize a proposed double-way prediction strategy to get more embedding capacity.
- (3) Through the experimental data of 108 images, it is found that when  $TH = TH^* = 2$ , the algorithm effectively and significantly improves the information hiding capacity by 65% but only reduces the PSNR value about 6%, and the PSNR value remains above 30 dB.
- (4) The method can completely recover the image without any error; therefore, it is an RDH scheme with high hiding capacity.

The rest of this paper is organized as follows: In Section 2, Wang et al.'s method is reviewed, and the proposed

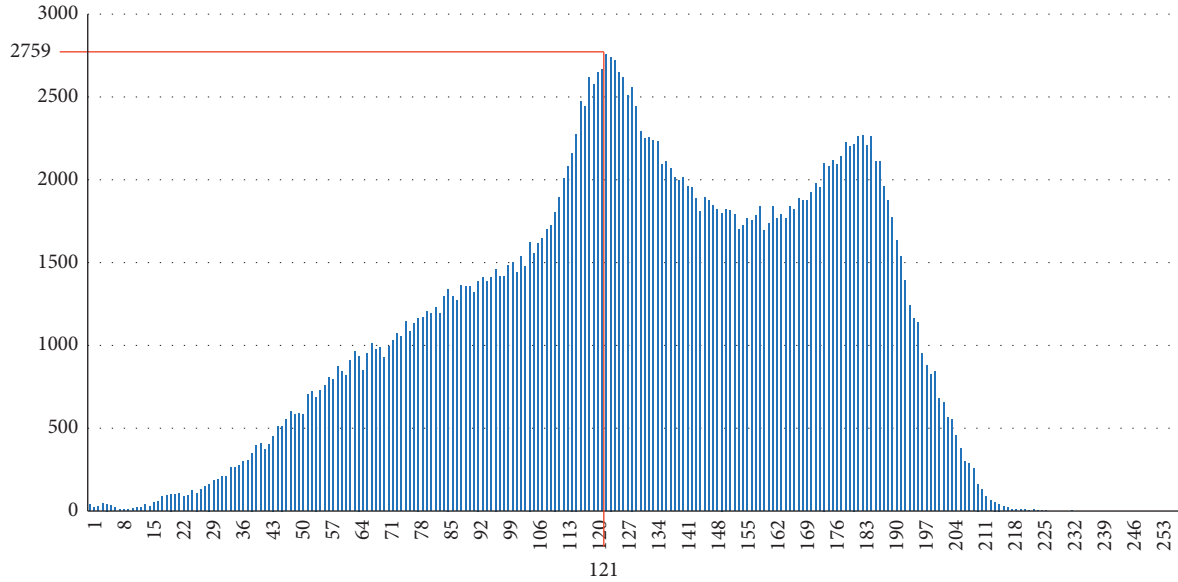


FIGURE 1: Histogram distribution of image “Baboon” with a peak point of 121.

method is described in Section 3. In Section 4, the experimental results are presented and discussed, and the conclusions are given in Section 5.

## 2. Review of Wang et al.’s Method

In this section, we introduce Wang et al.’s method [19], which is based on a double-way prediction error and right-left shift. To further decrease distortion of the stego image, the authors presented an extension of the traditional histogram shifting technique. The prediction error is produced by the difference between the value of the current pixel and the average value of the sum of its three adjacent pixels, i.e., the pixels in the right, bottom right, and bottom. Their technique is based on the principles that (1) the neighbouring pixels have a strong correlation and (2) the prediction-error distribution of adjacent pixels has a prominent maximum, that is, most of the prediction errors are expected to be very close to zero. The distribution of prediction errors has a zero mean value, similar to the Laplace distribution, which can be concluded from papers [20–23]. The authors utilize this characteristic of the distribution in combination with right-left shifting to improve the embedding capacity. To further introduce the algorithm, some basic concepts are provided in the next paragraph, and an example is given to illustrate the embedding and extraction procedures.

**2.1. Definition and Production of Prediction-Error Histogram.** To discuss the details of Wang et al.’s method, we use a grayscale image  $I$  of size  $H \times W$ , where  $H$  and  $W$  are the height and width of an image, respectively. Assume that the upper left coordinate position of  $I$  is  $(1, 1)$  and each pixel of  $I$  is assigned a coordinate. The coordinates of all pixels are successively increased from left to right and from top to bottom of the image. One of them is referred to as  $I(i, j)$ , as shown in Figure 2.

The scheme first predicts all pixels to form the prediction values  $P(i, j)$  of a pixel  $I(i, j)$  using the following equation:

$$P(i, j) = \text{fix}\left(\frac{I(i+1, j) + I(i, j+1) + I(i+1, j+1)}{3}\right), \quad (1)$$

where  $\text{fix}(\bullet)$  is a function, of which the input is a real number and the output is an integer. It rounds the input to the nearest integer toward zero. For example,  $\text{fix}(3.2) = 3$  and  $\text{fix}(8.6) = 8$ . Then, the PE (prediction-error) value  $P_e(i, j)$  of  $I(i, j)$  is obtained by the following equation:

$$P_e(i, j) = I(i, j) - P(i, j). \quad (2)$$

The pixels in the last row and the last column of an image are the reference pixels for recovery and thus cannot be predicted; the original values will be maintained. The preservation of these original data will play an important role in the extraction and recovery process.

Then,  $P_e(i, j)$  is produced by the formulation as formulated in equation (2). The prediction-error histogram (PEH) is generated by  $h(P_e)$  function, which counts the number of occurrences of a prediction-error (PE) value. It contains the *peak* bins, *zero* bins, and other bins, which are distributed symmetrically with values in the range of  $[-255, 255]$  for an 8 bit grayscale image. The PEH of “Barbara” is shown in Figure 3(a) as an example, where  $h(0) = 19,209$ ,  $h(81) = 337$ , and  $h(-68) = 100$ , which mean there are 19,209 pixels with a prediction-error value of “0,” 337 pixels with a prediction-error value of “81,” and 100 pixels with a prediction-error value of “-68.” Traditionally, the *peak* bin, which is also called the embeddable bin, is the bin that is located in the middle of the histogram. Here, the concept of peak bin is extended to all of the central bins within the threshold called  $TH$ , and the number of peak bins is defined as  $PK = 2 \times TH + 1$ , which consists of the number of traditional peak bins and the

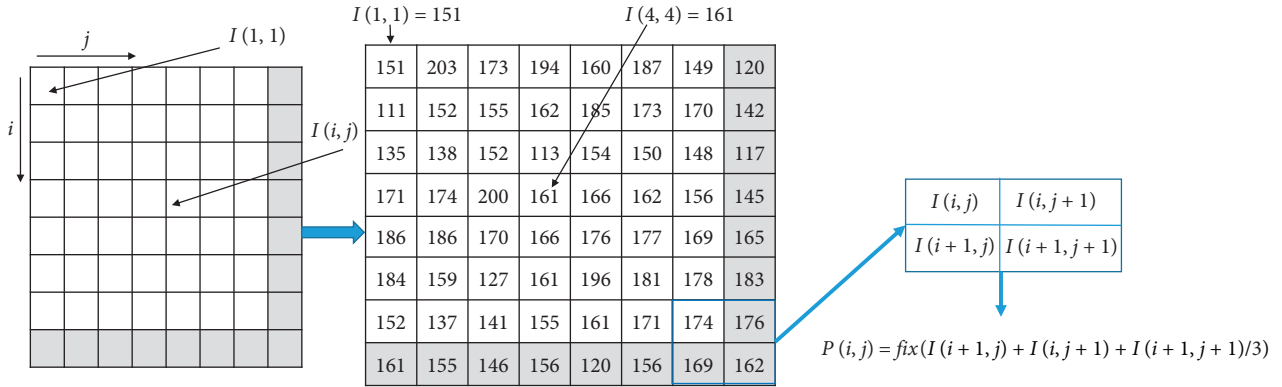


FIGURE 2:  $I(i, j)$  and  $P(i, j)$  of an image  $I$ .

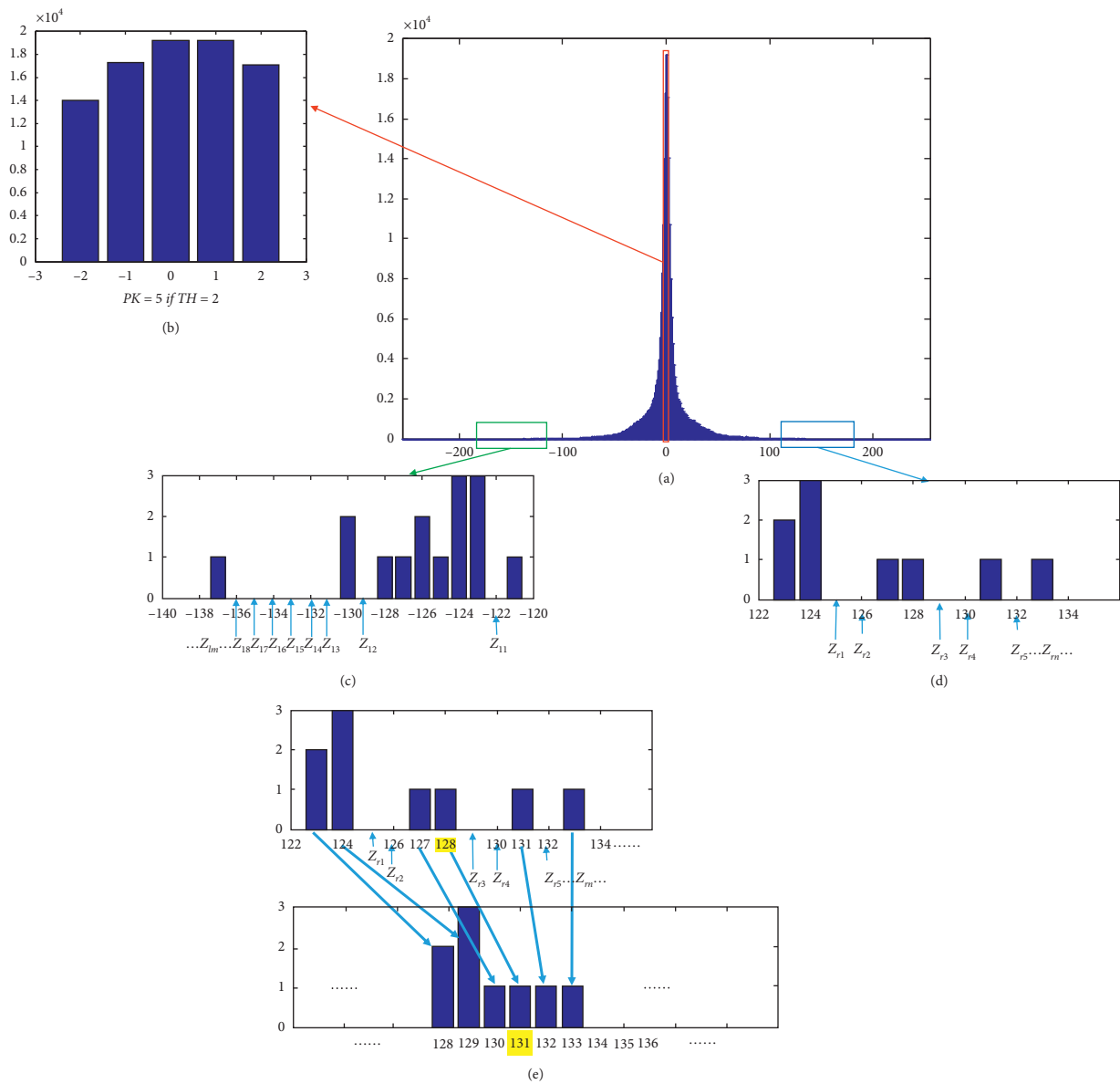


FIGURE 3: (a) Distribution of prediction errors of "Barbara." (b) Peak bins in the peak region. (c) Zero bins on the edge of the left outer region. (d) Zero bins on the edge of the right outer region. (e) Zero bins used for reducing shift and distortion.

bins on the left and right sides of it within the distance of  $TH$ , as shown in Figure 3(b).

During the embedding procedure, if the secret bit to be embedded is “1,” the corresponding *peak* bin will be expanded by 1 unit; otherwise, it remains unchanged. The outer bins will shift outward to make room for *peak* bins. *Zero* bins, which are located on two edges of the outer sides of the PEH, are denoted as  $Z_{rn}$  and  $Z_{lm}$ , respectively, acting as buffers for shifting. Here, zero means there are no PE values equal to that bin value. For example, in “Barbara,”  $Z_{r1} = 125$ , which means that there are no PE values equal to 125.  $Z_{r2} = 126$ ,  $Z_{r3} = 129$ ,  $Z_{r4} = 130$ ,  $Z_{r5} = 132$  and  $Z_{l1} = -122$ ,  $Z_{l2} = -129$ ,  $Z_{l3} = -131$  are the other *zero* bins, as shown in Figures 3(c) and 3(d), which are the enlarged version of Figure 3(a). In addition, subscripts  $n$  and  $m$  record the number of *zero* bins between the current bin and *peak* bins. For example, when the value of the current bin is 128, the largest  $Z_{rn}$  less than 128 is  $Z_{r2}$ , where  $n$  is 2, which means there are 2 *zero* bins between *peak* bins and 128. When right shifting is needed during the embedding procedure, these 2 *zero* bins can be occupied to reduce two moves and thus to reduce distortion. For instance, if we assume that all bins must shift to the right direction by  $\Delta bins = 5$ , then bin 128 will move to 131 rather than 133 because  $128 + \Delta bins - n = 128 + 5 - 2 = 131$ , and the occupancy of these 2 *zero* bins is achieved by subtracting 2, as shown in Figure 3(e). The same thing happens for a bin value of  $-130$ ; that is, the smallest  $Z_{lm}$  larger than  $-130$  is  $Z_{l2}$ , where  $m$  is 2, and when left shifting, these 2 *zero* bins can be occupied to reduce the quantity of moves by 2 and lessen distortion.

**2.2. Embedding Procedure.** The secret information to be hidden is denoted as a binary string  $S$ , with elements  $s_1, s_2, s_3, \dots, s_j, \dots, s_L$  of  $\{0, 1\}$ ,  $i, j \in [1, L]$ , where  $L$  is the length of  $S$ . We set  $TH = 2$  and take image “Barbara” as an example. The number of *peak* bins of “Barbara” is 5, as shown in Figure 3(a). According to the distribution of prediction errors of “Barbara,” Figure 4 is provided, and the details about the figure demonstrate right and left shifting of *peak* bins and the whole prediction-error histogram of “Barbara.” Because the image being discussed is a grayscale image, the histogram of prediction errors would be distributed in the interval  $[-255, 255]$ . Due to the similarity of the values of neighbouring pixels, the *peak* bins were concentrated on value 0 and its adjacent values. It is this feature that the authors used to hide information. There are two phases of movement of the PEH in the algorithm, which are first shifting to right and later shifting to left. The first line in Figure 4 is the original PEH of “Barbara,” the second line is the PEH after right shifting, and the third line is the PEH after left shifting.

When right shifting (RS), bin value  $-2$  remains as  $-2$  to hide the secret bit “0,” and it changes to  $-1$  to hide the secret bit “1.” In a similar way, bin value  $-1$  moves to 0 to embed the secret bit “0,” and it shifts to 1 to hide the secret bit “1.” One bin moved after another, bin value 2 will move to 6 and 7 to hide “0” and “1.” But the bins on the right and left sides of *peak* bins will not be used to hide secret bits. Due to the

movement of *peak* bins, the bins on the right side will have to shift outward to make room for *peak* bins. Thus, bin 3 will move to 8, bin 4 will move to 9, etc. To reduce the distortion, every *zero* bin can be used to buffer the amount of shifting by reducing the movement to the right side by one bin. Therefore, bin value 127 will move to 130 because there are 2 *zero* bins, i.e.,  $Z_{r1} = 125$  and  $Z_{r2} = 126$ , between bin value 127 and *peak* bins. The amount of moved units is  $PK - 2 = 5 - 2 = 3$ . Those bins with values less than  $-2$  will not be affected and maintain their original values.

When left shifting (LS),  $PK_l$  is 10, which is twice as large as the  $PK$  value. It is because after right shifting, bins of the *peak* region have already been expanded to twice the number of the original *peak* region. This time, all of the bin values that are larger than 8 remain unchanged, but those less than 8 will move. Concerning bin value 7, it remains unmoved to hide the secret bit, “0,” while it shifts to 6 to hide “1.” In a similar way, bin by bin, bin value  $-2$  moves to  $-11$  to embed the secret bit “0” and shifts to  $-12$  to hide the secret bit “1.” The bins on left sides of  $PK_l$  will not be used to hide secret data but must be moved to make room. For example, bin value  $-3$  will shift to  $-13$  and bin value  $-120$  will shift to  $-130$ . Instead of moving to the left side by  $PK_l = 10$ , bin value  $-123$  moves by  $10 - 1 = 9$  units because there is a *zero* bin  $Z_{l1} = -122$  between  $-123$  and *peak* bins, which is used to reduce one unit of movement.

To ensure better understanding, three examples are provided to explain how shifting and expanding work during the right-left shifting phase:

**Example 1.** The red values in Figure 4 are PE values of *peak* bins according to the definition of  $TH$ . In the cover image “Barbara,” the original pixel value  $I(2, 2)$  is 198, the corresponding prediction value  $P(2, 2)$  is 196 (calculated from equation (1)), and the prediction-error value is 2 (calculated from equation (2)), which is one of the *peak* bin values. In the RS (right shifting) stage, the amount of shifting is 5, which is the result of  $7 - 2$ , where the secret bit  $s_i$  is assumed to be “1” here. Thus, the value of the marked pixel is  $198 + 5 = 203$ . In the LS (left shifting) stage, the amount of shifting is 0, where the secret bit  $s_j$  is assumed as “0.” After the above procedure, the pixel value  $I(2, 2)$  of the cover image is changed from 198 to 203, which is the final value of its corresponding stego pixel, with embedding two secret bits.

**Example 2.** The green values in Figure 4 are bin values between 3 and 133. Number 3 is the smallest prediction error greater than  $TH$ , and number 133 is the greatest nonzero bin value. In the RS stage, these bin values will add a relative positive integer according to the subtraction calculation of  $PK - n$ , where  $n$  is the total quantity of *zero* bins less than the current bin value. In the LS stage, these bin values will remain unchanged. For example,  $I(371, 457) = 228$ ,  $P(371, 457) = 97$ , and  $P_e(371, 457) = 228 - 97 = 131$  from equation (2), and the amount of shifting is  $PK - n = 5 - 4 = 1$ , where  $n$  is 4 because the largest *zero* bin less than 131 is  $Z_{r4}$  (Figure 3(d)), which means there are 4 *zero* bins less than 131. Thus, bin 131 moves to 132, and the marked pixel value

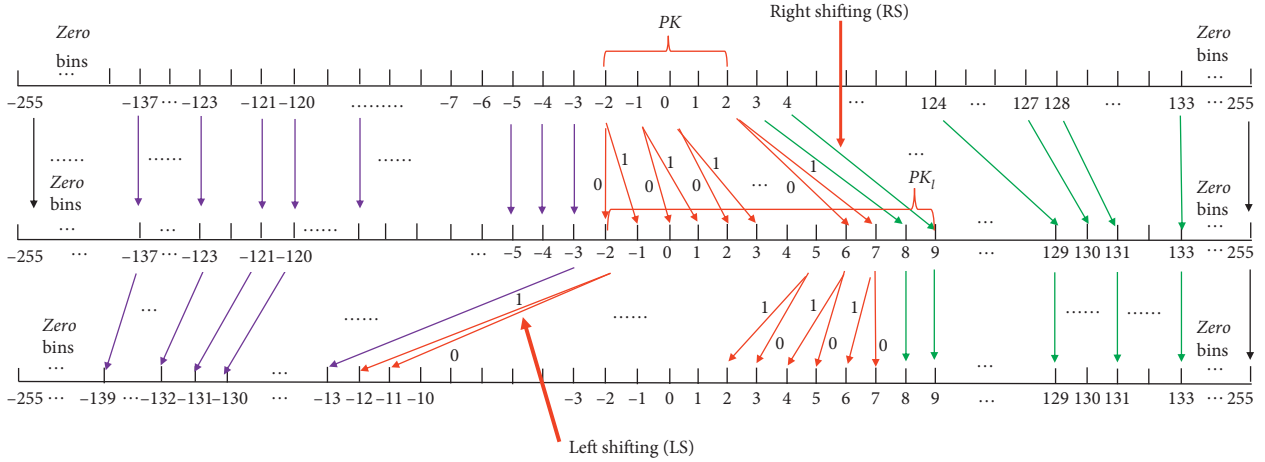


FIGURE 4: Right and left shifting of the whole prediction-error histogram (PEH) of “Barbara.”

is  $228 + 1 = 229$ . In the second-stage hiding procedure, the shifting of bin 132 is not needed, so the value of the final stego image pixel of  $I(371, 457)$  is 229.

*Example 3.* The purple values in Figure 4 are bin values between  $-3$  and  $-137$ . Number  $-3$  is the largest prediction error less than  $-TH$ , which is  $-2$ , and number  $-137$  is the smallest nonzero bin value. In the RS stage, these bin values will remain unchanged because they are less than  $-PK$ . In the LS stage, these bin values will subtract a relative positive integer according to the subtraction calculation of  $PK_L - m$ , where  $m$  is the total quantity of zero bins greater than the current discussing bin value. For example,  $I(378, 453) = 51$ ,  $P(378, 453) = 188$ , and  $P_e(378, 453) = 51 - 188 = -137$ . In the RS stage, bin value  $-137$  remained unchanged; thus, the amount of shifting of bin value  $-137$  is 0, which means pixel value 51 will remain unchanged. In the LS stage, the amount of shifting is  $PK_L - m = 10 - 8 = 2$ , where  $m$  is 8 because the smallest zero bin on the right side of bin  $-137$  is  $Z_{18}$  (Figure 3(c)). At last, the final stego-pixel value of  $I(378, 453)$  is  $51 - 2 = 49$ .

**2.3. Extracting Procedure.** After the embedding procedure,  $TH$ , the zero bins  $Z_{rn}$  and  $Z_{lm}$ , and the stego image, in which the last row and the last column are original and intact, will be sent to the receiver. Thus, the predicted pixel values can be calculated from the pixel values in the last row and the last column. To be specific, according to equation (1), for an image with a size of  $512 \times 512$ , the prediction value  $P(511, 511)$  is first calculated, and after that, the difference between the prediction value and the stego-pixel value of  $P(511, 511)$  is worked out. Then, based on the prediction error and how it relates to  $TH$ , the zero bins  $Z_{rn}$  and  $Z_{lm}$ , the original value of  $I(511, 511)$ , and the embedding bit value (if it had been hidden before) are solved. Now as a given message, the original value of  $I(511, 511)$  can be used to determine  $I(511, 510)$ . Thus, pixel by pixel, the whole secret information can be extracted and the original image can be recovered in the reverse processing order of the embedding procedure.

### 3. Proposed Scheme

To improve the quality of the stego image and increase the amount of secret information that can be hidden, we use bidirectional PEH shifting in double-way prediction based on Wang et al.’s method [19]. The process of generating the stego image is divided into two stages: embedding process 1 using the raster scan order and embedding process 2 using the inverse raster scan order. First of all, pixels in the last row and the last column are kept and used as reference pixels. The first prediction is made in a raster scan order, i.e., from the first pixel in the upper left corner of an image, top to bottom, and left to right, and the top-left pixel is predicted by the other three pixels in its square-shaped adjacent context. The calculation process is shown in equation (1), and the prediction error,  $P(i, j)$ , is calculated by equation (2); the second-stage prediction is processed in the reverse raster scan order from the pixel in the bottom right corner, right to left, and bottom to top, and the lower-right pixel is predicted by the other three pixels as follows, respectively:

$$P^*(i, j) = \text{fix} \left( \frac{I_1(i-1, j) + I_1(i, j-1) + I_1(i-1, j-1)}{3} \right), \quad (3)$$

$$P_e^*(i, j) = I_1(i, j) - P^*(i, j), \quad (4)$$

where  $I_1$  and  $I_1(i, j)$  are the stego image and its stego pixel after embedding process 1. In the process of information hiding in these two stages, PE (prediction-error) values are determined separately, and they are named  $P(i, j)$  and  $P^*(i, j)$ , respectively. In addition, during embedding process 1, we first expand or shift bins of the PEH to the right direction and afterward to the left direction. During embedding process 2, to avoid severe distortion which may occur, the movement is set to the opposite direction from the first embedding process. That is to say, during embedding process 2, we first expand or shift bins of the PEH to the left direction and then to right. The number of moving units depends on the interval in which the value of the bin falls. After the shifting of all the pixels has been finished, the

TABLE 1: Notations of the proposed method (all variables marked with a star are used in embedding process 2).

---

PEH: prediction-error histogram
PE: prediction error
S: secret information in binary bits with elements $s_1, s_2, s_i, \dots, s'_i, \dots, s_j, \dots, s'_j, \dots, s_L$ , where $L$ is the length of $S$
TH, $TH^*$ : $[-TH, TH]$ , $[-TH^*, TH^*]$ are the ranges of peak bins
PK, $PK^*$ : the quantity of peak bin values
$PK_b, PK_r^*$ : double the number of <i>peak</i> bins
$I, I(i, j)$ : an 8 bit grayscale cover image with a size of $H \times W$ ( $H$ is the height and $W$ is the width), $I(i, j)$ is a pixel of $I$ in row number $i$ and column number $j$
$I_1, I_1(i, j)$ : the first-stage stego image and one of its pixels
$\bar{I}, \bar{I}(i, j)$ : the final stego image and one of its pixels
$P(i, j), P^*(i, j)$ : prediction value of $I(i, j), I_1(i, j)$
$P_e(i, j), P_e^*(i, j)$ : prediction error of pixel $I(i, j)$
$P_{el}(i, j), P_{el}^*(i, j)$ : prediction value of pixels $I_{1r}(i, j)$ and $I_1(i, j)$ after left shifting
$P_{er}(i, j), P_{er}^*(i, j)$ : prediction value of pixels $I_{2l}^*(i, j)$ and $\bar{I}(i, j)$ after right shifting
$h(p_e)$ : the frequency of the prediction-error value $p_e$ in the histogram, where $p_e \in [-255, 255]$
$I_{1r}(i, j)$ : the stego image after right shifting of PEH in embedding process 1
$I_{2l}^*(i, j)$ : the stego image after left shifting of PEH in embedding process 2
$\Delta bin(i, j), \Delta bin_l(i, j)$ : the quantity of units to move during expanding and shifting in right shift and left shift in embedding process 1
$\Delta bin^*(i, j), \Delta bin_r^*(i, j)$ : the quantity of units to move during expanding and shifting in left shift and right shift in embedding process 2
$n, n^*$ : the number of zero bins between the current positive bin value and the <i>peak</i> bins
$m, m^*$ : the number of zero bins between the current negative bin value and the <i>peak</i> bins

---

original prediction error (PE) is altered to the modified prediction error (MPE). The difference of the MPE and PE, called  $\Delta bins$ , is added to the original pixel to create the value of the stego pixel.

The notations of this scheme are presented in Table 1, and the subsequent discussion explains the algorithm process in detail, including the pseudocode of the hiding and extracting information procedure.

The details of embedding process 1 are explained by the pseudocode presented in Algorithm 1.

Figure 5 is an illustration of embedding process 1. We assume  $TH = 2$ , and there is a pixel  $P(i, j)$  with its PE value  $P_e(i, j) = 0$ , which is a value belonging to bin value 0 shown in Figure 5(a). During right shifting of the PEH, bin value 0 is shifted to bin value 2, with “0” embedded, as shown in Figure 5(c). That is to say,  $\Delta bin(i, j) = P_e(i, j) + TH + s_i = 0 + 2 + 0 = 2$ . Then,  $I(i, j)$  is changed to  $I_{1r}(i, j)$ , where  $I_{1r}(i, j) = I(i, j) + \Delta bin(i, j) = I(i, j) + 2$ . Then, left shifting is used to embed “1” in  $I_{1r}(i, j)$ , which changes bin value 2 to  $-4$ .  $P_{er}(i, j) = P_e(i, j) + \Delta bin(i, j) = P_e(i, j) + 2$ ;  $\Delta bin_l(i, j) = P_{er}(i, j) - TH - PK - s_j = P_e(i, j) + 2 - 2 - 5 - 1 = -6$ . Finally,  $I_1(i, j) = I_{1r}(i, j) + \Delta bin_l(i, j) = I(i, j) + 2 - 6 = I(i, j) - 4$ . After that, pixel  $I(i, j)$  has been embedded with “0” and “1” with pixel value adding  $-4$ . Thus, with varied quantity of movements of *peak* region bins in the PE histogram, secret data are hidden. Moving the outer region makes room for the *peak* region to move and hide secret information. Those pixel values whose prediction-error values are located in the right outer region are increased. Those pixel values whose prediction-error values are located in the left outer region are decreased. In addition, with help of zero bins, the amount of movement in the rear of the outer region can be reduced to some extent. Of all these, the maximum change is  $2 \times PK$ .

Next, the current stego image  $I_1$  will be predicted with equation (3), and the order of prediction processing is in the reverse order from the prediction in embedding process 1. Then, the PEH shifting direction is toward left and then right which is the opposite order of embedding process 1 also. Compared with Wang et al.’s scheme [19], the distortion will be a little bit larger due to the increase in the hiding capacity of secret information. The pseudocode of embedding process 2 is given in Algorithm 2.

The overflow/underflow problem occurs when the values of some pixels, called boundary pixels, are changed from 255 to greater than 255 or from 0 to negative values. For these pixels, according to reference [24], we use a different approach from Wang et al.’s. The values that will exceed the boundary after  $\Delta bins$  are added will not be allowed to participate in the addition; for them, the original values are kept unchanged and called pseudo-values. The pixel values, whose addition result values are equal to the corresponding pseudovalues, are called genuine values. A boundary array can be recorded to distinguish genuine values from pseudovalues to avoid the confusion. Each member of the array corresponds to a boundary pixel in the stego image, with genuine “0” and pseudo “1,” respectively.

**3.1. An Example of Embedding Process.** As shown in Figure 6, pixel  $I(31, 12)$  of “*Baboon*” is identified as  $I(31, 12)$ , and its pixel value is 119. Based on equation (1),  $P_e(31, 12) = -2$  satisfies the condition  $P_e(i, j) < TH$ , where  $TH = 0$ . Thus,  $\Delta bin$  value in right shifting of embedding process 1 of  $I(31, 12)$  is 0,  $I_{1r}(31, 12) = I(31, 12) = 119$ , and  $P_{er}(31, 12) = -2$ . In left shifting of embedding process 1, due to  $P_{er}(31, 12) < TH = 0$ ,  $\Delta bin_l(31, 12) = -PK_l = -2$ ; therefore,  $I_1(31, 12) =$



```

Input: grayscale cover image  $I$ , secret  $s_i, s'_i$  as one bit of the secret message  $S$ , and threshold  $TH$ 
Output: stego image  $I_1$ 
for each pixel  $I(i, j)$  (except for the pixels in the last column and the last row)
  Scan in the raster scan order;
  Calculate the prediction error of  $I(i, j)$  as  $P_e(i, j)$  with equations (1) and (2);
  Generate PEH of  $I, PK = 2 \times TH, PK_l = 2 \times PK$ ;
  Find out zero bins and denote them as  $Z_{rn}$  and  $Z_{lm}$ ;
  /*right expanding and shifting*/
  if  $-TH \leq P_e(i, j) \leq TH$ 
     $\Delta bin(i, j) = P_e(i, j) + TH + s_i$ ;
  elseif  $TH < P_e(i, j)$ 
     $\Delta bin(i, j) = PK - n$ , where  $Z_{rn} \leq P_e(i, j) < Z_{r(n+1)}, n \leq PK$ ;
  else
     $\Delta bin(i, j) = 0$ ;
  endif;
   $I_{1r}(i, j) = I(i, j) + \Delta bin(i, j)$ ;
end;
/*finish right and begin left expanding and shifting*/
for each pixel  $I_{1r}(i, j)$ 
   $P_{er}(i, j) = I_{1r}(i, j) - P(i, j) = I(i, j) + \Delta bin(i, j) - P(i, j) = P_e(i, j) + \Delta bin(i, j)$ ;
  if  $-TH \leq P_{er}(i, j) \leq TH + PK_l$ 
     $\Delta bin_l(i, j) = P_{er}(i, j) - TH - PK - s'_i$ ;
  elseif  $P_{er}(i, j) < -TH$ 
     $\Delta bin_l(i, j) = -PK_l + m$ , where  $z_{l(m+1)} \leq P_{er}(i, j) < z_{lm}, m \leq PK_l$ ;
  else
     $\Delta bin_l(i, j) = 0$ ;
  endif;
   $I_1(i, j) = I_{1r}(i, j) + \Delta bin_l(i, j)$ ;
   $P_{el}(i, j) = P_e(i, j) + \Delta bin(i, j) + \Delta bin_l(i, j)$ ;
end;
/*finish embedding process 1*/

```

ALGORITHM 1: Pseudocode of embedding process 1.

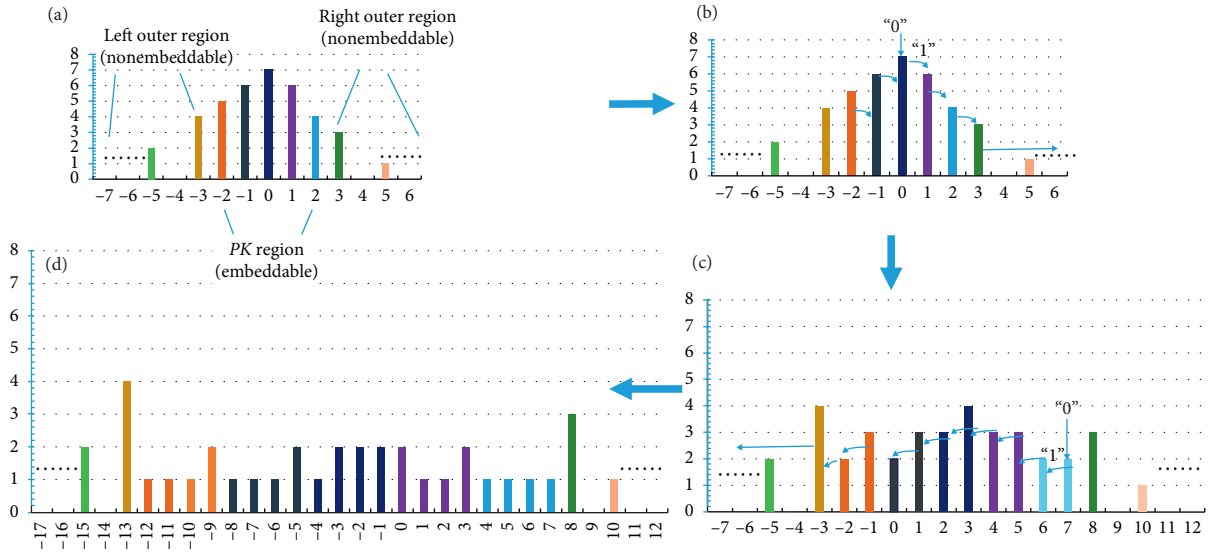


FIGURE 5: (a) Three regions of the histogram. (b) Right shift direction. (c) Left shift direction. (d) The final histogram.

$I_{1r}(31, 12) + \Delta bin_l(31, 12) = 119 + (-2) = 117$ , which is the result of embedding process 1. Next comes embedding process 2. The new prediction value is generated according

to equation (3), and the prediction error  $P_e^*(31, 12) = 0$  is obtained. Now comes left shifting of embedding process 2. According to the given threshold value, i.e.,  $TH^* = 0$ , we find

```

Input: first-stage stego image  $I_1$ , secret  $s_j$  and  $s'_j$  that are, respectively, one bit of the secret message  $S$ , and threshold  $TH^*$ 
Output: stego image  $\tilde{I}$ 
for each pixel,  $I_1(i, j)$ , except for the pixel in the first column and the first row of  $I_1$ 
  Scan in the reverse raster scan order from the pixel in the bottom right corner of  $I_1$ ;
  Calculate the prediction error of  $I_1(i, j)$  as  $P_e^*(i, j)$  with equations (3) and (4);
  Generate PEH* of  $I_1$  and  $PK^* = 2 \times TH^*$ ,  $PK_r^* = 2 \times PK^*$ ;
  Compute zero bins and denote them as  $Z_{rn}^*$  and  $Z_{lm}^*$ ;
  /*left expanding and shifting*/
  if  $-TH^* \leq P_e^*(i, j) \leq TH^*$ 
     $\Delta bin^*(i, j) = P_e^*(i, j) - TH^* - s_j$ ;
  elseif  $P_e^*(i, j) < -TH^*$ 
     $\Delta bin^*(i, j) = -PK^* + m^*$ , where  $Z_{l(m^*+1)}^* \leq P_e^*(i, j) < Z_{lm^*}^*$ ,  $m^* \leq PK^*$ ;
  else
     $\Delta bin^*(i, j) = 0$ ;
  endif;
   $I_{2l}^*(i, j) = I_1(i, j) + \Delta bin^*(i, j)$ ;
end;
/*finish left and begin right expanding and shifting*/
for each pixel  $I_{2l}^*(i, j)$ 
   $P_{el}^*(i, j) = I_{2l}^*(i, j) - P^*(i, j) = P_e^*(i, j) + \Delta bin^*(i, j)$ ;
  if  $-TH - PK^* \leq P_{el}^*(i, j) \leq TH^*$ 
     $\Delta bin_r^*(i, j) = P_{el}^*(i, j) + SH^* + PK^* + s'_j$ ;
  elseif  $P_{el}^*(i, j) > SH^*$ 
     $\Delta bin_r^*(i, j) = PK_r^* - n^*$ , where  $Z_{rn^*} \leq P_{el}^*(i, j) < Z_{r(n^*+1)}$ ,  $n \leq PK_r^*$ ;
  else
     $\Delta bin_r^*(i, j) = 0$ ;
  endif;
   $\tilde{I}(i, j) = I_{2l}^*(i, j) + \Delta bin_r^*(i, j)$ ;
   $P_{er}^*(i, j) = P_e^*(i, j) + \Delta bin^*(i, j) + \Delta bin_r^*(i, j)$ ;
end;
/*finish embedding process 2*/

```

ALGORITHM 2: Pseudocode of embedding process 2.

that  $P_e^*(31, 12) = TH^*$ . Assume that the current information to be hidden is  $s_j = 1$ . Thus,  $\Delta bin^*(31, 12) = -1$ , which expands bin value 0 to  $-1$  to hide the secret bit "1"; thus,  $I_{2l}^*(31, 12) = 116$ , and  $P_{el}^*(31, 12) = -1$ . Then, we proceed with right shifting of embedding process 2.  $P_{el}^*(31, 12)$  satisfies the condition  $-TH - PK^* \leq P_{el}^*(i, j) \leq TH^*$ , and we assume that the current information to be hidden is  $s'_j = 1$  so that  $\Delta bin_r^*(31, 12) = 1$  hides "1." Finally, the stego pixel is  $\tilde{I}(31, 12) = 117$ . At this point, the embedding processes are finished.

**3.2. An Example of Extraction Process.** Now, we assume that the receiver receives the stego image, threshold  $TH$ ,  $TH^*$ , zero bins  $Z_{rn}$ ,  $Z_{lm}$ ,  $Z_{rn}^*$ , and  $Z_{lm}^*$ , and a boundary array of over/underflow pixels trying to extract the secret and recover the original image. First, based on the first line and first column pixels that have been preserved, the receiver can determine the prediction value of pixel (2, 2) and, then, its  $I_1(2, 2)$  pixel value. Then pixel by pixel, in the raster scan order, all of the ones before  $I_1(31, 12)$  will be recovered. Then, calculating by equation (3) and using  $I_1(30, 11)$ ,  $I_1(30, 12)$ , and  $I_1(31, 12)$  that have been determined, the prediction value is determined, that is,  $P^*(30, 12) = 117$ . And the difference between  $P^*(30, 12)$  and stego pixel  $\tilde{I}(31, 12)$ , which is  $P_{er}^*(31, 12) = 0$ , can then be figured out.

Based on the difference value and  $TH^* = 0$ , we can determine that  $P_{er}^*(31, 12) = -1$  and  $s'_j = 1$ . Thus,  $I_{2l}^*(31, 12) = P_{er}^*(31, 12) + P^*(31, 12) = -1 + 117 = 116$ ,  $s_j = 1$ , and  $P_e^*(31, 12) = 0$ . Thus,  $I_1(31, 12) = 117$ . Through the same process, we can determine every  $I_1(i, j)$ , pixel by pixel. Subsequently, the first phase of the extraction process is completed. And, based on the values of  $I_1$  and all of the values in the last column and last row of the image that have maintained their original values from the cover image, we can determine, pixel by pixel, the prediction value and prediction errors of all the pixels. The difference between  $I_1(31, 12)$  and  $P(30, 12)$ , called  $P_{el}(31, 12)$ , is  $-4$ ; thus,  $P_{er}(31, 12) = -2$ ,  $\Delta bin_l^*(i, j) = -2$ , and  $I_{1r}(31, 12) = I_1(31, 12) - \Delta bin_l(31, 12) = 117 + 2 = 119$  because of the given threshold value  $TH = 0$ ; also,  $\Delta bin(31, 12) = 0$  and  $P_e(31, 12) = 0$ . Thus,  $I(31, 12) = I_{1r}(31, 12) - \Delta bin(31, 12) = 119$ , which is the original pixel value. The process is shown in Figures 6 and 7, in which the blue arrow points out the embedding process and the orange arrow indicates the extraction process.

## 4. Experimental Results and Analysis

The experimental platform we use consists of an Intel 3.41 GHz i7-6700 CPU, 32 GB RAM, and the Windows 10

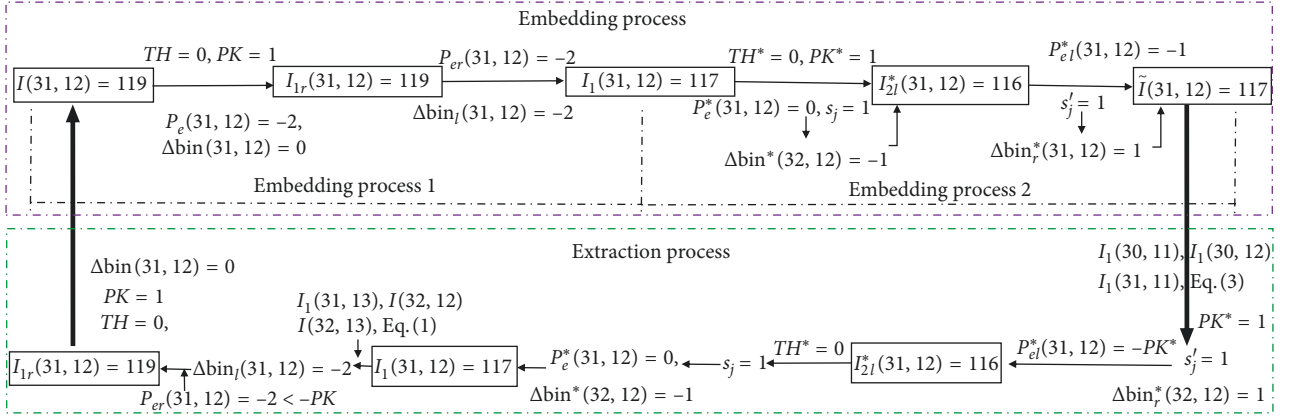
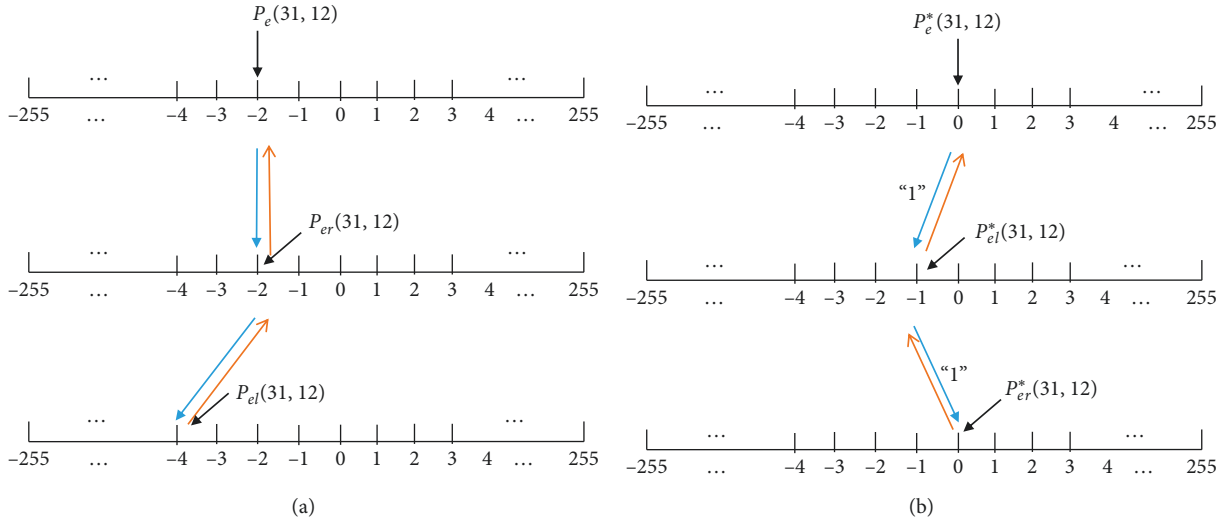


FIGURE 6: An example of processing the pixels of image "Baboon."

FIGURE 7: Histogram shifting of embedding process 1, embedding process 2, and the recovery process of  $I(31, 12)$ . (a) Right-left shifting ( $TH=0$ ). (b) Left-right shifting ( $TH^*=0$ ).

Professional 64 bit operating system. Our experiments are implemented by MATLAB. The performance of the proposed scheme is estimated by some criteria, i.e., embedding capacity (EC), peak signal-to-noise ratio (PSNR), and execution time (s: seconds). Tables 2–5 show some figures that are determined by using standard  $512 \times 512$  grayscale common, medical, texture, and aerial images, respectively.

EC can be controlled by quantity of *peak* bins, and it can be calculated by equation (6). The embedding rate (ER) represents the percentage of the embedded secret bits in the whole pixels of the cover image. ER is defined as in (7). Here,  $H \times W$  is the size of the cover image:

$$EC = h(p_e), \text{ where } -PK \leq p_e \leq PK, \quad (5)$$

$$ER = \frac{EC}{H \times W}, \quad (6)$$

where  $PK$  is the quantity of *peak* bin values we use to hide the secret messages, the value of which, in fact, will change in different embedding phases according to the  $TH$  value we use.

In general, the peak signal-to-noise ratio (PSNR) is used to measure the visual quality of the reconstructed image or the stego image. The PSNR is defined as

$$PSNR = 10 \times \log\left(\frac{255^2}{MSE}\right), \quad (7)$$

where MSE is the mean square error between the cover image and the stego image. For a grayscale cover image of size  $H \times W$ , the MSE is defined as

$$MSE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (p_{ij} - \bar{p}_{ij})^2, \quad (8)$$

where  $p_{ij}$  represents the pixel value in row  $i$  and column  $j$  of the original image and  $\bar{p}_{ij}$  is the corresponding stego-pixel value of  $p_{ij}$ .

As a common sense, the difference between the cover image and the stego image is invisible to naked eyes when the value of the PSNR exceeds 30 dB. Thus, to control the imperceptible distortion of a stego image, the embedding process will always be stopped when the PSNR value is less

TABLE 2: Increased embedding capacity ( $\Delta EC$ ) and its increasing rate (IR), the PSNR (dB) and its decreasing rate (DR), and the increased time ( $\Delta t$ ) of eight common images.

Value names		Lena	Barbara	Baboon	Lake	Boat	Bird	Peppers	Harbour
$TH=0$ $TH^*=0$	$\Delta EC$ (bits)	35188	27332	14824	25812	23924	20556	29530	25790
	IR	67.3%	71.1%	85.3%	66.9%	71.1%	63.1%	68.9%	62.2%
	PSNR (dB)	40.9	40.8	40.5	40.5	40.6	40.8	40.3	40.3
	DR	8.1%	8.2%	8.5%	8.9%	8.7%	8.0%	9.4%	9.2%
	$\Delta t$ (s)	1.74	1.35	1.34	6.05	4.24	1.44	1.38	1.32
$TH=1$ $TH^*=0$	$\Delta EC$ (bits)	16848	13533	8322	12697	11893	9979	13770	11795
	IR	11.2%	12.1%	8.21%	10.9%	7.8%	8.27%	8.6%	7.36%
	PSNR (dB)	34.44	34.60	34.12	34.39	34.45	34.9	34.33	34.37
	DR	3.0%	2.11%	2.40%	2.62%	2.48%	2.54%	3.05%	2.74%
	$\Delta t$ (s)	1.35	1.33	1.34	1.33	1.39	1.34	1.39	1.37
$TH=1$ $TH^*=1$	$\Delta EC$ (bits)	67764	54926	33240	50796	48708	43870	55856	47600
	IR	45.2%	49.3%	65.1%	46.1%	49.4%	60.7%	43.8%	40.1%
	PSNR (dB)	32.82	32.70	32.14	32.41	32.49	32.43	32.26	32.16
	DR	11.2%	11.2%	11.5%	12.1%	11.8%	11.0%	12.2%	12.3%
	$\Delta t$ (s)	1.35	1.32	1.35	1.39	1.40	1.43	1.37	1.34
$TH=2$ $TH^*=0$	$\Delta EC$ (bits)	9122	7428	4558	6847	6551	5425	7200	6536
	IR	3.9%	4.3%	5.4%	4.0%	4.2%	4.9%	3.6%	3.6%
	PSNR (dB)	31.15	30.85	30.17	30.69	30.71	30.33	30.78	30.72
	DR	2.0%	1.9%	2.2%	2.5%	2.3%	2.1%	2.2%	2.4%
	$\Delta t$ (s)	1.34	1.31	1.33	1.39	1.42	3.47	1.51	1.46
$TH=2$ $TH^*=1$	$\Delta EC$ (bits)	54966	44662	27666	41252	39150	33694	44082	38686
	IR	23.7%	25.7%	32.7%	24.1%	24.9%	30.2%	21.8%	21.3%
	PSNR (dB)	30.93	30.70	30.09	30.42	30.47	30.26	30.38	30.43
	DR	7.8%	7.6%	8.1%	8.6%	8.3%	7.7%	9.4%	8.6%
	$\Delta t$ (s)	1.41	1.39	1.42	1.68	1.38	1.35	1.51	1.36
$TH=2$ $TH^*=2$	$\Delta EC$ (bits)	90566	74417	46468	68132	65416	59622	73688	64756
	IR	39.1%	42.9%	55.1%	40.2%	41.6%	53.4%	36.4%	35.7%
	PSNR (dB)	30.07	30.12	30.08	30.15	30.18	30.25	30.19	30.17
	DR	13.4%	13.3%	13.7%	14.3%	14.0%	13.2%	15.2%	14.4%
	$\Delta t$ (s)	1.32	1.33	1.36	1.33	1.38	1.32	1.36	1.31

TABLE 3: Experimental results for eight medical images ( $TH=2$ ,  $TH^*=2$ ).

Images (512 $\times$ 512)	Payload (bits)	PSNR of the stego image (dB)		ER (bpp: bits per pixel)
	Increasing rate	Decreasing rate		
Mpic1	1021589	36.1		3.90
	95.6%	13.5%		
Mpic2	1017817	35.9		3.89
	94.9%	13.8%		
Mpic3	1020413	36.0		3.42
	95.4%	13.6%		
Mpic4	1021301	36.1		3.91
	95.6%	13.5%		
Mpic5	1021235	36.0		3.90
	95.6%	13.6%		
Mpic6	1021455	34.1		3.9
	95.6%	13.5%		
Mpic7	1021575	34.1		3.90
	95.7%	13.4%		
Mpic8	1010503	33.8		3.85
	93.5%	14.4%		

than 30. In our scheme, to avoid huge distortion, we only provide the experimental data with both of  $TH$  and  $TH^*$  values less than 2.

Compared with the original algorithm of Wang et al.'s [19] that made only one-way prediction, our algorithm uses

double-way predictions to generate two PEHs in order to hide more information. To minimize PSNR growth, during the second prediction, histogram shifting will be conducted from left to right, which is contrary to the shift order of the first prediction phase. Moreover, the original algorithm

TABLE 4: Experimental results for eight texture images ( $TH = 1$ ,  $TH^* = 1$ ).

Images (512 × 512)	PSNR of the stego image (dB)		ER (bpp: bits per pixel)
	Payload (bits) Increasing rate	Decreasing rate	
Texture1	88669	33.8	0.34
	73.2%	8.5%	
Texture2	144165	34.2	0.55
	60.5%	8.4%	
Texture3	33855	34.8	0.13
	70.5%	5.5%	
Texture4	71375	34.1	0.27
	77.1%	8.2%	
Texture5	860807	37.7	3.28
	70.4%	12.0%	
Texture6	61407	33.3	0.23
	82.8%	9.4%	
Texture7	91077	33.7	0.35
	71.8%	9.4%	
Texture8	31029	34.1	0.12
	86.4%	7.6%	

TABLE 5: Experimental results for eight aerial images ( $TH = 1$ ,  $TH^* = 1$ ).

Images (512 × 512)	PSNR of the stego image (dB)		ER (bpp: bits per pixel)
	Payload (bits) Increasing rate	Decreasing rate	
Aerial 1	94641	32.8	0.36
	66.5%	11.9%	
Aerial 2	155697	33.5	0.59
	53.0%	10.1%	
Aerial 3	235241	32.6	0.91
	37.3%	13.9%	
Aerial 4	87289	32.9	0.33
	70.0%	11.4%	
Aerial 5	151937	33.3	0.58
	54.8%	10.9%	
Aerial 6	104207	33.2	0.41
	65.2%	10.6%	
Aerial 7	81633	33.2	0.31
	62.7%	10.8%	
Aerial 8	122205	32.8	0.47
	43.0%	12.7%	

subtracted 255 when pixel values were greater than 255, or simply added 255 when pixel values were less than zero. The solution of the underflow/overflow problems they used was inefficient. In this paper, we solve the problems in an efficient way based on the study [24] by avoiding some boundary pixels from performing huge transition changes, such as from white to black or from black to white. Increased embedding capacity ( $\Delta EC$ ) in bits and the EC increasing rates (IR), as well as the PSNR and its decreasing rate (DR), for the eight images in Figure 8 are shown in Table 2. Comparing our proposed algorithm with the algorithm proposed by Wang et al., Table 2 shows that, by using our proposed algorithm, there are a huge increase in EC and a slight decrease in PSNR, and the decrease rate of PSNR is significantly smaller than the growth rate of EC.

Table 2 indicates that, by using the proposed double-way prediction, the embedding capacity is increased for all the

images we studied. However, the increment of each image is different, and the increase range of the same image under different  $TH$  and  $TH^*$  conditions is also different. Moreover, PSNR values of all of the images decreased, and the amounts of the decreases are different. It is worth noting that the increase rate of EC is far greater than the decrease rate of PSNR. And an example is used to illustrate this finding. For image “Lena,” when  $TH = 0$  and  $TH^* = 0$ ,  $\Delta EC$  is 35,188 bits, which means that compared with the original algorithm, the proposed algorithm can embed 35,188 more bits in the image, improving the embedding capacity by 67.3%; for the same conditions, the PSNR value maintains the ideal value of 40.9, with a decrease rate of only 8.1%; time cost increased by only 1.74 seconds. Thus, compared with the original algorithm, the proposed algorithm sacrifices a little bit of PSNR value and time cost in order to hide a lot more secret information. This phenomenon is even more pronounced for

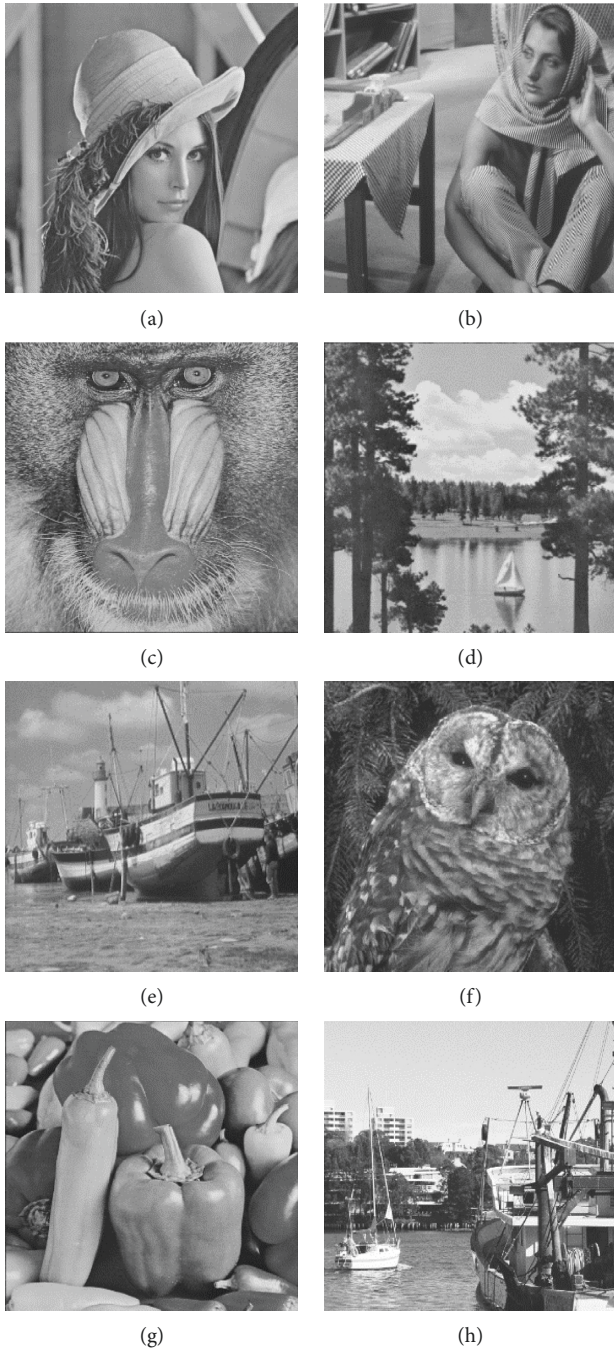


FIGURE 8: Eight common images with sizes of  $512 \times 512$  pixels. (a) Lena. (b) Barbara. (c) Baboon. (d) Lake. (e) Boat. (f) Bird. (g) Peppers. (h) Harbour.

image “*Baboon*.” The amount of hidden information is increased greatly by more than 85%, while the PSNR value decreased by slightly less than 9% when  $TH = 0$  and  $TH^* = 0$ . Similar results are obtained for the other images. Therefore, we come to the conclusion that, for common images, the proposed scheme provides a very large increase in information hiding capacity with only a small reduction in PSNR. Next, we use three groups of images, i.e., medical images, texture images, and aerial images, and each group has eight images to illustrate the relevant issues.

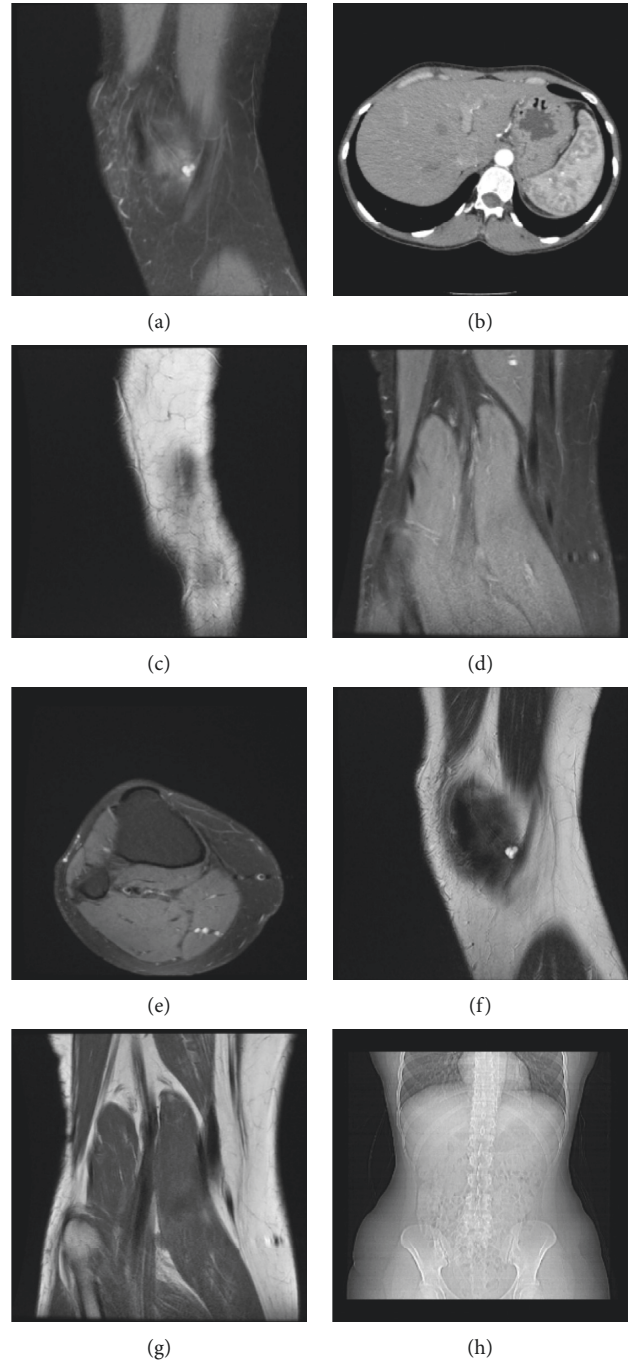


FIGURE 9: Eight medical images with sizes of  $512 \times 512$ .

Figure 9 shows eight medical images with sizes of  $512 \times 512$ , corresponding to these pictures; Table 3 shows us the payloads (bits), PSNR, and embedding rate (ER) with condition values of  $TH = 2$  and  $TH^* = 2$ . Table 3 indicates that the hiding capacities of all eight medical images are increased by about 95%, and their bpp (bits per pixel) values generally are greater than 3.8. In fact, most of the decrease rates in PSNR values are about 13%, which means that compared with the original algorithm [19], the proposed scheme provides significantly increased EC values, while PSNR values are reduced slightly. It is concluded that the

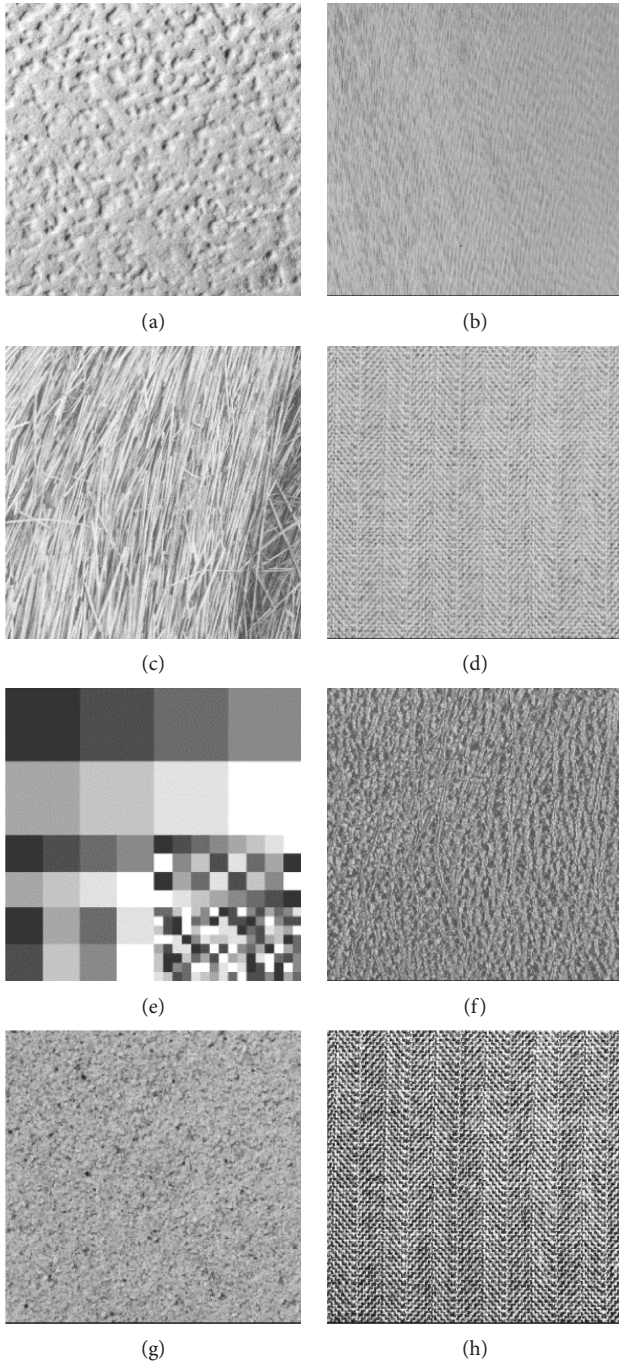


FIGURE 10: Eight texture images with sizes of  $512 \times 512$ .

proposed double-way prediction algorithm significantly increases storage capacity for secret messages and guarantees ideal ER and good PSNR values of stego medical images. Thus, the scheme is particularly suitable for medical images as cover images.

The texture images shown in Figure 10 illustrate the related information in Table 4. Since the structure of the texture of each image is different, the information hiding capacity of each image is also different. Some images, e.g., Figure 10(e), have large smooth blocks that make the histogram sharper and thus allow more information to be

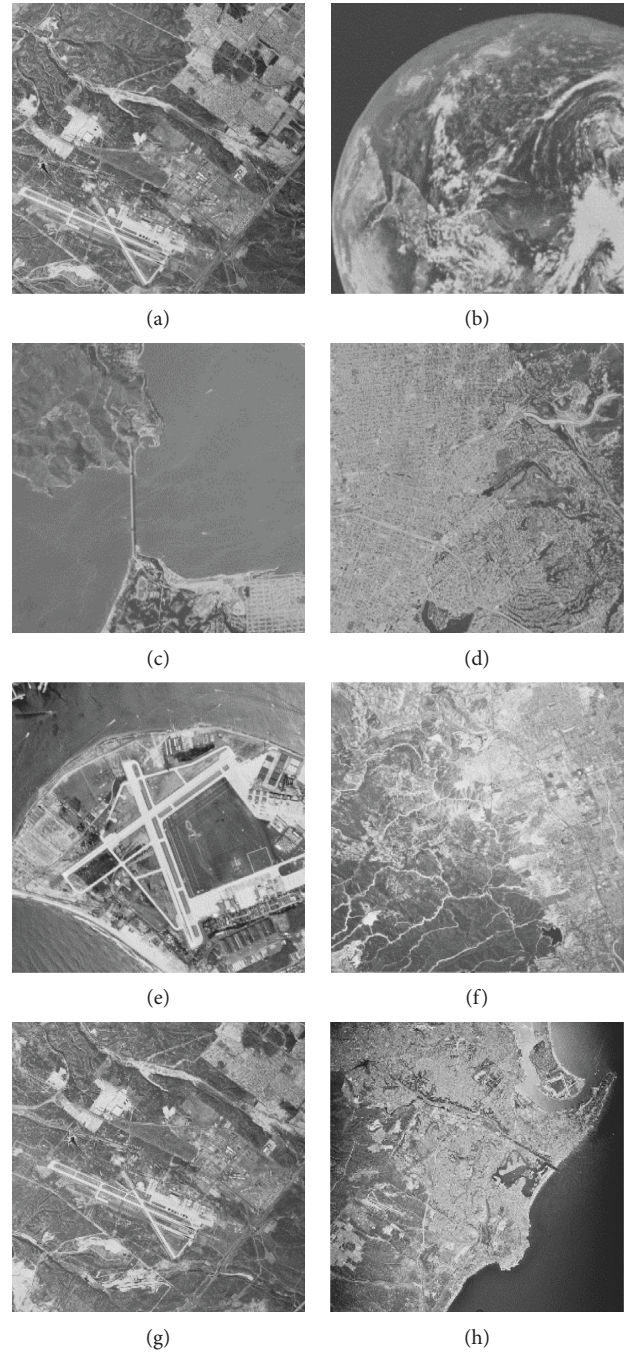


FIGURE 11: Eight aerial images with sizes of  $512 \times 512$ .

hidden, and its ER value is 2.95. However, Figure 10(h) has a complex and nonsmooth texture, and it produces a flatter histogram, thereby dramatically reducing the information that can be hidden, and its ER value is as low as 0.11. Thus, it is apparent that the proposed scheme will allow images with more smooth areas to hide much more information.

For the eight aerial images shown in Figure 11, the payload, PSNR, and ER of them are given in Table 5. From the experimental data, we can get to the conclusion that all of the increasing rates of payload are greater than 50% and all

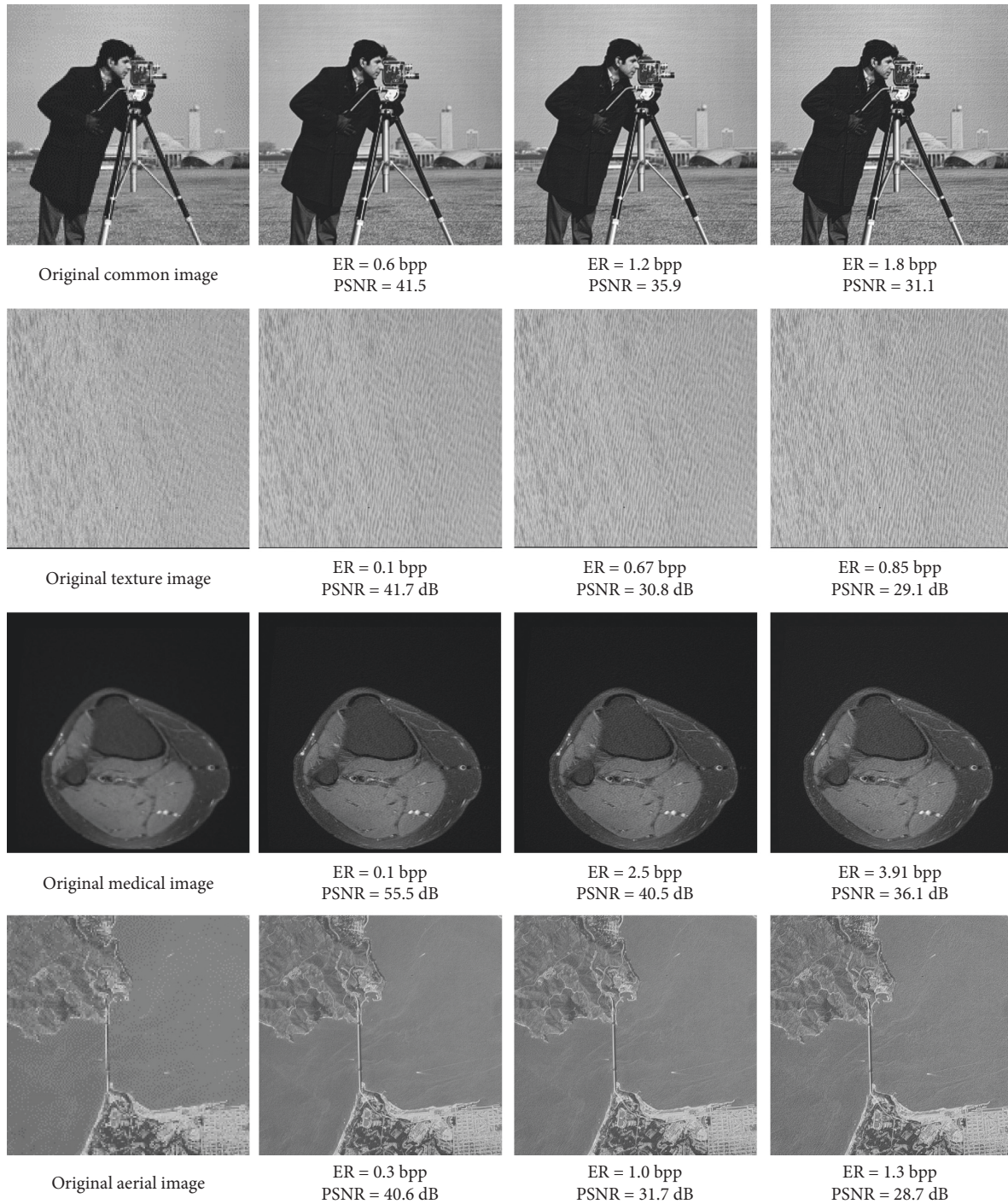


FIGURE 12: Four images of common, medical, texture, and aerial categories and their PSNR values with different ER.

of the decreasing rates of PSNR are about 10%. But the ER value here is a little bit low, being within 1.

From Figure 12, we can hardly see the visual differences of the stego images from common, medical, texture, and aerial categories with different ER and PSNR.

We use 108 images randomly selected from SIPI and Dicom image databases to get the experimental data, and

from Figure 13, we can see that ER is improved while the PSNR is decreased, but the increasing rate of ER is greater than the decreasing rate of PSNR obviously. Among them, Wang et al.'s algorithm has only one parameter,  $TH$ , and the proposed algorithm has two parameters,  $TH$  and  $TH^*$ . Therefore, ER and PSNR value curves of Wang et al.'s are not affected by the second parameter as the proposed



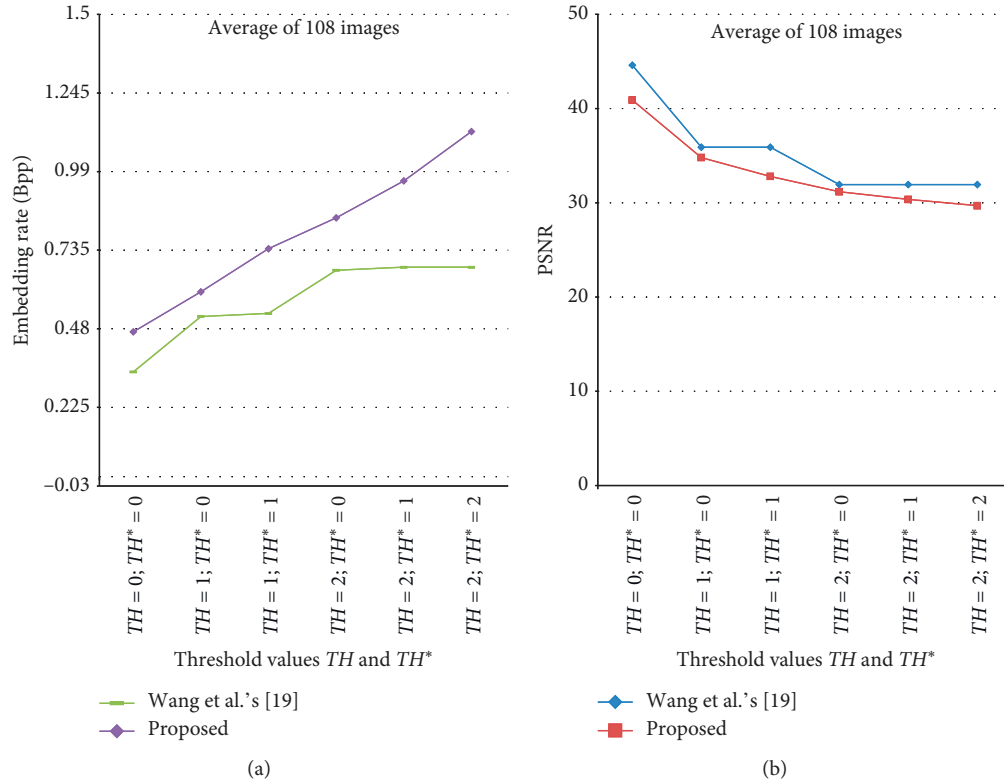


FIGURE 13: Average performance comparison for 108 test images randomly selected from SIPI and Dicom image databases at  $TH, TH^* = 0, 0, 1, 0, 1, 1, 2, 0, 2, 1, \text{ and } 2, 2$ , respectively.

scheme. Furthermore, the ER value of the proposed scheme is steadily increased, and when  $TH=2, TH^*=2$ , the average ER value of those 108 images is about 1.2 which is ideal for medium data payload. At the same time, the PSNR value is 29.7, making embedded information visually imperceptible.

## 5. Conclusions

In this paper, we proposed an improved, bidirectional shift-based reversible data hiding scheme using a double-way prediction strategy. The embedding process consists of two phases, i.e., (1) creation of the first histogram of prediction-error and right-left shifting with embedding and (2) creation of the second histogram of prediction-error and left-right shifting with embedding. Both of the extraction of the embedded message and the recovery of the original image can be realized exactly using the stego image and relevant information. After experiments on common, texture, medical, and aerial images, we found that the embedding rate of medical images increases the most. Whether it is medical images or other types of images, we can all come to the conclusion that the proposed scheme is significantly superior to the previous scheme in embedding capacity, with a little bit of time loss and PSNR decline, such that when  $TH = TH^* = 2$ , the algorithm effectively and significantly improves the information hiding rate by 65% but only reduces the PSNR value about 6%, and the PSNR value remains very close to 30 dB. In our future work, we will

further improve the visual quality with the increasing hiding rate.

## Data Availability

No external data were used to support this study. All derived data sets have been generated in our infrastructure.

## Disclosure

This research was performed as part of the employment of the authors.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Soft Science Foundation of Fujian Province, China (Grant no. B19085).




## References

- [1] D. A. Perez, *Dubrull Indice de Libros Prohibidos*, Vatican, Europe, 1880.
- [2] Typis Vaticanis, *Index Librorum Prohibitorum*, Vatican, Europe, 1900.
- [3] M. Barni, F. Bartolini, I. J. Cox, J. Hernandez, and F. Perez-Gonzalez, "Digital watermarking for copyright protection: a

- communications perspective," *IEEE Communications Magazine*, vol. 39, no. 8, pp. 90-91, 2001.
- [4] I. J. Cox, M. Miller, J. Bloom et al., *Digital Watermarking and Steganography*, 2nd edition, 2007, <https://www.elsevier.com/books/digitalwatermarkingnd-teganography/cox/978-12-1>.
- [5] C.-C. Lin, Y. Huang, and W.-L. Tai, "A novel hybrid image authentication scheme based on absolute moment block truncation coding," *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 463-488, 2017.
- [6] C. C. Chang, T. C. Lu, and Y. L. Liu, "A secret information hiding scheme based on switching tree coding," in *Computer Security in the 21st Century*, D. T. Lee, S. P. Shieh, and J. D. Tygar, Eds., Springer, Boston, MA, USA, 2005.
- [7] C.-C. Chang, C.-Y. Lin, and Y.-Z. Wang, "VQ image steganographic method with high embedding capacity using multi-way search approach," *Lecture Notes in Computer Science*, pp. 1058-1064, 2005.
- [8] M. Min Wu and B. Bede Liu, "Data hiding in image and video. I. Fundamental issues and solutions," *IEEE Transactions on Image Processing*, vol. 12, no. 6, pp. 685-695, 2003.
- [9] J. Barton, "Method and apparatus for embedding authentication information within digital data," US Patent 5,646,997, 1997.
- [10] C. W. Honsinger, P. W. Jones, M. Rabbani et al., "Lossless recovery of an original image containing embedded data," Google Patents, 2001.
- [11] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding—new paradigm in digital watermarking," *EURASIP Journal on Advances in Signal Processing*, vol. 2002, no. 2, 2002.
- [12] M. U. Celik, G. Sharma, A. M. Tekalp et al., "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 157-160, 2002.
- [13] Z. Ni, Y. Q. Shi, N. Ansari et al., "Reversible data hiding," in *Proceedings of the 2003 International Symposium on Circuits and Systems, ISCAS '03*, Bangkok, Thailand, May 2003.
- [14] J. Jun Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896, 2003.
- [15] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147-1156, 2004.
- [16] C.-C. Lin, W.-L. Tai, and C.-C. Chang, "Multilevel reversible data hiding based on histogram modification of difference images," *Pattern Recognition*, vol. 41, no. 12, pp. 3582-3591, 2008.
- [17] W. Tai, C. Yeh, and C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 906-910, 2009.
- [18] D. M. Thodi, J. J. Rodriguez, and S. Member, "Expansion Embedding Techniques for Reversible Watermarking," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 721-730, 2007.
- [19] W. Wang, J. Ye, and T. Wang, "A high capacity reversible data hiding scheme based on right-left shift," *Signal Processing*, vol. 150, pp. 102-115, 2018.
- [20] R. M. Wang, "The double exponential distribution: using calculus to find a maximum likelihood estimator," *The American Statistician*, vol. 38, no. 2, pp. 135-136, 1984.
- [21] D. Zou, Y. Q. Shi, and Z. Zhicheng Ni, "A semi-fragile lossless digital watermarking scheme based on integer wavelet transform," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 10, pp. 1294-1300, 2006.
- [22] Z. Wei Su, Y. Q. Shi, N. Ansari et al., "Robust lossless image data hiding designed for semi-fragile image authentication," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 4, pp. 497-509, 2008.
- [23] H. J. Kim, V. Sachnev, Y. Q. Shi et al., "A novel difference expansion transform for reversible data embedding," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 456-465, 2008.
- [24] L. Luo, Z. Chen, M. Chen et al., "Reversible image watermarking using interpolation technique," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 187-193, 2010.

## Research Article

# Using XGBoost to Discover Infected Hosts Based on HTTP Traffic

Weina Niu <sup>1</sup>, Ting Li,<sup>1</sup> Xiaosong Zhang <sup>1,2</sup>, Teng Hu <sup>1,3</sup>, Tianyu Jiang,<sup>1</sup> and Heng Wu<sup>4</sup>

<sup>1</sup>School of Computer Science and Engineering, Institute for Cyber Security,  
University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China

<sup>2</sup>Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen, Guangdong 518040, China

<sup>3</sup>Institute of Computer Application, China Academy of Engineering Physics, Mianyang, Sichuan 621900, China

<sup>4</sup>Glasgow College, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China

Correspondence should be addressed to Teng Hu; [mailhuteng@gmail.com](mailto:mailhuteng@gmail.com)

Received 26 April 2019; Revised 27 September 2019; Accepted 9 October 2019; Published 6 November 2019

Guest Editor: Mehdi Hussain

Copyright © 2019 Weina Niu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, the number of malware and infected hosts has increased exponentially, which causes great losses to governments, enterprises, and individuals. However, traditional technologies are difficult to timely detect malware that has been deformed, confused, or modified since they usually detect hosts before being infected by malware. Host detection during malware infection can make up for their deficiency. Moreover, the infected host usually sends a connection request to the command and control (C&C) server using the HTTP protocol, which generates malicious external traffic. Thus, if the host is found to have malicious external traffic, the host may be a host infected by malware. Based on the background, this paper uses HTTP traffic combined with eXtreme Gradient Boosting (XGBoost) algorithm to detect infected hosts in order to improve detection efficiency and accuracy. The proposed approach uses a template automatic generation algorithm to generate feature templates for HTTP headers and uses XGBoost algorithm to distinguish between malicious traffic and normal traffic. We conduct a performance analysis to demonstrate that our approach is efficient using dataset, which includes malware traffic from MALWARE-TRAFFIC-ANALYSIS.NET and normal traffic from UNSW-NB 15. Experimental results show that the detection speed is about 1859 HTTP traffic per second, and the detection accuracy reaches 98.72%, and the false positive rate is less than 1%.

## 1. Introduction

With the booming of the Internet and the popularity of computers, today's computers are facing serious security problems, whose biggest cause is the explosive growth of malicious code. The malicious code refers to a computer code that is intentionally written by individuals or organizations to pose a security risk to a computer or network. It usually contains malicious sharing software, adware Trojans, viruses, worms, etc., each of which has different kinds of variants [1–5]. In the first half of 2018, China Internet Security News from 360 Internet Security Center shows that a total of 140 million new malicious programs were intercepted and an average of 795,000 new malicious programs were intercepted every day. Among them, the number of malicious programs on the PC side was 14,098,000, and an

average of 779,000 new malicious programs were intercepted every day [6]. In the fourth quarter of 2017, McAfee Labs detected the highest number of new malware in history, with a total of 63.4 million new samples. McAfee Labs records an average of eight new malware samples per second, a significant increase from the four new samples recorded in the third quarter [7]. The malware not only brings huge economic losses to users, but also rapid changes have brought great trouble and pressure to the antiskilling technology of malicious programs. The current technology has been difficult to detect malware before the host is infected.

Based on this background, detecting malware-infected hosts in network traffic can make up for the shortcoming [8] because most malware will communicate with externally hosted command and control (C&C) servers using the HTTP protocol after infecting the device. The C&C server is

the control center that sends malware execution commands, and it is where malware collects data. After an attacker attacks the host with malware, the controlled host sends a connection request to the C&C server. The traffic generated by the connection is malicious external traffic. Currently, there are two main ways to detect malicious external traffic. One is to filter malicious domain names based on blacklists, and the other is to use rules to match malicious external traffic. Both of these solutions have certain limitations. The blacklist-based filtering scheme can only identify malicious external traffic when connecting to a known malicious website and has no perception of domain name changes. However, based on the feature detection scheme, it is necessary for the security practitioner to analyze the samples one by one, which consumes large manpower and is difficult to detect the malicious external connection traffic of the variant.

As a supplement to the prior art, malicious traffic can be detected through machine learning. Using machine learning to discover the commonality between malicious traffic and use it as a basis to detect malicious traffic, a good algorithm can greatly reduce the workload of security practitioners. Specifically, the contributions of this work are specified as follows:

- (1) We propose an approach-combined machine learning and HTTP header template to discover traffic involved in malware infection and develop it into the MalDetector system.
- (2) We use the statistical technique to aggregate similar features of HTTP header fields, which is also called HTTP header template, from large-scale network traffic.
- (3) We use the GridSearchCV function to coordinate the eXtreme Gradient Boosting (XGBoost) algorithm and verify their effectiveness in the dataset consisting of malicious external traffic generated from malicious samples from MALWARE-TRAFFIC-ANALYSIS.NET [9] running in the sandbox and the UNSW-NB 15 dataset [10].

The structure of this paper is arranged as follows. We introduce the related work in Section 2. Section 3 presents an overview of the proposed approach. The process of template automatic generation from the HTTP header is described in Section 4. Section 5 completes the experimental evaluation metrics and illustrates the experimental results. We make a conclusion of the paper in Section 6.

## 2. Related Work

At present, the malware traffic identification approach based on HTTP traffic mainly focuses on two aspects [11–24]; one is based on the request and response statistical features [11–16] and the other is based on the content of the HTTP packet [17–26].

*2.1. The Request and Response Statistical Features.* The approach mainly analyzes the behavior characteristics of HTTP

request/response time interval, quantity, and packet size to model malicious behavior and identify malware traffic. Perdisci et al. [11] developed a novel network-level behavioral malware clustering system. They performed coarse-grained clustering through statistical features, such as the total number of HTTP requests, the number of GET requests, the number of POST requests, the average length of the URLs, the average number of parameters in the request, the average amount of data sent by POST requests, and the average response length. Then, they performed fine-grained clustering by calculating the difference in URL structure between two malware samples. At last, they merged together fine-grained clusters of malware variants that behave similarly enough. Their work can be able to unveil similarities among malware samples that may not be captured by current system-level behavioral clustering systems. Ogawa et al. [12] extracted new features such as HTTP request interval, body size, and header bag-of-words from HTTP request/response pairs and calculated cluster appearance ratio per communication host pairs and identified malware originated communication host pairs. However, the identification approach based on the request and response statistical features is limited to malware samples that perform some interesting actions (i.e., malicious activities) during the execution time  $T$ . The identification approach based on the content of HTTP requests and responses can overcome this limitation.

*2.2. The Content of HTTP Packets.* The approach performs an analysis of the content of HTTP requests and responses, extracts relevant field information to process it, and combines machine learning algorithm to identify malware traffic. Zhang et al. [17, 18] used a learning-based approach to discover dependencies of network with the help of HTTP request features and thus detect malicious traffic. Srivastava et al. [19] developed a system called ExecScent that is closest to this work. They used all the HTTP header fields to detect botnet traffic. They manually created templates by themselves, such as URL-Path, Query, and User-Agent, and formatted them using regular expressions. Zhang et al. [20] proposed a method that used the User-Agent field to detect malicious external traffic generated by malware. They used regular expressions to format HTTP header information and used the operating system's fingerprint technology to identify whether it was a fake user agent domain to infer if there was a malware infection. Grill and Rehak [21] also used the User-Agent field to detect the presence of malicious external traffic. They found that all User-Agent field information can be divided into five categories: legitimate user browser information, null, specific, spoofed, and inconsistent. According to their findings, some malware deliberately forged requests that were sent from a web browser, making it difficult to detect malicious outbound traffic from the User-Agent field. Li et al. [22] proposed MalHunter based on behavior-related statistical characteristics. They detected malware communication patterns from three types of features: character distribution of the URL, HTTP header fields, and HTTP header sequence. However, these

approaches are either based on a single field or based on all fields, and their feature validity is low.

Moreover, Zhang et al. [23] presented a system SMASH that uses unsupervised data mining methods to detect various attack activities and malicious communication activities, focusing on detecting malicious HTTP activity from the perspective of server-side communication. Mekky et al. [24] put forward a method for identifying HTTP redirected malicious links. They built per-user chains from passively collected traffic and extracted new statistical features from them to capture the inherent characteristics of malicious redirect cases. The supervised decision tree classifier is then applied to identify malicious links. Liu et al. [25] proposed an identification approach by analyzing HTTP connections established by clients in a monitored network and combining stream classification with graph-based fractional propagation methods to identify previously undetected Internet Service Provider (ISP) networks.

### 3. HTTP-Based Infected Host Detection Approach

The proposed HTTP-based infected host detection system includes four modules: HTTP traffic filtering, header feature extraction, template automatic generation, and infected host detection. Figure 1 gives an overview of the framework of our proposed infected host detection approach using HTTP traffic.

*3.1. HTTP Traffic Filtering and Header Feature Extraction.* We save the HTTP header to reduce the amount of stored traffic. We also select the important information from the HTTP header for further analysis. The number of distinct HTTP header fields could be roughly 10 K. Moreover, some unrelated features may expose the machine learning model to the risk of overfitting. Rare fields are nonversatile, so the selection criteria are that we do not extract fields that appear less than 10 times or never appear in training data.

In addition, we mainly focus on the detection of malware that leverages the HTTP as the primary channel to communicate with the C&C server or to launch attack activities. Thus, our approach mainly focuses on HTTP requests rather than responses. If the C&C server is temporarily offline or changes its response content, there is little impact on our detection capabilities. Therefore, the selected fields are URI, Host, User-Agent, Request-Method, Request-Version, Accept, Accept-Encoding, Connection, Content-type, Cache-Control, Content-length, and some identification fields like Frame-time, srcIP (source IP), srcPort (source port), dstIP (destination IP), and dstPort (destination port).

Table 1 lists the description of the selected fields. The reason for selecting them is that they are often used in HTTP traffic and may be helpful in distinguishing legitimate traffic and malicious traffic.

*3.2. Template Automatic Generation.* When malware communicates with externally hosted C&C servers, malware developers typically use custom formats to construct

packets. The network traffic generated by the malware belonging to the same family usually has a similarity. Therefore, we use statistical techniques to aggregate similar features of the HTTP header fields, that is, to generate similar templates for malicious traffic, and then use the template to detect new malicious traffic. A template is a series of strings, the character part represents the same part of the value of an HTTP header field, and \* represents the different parts of the value of the header field. Templates are generated to display the variability of words constituting the HTTP header fields and aim to compress their information. The template automatic generate module consists of three steps: scoring, clustering, and generating templates [27], which is explained in detail in Section 4.

*3.3. Infected Host Detection.* Many winners in Kaggle’s competitions like to use XGBoost [28] due to Parallelization, Distributed Computing, Out-of-Core Computing, and Cache Optimization of data structures and algorithms. Thus, we use the XGBoost algorithm to classify malicious traffic and normal traffic in this work.

## 4. Template Automatic Generation

This section introduces focuses on how template automatic generation algorithm works.

*4.1. Scoring.* We first calculate the score for each value of the selected HTTP header fields by using the score calculation method, and then sort each selected HTTP header field’s values according to their scores. Each field in the HTTP header is divided by the following four separators: space, “/”, “=”, and “,”. Thus, the score calculation method is that we split each selected HTTP header field by separator and then calculate the percentage of their values’ scores. For a value  $w$  in the field  $F$ , its score is  $S(w; F)$ , which can be calculated using

$$S(w; F) = \frac{P(w | \text{pos}(w, F), \text{len}(F))}{n(\text{pos}(w, F), \text{len}(F))} \quad (1)$$

where  $\text{pos}(w, F)$  is the position of the value in the field  $F$ ,  $\text{len}(F)$  is the number of values in the field  $F$ . For example,  $F = \{\text{foo}, \text{bar}, \text{baz}, \text{quz}\}$ ,  $w = \text{bar}$ ,  $\text{pos}(w, F) = 2$ , and  $\text{len}(F) = 4$ .  $n(X)$  is the number of times that  $X$  appears in all the HTTP header,  $n(w, \text{pos}(w, F), \text{len}(F))$  represents the number of times that  $w$  appears in all the pos field of all data, and  $n(\text{pos}(w, F), \text{len}(F))$  indicates the number of times that the pos field appears in all the HTTP header. As shown in Figure 2, the score of “rv: 19.0” is 0.33 ( $S(w, F) = 1/3 = 0.33$ ).

*4.2. Clustering.* We use the idea of the DBSCAN [29, 30] algorithm to cluster the values of the selected HTTP header fields. In the selected HTTP header field, when the score of the next value differs from the score of the previous value by

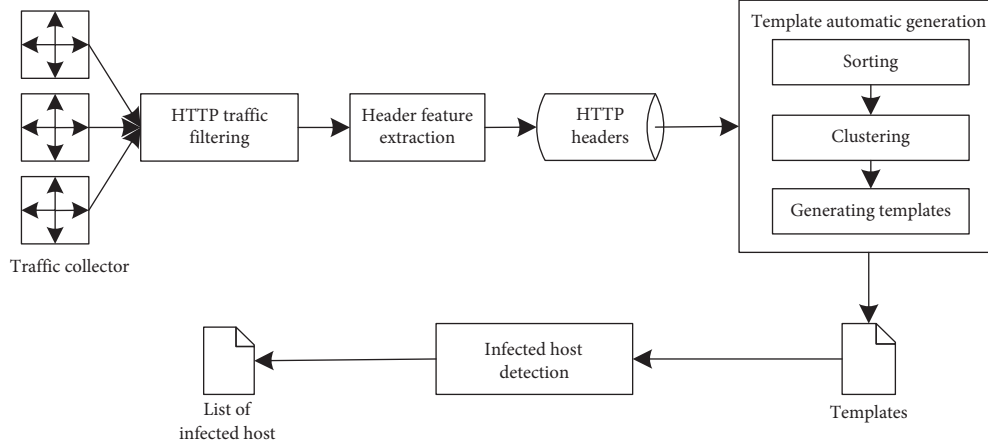


FIGURE 1: The framework of our proposed approach.

TABLE 1: The description of the selected fields in HTTP request header.

The selected fields name	Description
URI	URI is uniform resource identifier, a reference for resources available on the Internet such as HTML documents, images, or videos, and the URI field plays an important role in detection for malicious traffic [19, 22]
Host	The Host field recorded the domain name of the server and the TCP port number monitored by the server
User-Agent	The User-Agent field is still an effective indicator of compromised hosts because malware may carry the fake browser-like information or its own unique identification
Request-Method	Request type
Request-Version	HTTP protocol version
Accept	This field contains media type information and relative priority of media type
Accept-Encoding	The information in Accept-Encoding field is an encoding method of the content received by the client, and it is usually some kind of compression algorithm
Connection	The Connection field represents a connection state of the client and the server
Content-type	The value of the Content-type could help our model filter some legal traffic. The HTTP protocol carries data transmission of various types, such as text, pictures, sounds, videos, and others. Legal traffic tends to vary significantly. In contrast, most malware chooses text-related values such as text/html; charset = UTF-8
Cache-Control	Cache-Control message indicating a request caching mechanisms need to be implemented
Content-length	This field indicates the size of the entity-body

less than  $\delta$ , the next value is added as the current cluster; otherwise, the next value is added to the other clusters. Repeat the above process until all values have been added to the cluster. Here, the DBSCAN algorithm requires two parameters: scan radius (*eps*) and minimum inclusion points (*minPts*). The working process of the DBSCAN algorithm is as follows.

Starting with an unvisited point and finding all nearby points within the *eps* (including *eps*). If the number of nearby points is not smaller than *minPts*, the current point forms a cluster with its nearby points, and the starting point is marked as visited. Then recursively, all the points in the cluster that are not marked as visited are processed in the same way, thereby expanding the cluster. If the number of nearby points is smaller than *minPts*, the point is temporarily marked as a noise point. If the cluster is fully extended, i.e., all points within the cluster are marked as accessed, then the same algorithm is used to process the unvisited points.

Finally, we describe our clustering approach with the scoring method and DBSCAN algorithm in the following.

First, we need to introduce the following two parameters: ( $\delta \geq 0$ ) and  $\beta$  ( $0 < \beta < 1$ ),  $\delta$  is the minimum distance between two clusters,  $\beta \times \text{len}(F)$  for the minimum number of points in the cluster, and  $\text{len}(F)$  refers to the number of value in a field. In this work, the  $\delta$  is set to 0.1 and  $\beta$  is set to 0.5.

Then, we sorted each word in descending score. When the score of the next word differs from the mean score of a cluster by less than  $\delta$ , the next word is added to the current cluster. Otherwise, the next word is assigned to a new current cluster. This process is repeated until all words are included in either cluster.

**4.3. Generating Templates.** The results of the clustering are filtered to preserve only the clusters whose values are larger than  $\beta \times \text{len}(F)$  and the remaining clusters are replaced with “\*”, where  $\delta$  is the minimum distance between two clusters, whose value is not smaller than 0;  $\beta \times \text{len}(F)$  ( $0 < \beta < 1$ ) is the minimum number of points in the cluster, and  $\text{len}(F)$  is the number of values of the field. The overall generation process is shown in Figure 2.

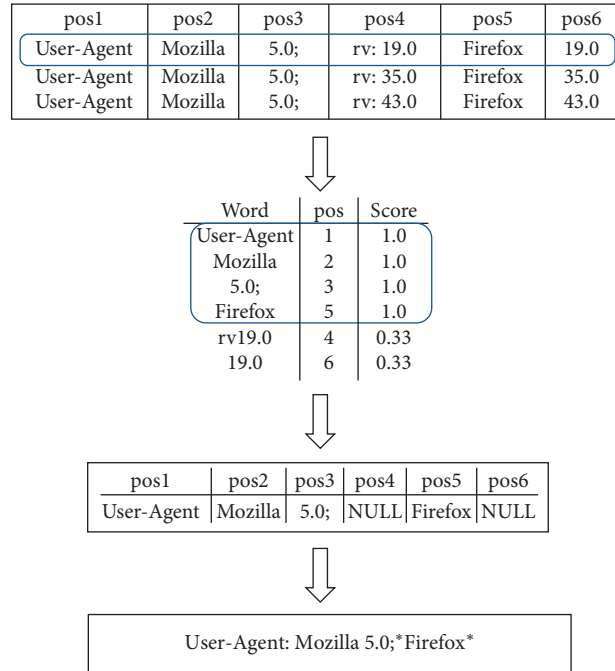


FIGURE 2: Template generation process.

The generated HTTP header field information and HTTP template are shown in Table 2.

We also performed statistics on the templates generated by the training data. The statistical results are shown in Figure 3.

As can be seen from Figure 3, the number of templates for malicious traffic is generally several times larger than the number of templates for normal traffic. The maximum number of templates generated is the URI and User-Agent fields. It can be inferred that malicious traffic may be distinguished mainly based on templates of these several fields. It has been observed that some fields do not even have the generation of malicious traffic templates. It can be inferred that the HTTP request information of malicious traffic may be short, including only information of several fields. Probably because normal HTTP request traffic is usually a connection made through a browser, the browser logs information for many fields. Malicious traffic is a connection made to the C&C server through malware, and the data format is usually constructed by a malware developer, so the HTTP request message is shorter.

## 5. Experiments and Results

This section introduces the dataset, the experimental setup, the performance metrics, and the obtained results.

**5.1. Dataset.** The malware traffic used in this work is from MALWARE-TRAFFIC-ANALYSIS.NET [9]. We collect malicious external traffic by running malicious samples collected from June 2013 to December 2017 in the sandbox and use SecurityOnion (a tool for network security monitoring) to detect traffic and get the result. The normal traffic samples are from the UNSW-NB 15 dataset shared

by the Cyber Range Lab of the Australian Cyber Security Center (ACCS) in 2015 [10]. They used the tcpdump tool to capture 100 GB of raw traffic (PCAP files) for evaluating network intrusion detection systems and gave a labeled dataset. The labeled file contains the time period, the source port, the source IP address, the destination port, the destination IP address, the protocol type, and other information of the threat traffic, which is shown in Table 3. There are 373864 HTTP request records and only 6401 malicious traffic records in the 100G raw traffic data. We remove malicious HTTP traffic based on source IP, destination IP, source port, destination port, and the time period (from the start time to the last time) in the given labeled file. When the protocol type is HTTP and the time period, source port, source IP, destination port and destination IP address are matched successfully, the traffic is labeled as malicious traffic.

We set the ratio of the training set to the testing data as 7:3. Thus, the dataset in the experiment is shown in Table 4, which consists of 34,239 malicious HTTP requests and 35,481 normal HTTP requests.

**5.2. Experimental Setup.** The system had been implemented in Python 3.5, and all experiments were performed using an off-the-shelf server with 64 GB of RAM memory and 6-core processor. In order to evaluate the true positive rates and false positive rates of our detection approach, we tune the model parameters on the training set. The initial key parameters of the XGBoost model are shown in Table 5.

Table 5 shows that the accuracy of cross-validation of the training set with the initial parameters is 99.5%, but the accuracy of the testing set is only 92.89% due to over-fitting.

TABLE 2: HTTP header field information and template comparison.

HTTP header field information	Generated templates
Accept: Text json	Accept: Text *
Accept-Encoding: Gzip deflate	Accept-Encoding: Gzip *
Connection: Keep-Alive	Connection: *
User-Agent: Mozilla 4.0 (compatible; MSIE 6.0; Windows NT 5.1)	User-Agent: Mozilla * (compatible; MSIE * Windows NT *

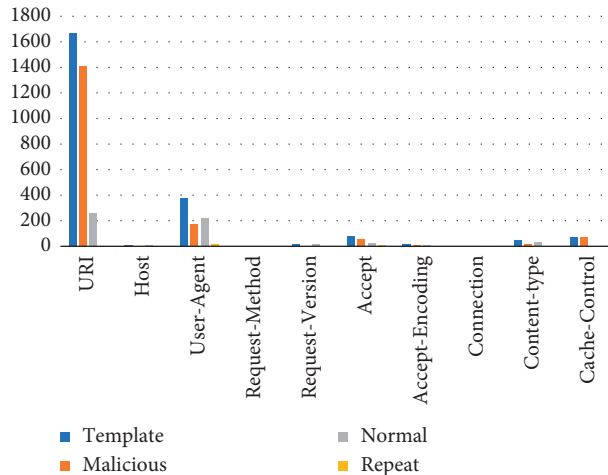


FIGURE 3: Template statistical histogram.

TABLE 3: The labeled file of UNSW-NB 15.

Start time	Last time	Attack category	Attack subcategory	Protocol	Src IP	Src port	Dst IP	Dst port
1421927415	1421927415	Exploits	Unix'r' Service	udp	175.45.176.3	21223	149.171.126.18	32780
1421927416	1421927415	Exploits	Brower	tcp	175.45.176.2	23357	149.171.126.16	80
1421927418	1421927415	Exploits	Cisco IOS	tcp	175.45.176.4	26939	149.171.126.10	80
1421927420	1421927415	DoS	IXIA	tcp	175.45.176.1	23910	149.171.126.15	80
1421927421	1421927415	Generic	Brower	tcp	175.45.176.2	23909	149.171.126.14	3000

TABLE 4: Dataset in the experiment.

Dataset	Traffic type	The number of HTTP request
Training set	Malicious traffic	27789
	Normal traffic	28915
Testing data	Malicious traffic	6450
	Normal traffic	6566

In order to further improve the accuracy of the prediction, we further adjust the parameters of the XGBoost algorithm.

We use the GirdSerachCV function in the SCIKIT-learn [31] package to adjust the parameters, which traverses the value range of parameters. We adjust three of the key parameters, and the adjustment steps are as follows:

- (1) We first adjust two parameters  $max\_depth$  and  $min\_child\_weight$  that play a decisive role in the model. The value range of  $max\_depth$  is set to [4, 6, 8, 10, 12]. The value range of  $min\_child\_weight$  is very large and seriously affects the experimental results. If  $min\_child\_weight$  is over-fitting, the value of  $min\_child\_weight$

should be increased. Thus, its value range is set to [1, 10, 100, 1000]. The results of the parameter adjustment are shown in Table 6. The experimental results show that the model performs optimally when  $max\_depth = 10$  and  $min\_child\_weight = 1$ .

- (2) Based on the adjusted  $max\_depth$  and  $min\_child\_weight$  parameters, we adjust the parameter  $gamma$ , which participates in the pruning of the decision tree. The larger the value of the parameter is, the less the impact on the model is. Here, we set the value range of  $gamma$  to [0~8]. The results of the parameter adjustment are shown in Table 7. The experimental results show that the model with the best performance when  $gamma = 0$ .
- (3) We adjust the two parameters  $subsampling$  and  $colsample\_bytree$  at last, which is related to the proportion of samples used. If the sampling setting ratio is too small, the accuracy may be reduced. Here, the value range of the  $subsampling$  is set to [0.7~1], and the value range of  $colsample\_bytree$  is also to [0.7~1]. The



TABLE 5: Experimental parameters settings.

Parameter	Description	Value
<i>booster</i>	Tree model	gbtree
<i>gamma</i>	For pruning	1
<i>max_depth</i>	Depth of decision tree	12
<i>scale_pos_weight</i>	Balance positive and negative sample weights	1
<i>subsample, colsample_bytree</i>	Proportion of random collected samples each time	1
<i>min_child_weight</i>	Number of leaf nodes in the decision tree	1000
<i>eta</i>	Learning rate	0.1

TABLE 6: Tuning results of *max\_depth* and *min\_child\_weight*.

<i>max_depth</i>	<i>min_child_weight</i>	Auc
4	1	0.99826
4	10	0.99770
4	100	0.99667
4	1000	0.99535
6	1	0.99830
6	10	0.99782
6	100	0.99726
6	1000	0.99561
8	1	0.99838
8	10	0.99799
8	100	0.99780
8	1000	0.99574
10	1	0.99854
10	10	0.99827
10	100	0.99798
10	1000	0.99576
12	1	0.99852
12	10	0.99818
12	100	0.99806
12	1000	0.99576

TABLE 7: Tuning results of *gamma*.

<i>gamma</i>	Auc
0	0.99925
1	0.99854
2	0.99820
3	0.99793
4	0.99776
5	0.99771
6	0.99702
7	0.99689
8	0.99656

results of the parameter adjustment are shown in Table 8. The experimental results show that the model performs best when *subsample* = 0.8 and *colsample\_bytree* = 0.8.

5.3. *Evaluation Metrics.* The evaluation metrics of our proposed infected host detection approach using malicious external HTTP traffic are expressed as follows: TP refers to the number of malicious HTTP requests that are recognized as malware HTTP requests, TN indicates that the number of normal HTTP requests that are recognized as normal HTTP requests, FP refers to the number of normal HTTP requests that have been mistaken for malware HTTP requests, and

TABLE 8: Tuning results of *subsample* and *colsample\_bytree*.

<i>colsample_bytree</i>	<i>subsample</i>	Auc
0.7	0.7	0.99793
0.7	0.8	0.99799
0.7	0.9	0.99782
0.7	1	0.99774
0.8	0.7	0.99912
0.8	0.8	0.99926
0.8	0.9	0.99891
0.8	1	0.99857
0.9	0.7	0.99802
0.9	0.8	0.99840
0.9	0.9	0.99846
0.9	1	0.99793
1	0.7	0.99789
1	0.8	0.99821
1	0.9	0.99844
1	1	0.99817

FN indicates that the number of normal HTTP requests that are incorrectly identified as malware HTTP requests. The higher the value of precision, recall, and *F1*, the better the recognition effect of the infected host detection approach.

- (1)  $ACC = (TP + TN) / (TP + TN + FP + FN)$
- (2) ROC curve whose horizontal axis is FRP and vertical axis is TRP, where  $FPR = FP / (TN + FP)$  and  $TPR = TP / (TP + FN)$
- (3) PRC curve whose vertical axis is precision and horizontal axis is recall, where precision ( $P$ ) =  $TP / (TP + FP)$  and recall ( $R$ ) =  $TP / (TP + FN)$
- (4)  $F1 = (2 * P * R) / (P + R)$

5.4. *Experimental Results.* When the ratio of the number of HTTP requests in the training set and testing set is 7 : 3, the experimental results are shown in Table 9.

The accuracy of the testing set is 98.72%, and the false positive rate is less than 1%. The total testing time is about 7 s. Therefore, the proposed approach can quickly detect the network traffic and conclude whether the host is infected by malware so that the user can respond to the action as soon as possible. The PRC curve matching the threshold is shown in Figure 4. It can be seen that the algorithm has maintained a high precision with the increase of the recall rate. Finally, 0.8 is selected as the matching threshold. At this time, the accuracy of the algorithm is 93.56%, the recall rate is 97.14%, and the *F*-value is 0.9532.

TABLE 9: The experimental results when the ratio of the number of HTTP requests in the training set and testing set is 7 : 3.

Best iteration	218
Train-auc	99.8944%
Cross-validation-auc	99.8599%
Test-auc	98.726487%
Cost time	7.28223S

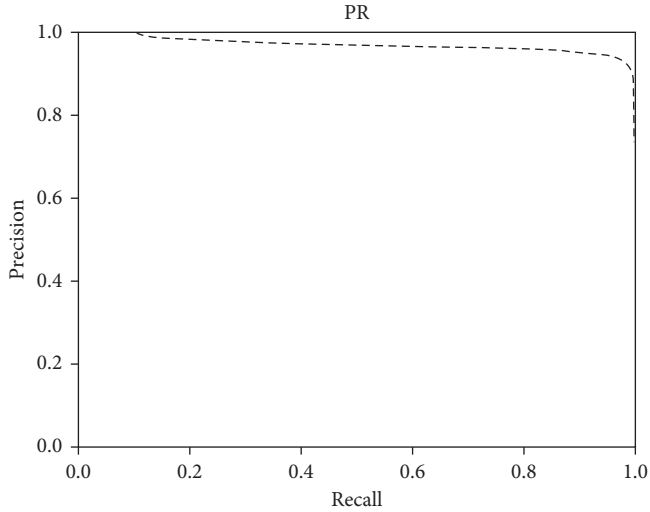


FIGURE 4: PRC curve of the detection approach.

To better validate our proposed approach, we also compare our approach to the other two methods of Ogawa et al. [12] and Li et al. [22]. We reproduced these two comparison experiments using our own data set. The experimental results are shown in Table 10.

Table 10 shows that the ACC,  $P$ ,  $R$ , and  $F1$  of our proposed approach are the largest, and they are 0.9827, 0.9356, 0.9714, and 0.9532, respectively. Therefore, our proposed approach using XGBoost and HTTP header statistical template is better to detect HTTP malware traffic than the method that uses HTTP header combined machine learning. The main reason is that Ogawa et al.'s approach and Li et al.'s approach are either based on a single field or based on all fields, their feature validity is low. Our proposed approach uses statistical techniques to aggregate similar features of the malicious HTTP header fields. Thus, our approach can more effectively characterize malware traffic characteristics, which can further improve the accuracy of malware HTTP traffic recognition.

In addition, we select 10%, 20%, 30%, . . . , 90% of the samples as the training set and set the matching threshold to 0.8 to test other sample data. The correct rate and false positive rate of malicious traffic and normal traffic are separately measured, whose results are shown in Figure 5. It can be seen that the detection rate of the normal HTTP requests has been maintained above 99%. For malicious samples, the detection accuracy rate is based on the diversity of the model. In the case that the training set is only 10% and the model data is insufficient, the algorithm can still detect 77.65% of malicious traffic, indicating that the algorithm has better generalization ability for malicious traffic variants.

TABLE 10: The experimental result of comparative testing.

Approach	ACC (%)	$P$ (%)	$R$ (%)	$F1$ (%)
Our approach	98.72	93.56	97.14	95.32
Ogawa et al.'s approach	96.25	92.84	94.99	93.9
Li et al.'s approach	97.17	93.25	96.19	94.7

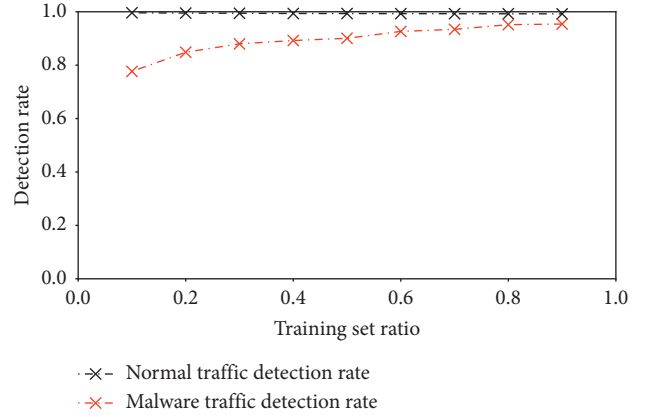


FIGURE 5: The impact of different ratios between the training set and the testing set.

TABLE 11: The experimental result under different malware traffic ratios.

Malicious traffic ratio in dataset (%)	ACC (%)	$P$ (%)	$R$ (%)	$F1$ (%)
40	94.81	94.58	92.90	93.57
30	91.99	87.41	92.11	89.70
20	97.16	95.87	87.03	92.02
10	97.52	94.04	78.14	85.36

We also change the malicious traffic and normal traffic ratio in our training set and testing set. The experimental results are shown in Table 11.

The accuracy rates under different malware traffic ratios all remained above 90%. However, the model has high precision but a low recall rate when malicious traffic accounted for 10% and 20%, respectively. The main reason is that the proportion of malicious traffic is too small, resulting in insufficient training of the model. The results show that if we want to build a machine learning model which can correctly identify malicious traffic, the proportion of malicious traffic and the normal flow ratio needs to be maintained at a relative balance. Malicious traffic accounts for less than 1% of the data in real-world samples. Thus, it is necessary to further process the sample, such as subsampling or oversampling, to increase the proportion of malicious traffic, thereby improving detection accuracy.

**5.5. MalDetector System Testing.** We also use the malicious traffic samples that do not exist in the training data and testing data to verify if the system has the ability to detect new malware and its variants. The selected malicious traffic

	Mal-Check	Date/Time	SrcIP	SrcPort	DstIP	stPo	URI	Host	User-Agent	Reque
1		1520639820	10.3.9.101	49159	169.255.59.27	80		sir-iyke.com	Mozilla/4.08 (Charon; Inferno)	POST
2		1520639821	10.3.9.101	49160	169.255.59.27	80		sir-iyke.com	Mozilla/4.08 (Charon; Inferno)	POST
3		1520639882	10.3.9.101	49162	169.255.59.27	80		sir-iyke.com	Mozilla/4.08 (Charon; Inferno)	POST
4		1520639943	10.3.9.101	49166	169.255.59.27	80		sir-iyke.com	Mozilla/4.08 (Charon; Inferno)	POST
5		1520640004	10.3.9.101	49168	169.255.59.27	80		sir-iyke.com	Mozilla/4.08 (Charon; Inferno)	POST
6		1520640065	10.3.9.101	49169	169.255.59.27	80		sir-iyke.com	Mozilla/4.08 (Charon; Inferno)	POST
7		1520640127	10.3.9.101	49170	169.255.59.27	80		sir-iyke.com	Mozilla/4.08 (Charon; Inferno)	POST

(a)

	Mal-Check	Date/Time	SrcIP	SrcPort	DstIP	stPo	URI	Host	User-Agent	Reque
1	✘	1520639820	10.3.9.101	49159	169.255.59.27	80		sir-iyke.com	Mozilla/4.08 (Charon; Inferno)	POST
2	✘	1520639821	10.3.9.101	49160	169.255.59.27	80		sir-iyke.com	Mozilla/4.08 (Charon; Inferno)	POST
3	✘	1520639882	10.3.9.101	49162	169.255.59.27	80		sir-iyke.com	Mozilla/4.08 (Charon; Inferno)	POST
4	✘	1520639943	10.3.9.101	49166	169.255.59.27	80		sir-iyke.com	Mozilla/4.08 (Charon; Inferno)	POST
5	✘	1520640004	10.3.9.101	49168	169.255.59.27	80		sir-iyke.com	Mozilla/4.08 (Charon; Inferno)	POST
6	✘	1520640065	10.3.9.101	49169	169.255.59.27	80		sir-iyke.com	Mozilla/4.08 (Charon; Inferno)	POST
7	✘	1520640127	10.3.9.101	49170	169.255.59.27	80		sir-iyke.com	Mozilla/4.08 (Charon; Inferno)	POST

(b)

FIGURE 6: Loki-Bot traffic and the detection result of MalDetector.

	Mal-Check	Date/Time	SrcIP	SrcPort	DstIP	DstPort	URI	Host	User-Agent
1		1523493460	10.4.12.101	49165	69.163.216.29	80		innervation.com	Mozilla/5.0 (Windows NT 6.1; ...)
2		1523493486	10.4.12.101	49172	157.7.188.210	80		ninestars.jp	
3		1523493497	10.4.12.101	49174	167.114.1.241	4143		167.114.1.241:...	Mozilla/4.0 (compatible; MSIE ...)
4		1523493510	10.4.12.101	49175	167.114.1.241	4143		167.114.1.241:...	Mozilla/4.0 (compatible; MSIE ...)
5		1523493511	10.4.12.101	49175	167.114.1.241	4143		167.114.1.241:...	Mozilla/4.0 (compatible; MSIE ...)

(a)

	Mal-Check	Date/Time	SrcIP	SrcPort	DstIP	DstPort	URI	Host	User-Agent
1	✘	1523493460	10.4.12.101	49165	69.163.216.29	80		innervation.com	Mozilla/5.0 (Windows NT 6.1; ...)
2	✘	1523493486	10.4.12.101	49172	157.7.188.210	80		ninestars.jp	
3	✘	1523493497	10.4.12.101	49174	167.114.1.241	4143		167.114.1.241:...	Mozilla/4.0 (compatible; MSIE ...)
4	✘	1523493510	10.4.12.101	49175	167.114.1.241	4143		167.114.1.241:...	Mozilla/4.0 (compatible; MSIE ...)
5	✘	1523493511	10.4.12.101	49175	167.114.1.241	4143		167.114.1.241:...	Mozilla/4.0 (compatible; MSIE ...)

(b)

FIGURE 7: Emotet traffic and the detection result of MalDetector.

samples have the same source as the training data, both of which are MALWARE-TRAFFIC-ANALYSIS.NET.

**5.5.1. Loki-Bot.** Loki-Bot [32] uses a malicious website to push fake “Adobe Flash Player,” “APK Installer,” “System Update,” “Adblock,” “Security Certificate,” and other application updates to induce user installation. The Loki-Bot malware is a bank hijacking Trojan, a variant of the BankBot Trojan. The traffic sample of running Loki-Bot and the testing result using MalDetector are shown in Figure 6. The experimental results show that MalDetector detects all the malicious HTTP traffic of Loki-Bot.

**5.5.2. Emotet.** Emotet [33] is a new type of banking Trojan in Germany. The sample flow is a new variant of Emotet that appeared in September 2017. It has its own ability to evade safety detection and cannot be recognized by antivirus software. The traffic sample of running Emotet and the testing result using MalDetector are shown in Figure 7. The experimental results show that MalDetector detects all the malicious HTTP traffic of Emotet.

## 6. Conclusion

The diversification of malware and the complication of its technologies have brought new challenges to cybersecurity. Unfortunately, rule-based traditional malware traffic detection methods are unable to detect malware variants. Machine learning-based methods can make up for this defect, and most malware uses the HTTP protocol to send malicious external traffic to the C&C server. Thus, we propose an approach to detect infected hosts using HTTP traffic combined with a machine learning algorithm. We mainly extract the common templates for the HTTP traffic header, so it still works for the traffic generated by the confusing malware. We also use the most popular XGBoost algorithm to detect infected hosts, which has the advantages of high efficiency and high accuracy. The experimental results show that the accuracy of the method reaches 98.72% and the false positive rate is less than 1%, where the experimental data is from MALWARE-TRAFFIC-ANALYSIS.NET and UNSW-NB 15. We also used two real samples that are Loki-Bot and Emotet to verify the effectiveness of the MalDetector system. We plan to combine the approach with malware dynamic analysis to further improve its detection accuracy in the future. Furthermore, some malware utilizes HTTPS to hide its content from the analyzer so that it further reduces detection possibility. Because the header information of HTTPS traffic has been encrypted, our method cannot be applied. We will consider new fields and combine with DNS traffic to refine the templates to detect anomaly-based malware infection in the future.

## Data Availability

The experimental data were collected and synthesized by ourselves. It has not been published online yet.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the National Key Research and Development Project (Grant no. 2016QY04W0800), the National Defense Innovation Special Zone Program of Science and Technology (Grant no. JG2019055), and the National Natural Science Foundation of China (Grant nos. 61902262 and 61572115).

## References

- [1] D. Zhao, I. Traore, B. Sayed et al., “Botnet detection based on traffic behavior analysis and flow intervals,” *Computers & Security*, vol. 39, pp. 2–16, 2013.
- [2] A. Saracino, D. Sgandurra, G. Dini, and F. Martinelli, “MADAM: effective and efficient behavior-based android malware detection and prevention,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 1, pp. 83–97, 2016.
- [3] Y. Yu, J. Long, and Z. Cai, “Network intrusion detection through stacking dilated convolutional autoencoders,” *Security and Communication Networks*, vol. 2017, Article ID 4184196, 10 pages, 2017.
- [4] G. Zhao, K. Xu, L. Xu, and B. Wu, “Detecting APT malware infections based on malicious DNS and traffic analysis,” *IEEE Access*, vol. 3, pp. 1132–1142, 2015.
- [5] A. Souri and R. Hosseini, “A state-of-the-art survey of malware detection approaches using data mining techniques,” *Human-Centric Computing and Information Sciences*, vol. 8, no. 1, p. 3, 2018.
- [6] 360 Internet Security Center, China Internet Security Report for the Third Quarter of 2017, 2017.
- [7] McAfee Labs, McAfee Labs Threats Report: December 2017, McAfee, Santa Clara, CA, USA, 2017.
- [8] X. Hu, J. Jang, M. P. Stoecklin et al., “BAYWATCH: robust beaconing detection to identify infected hosts in large-scale enterprise networks,” in *Proceedings of the 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 479–490, Toulouse, France, June 2016.
- [9] Malware-Traffic-Analysis.net, <http://malware-traffic-analysis.net/>.
- [10] The UNSW-NB15 Data Set, <https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/>.
- [11] R. Perdisci, W. Lee, and N. Feamster, “Behavioral clustering of HTTP-based malware and signature generation using malicious network traces,” in *Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI '10)*, vol. 10, p. 14, San Jose, CA, USA, April 2010.
- [12] H. Ogawa, Y. Yamaguchi, H. Shimada et al., “Malware originated http traffic detection utilizing cluster appearance ratio,” in *Proceedings of the 2017 International Conference on Information Networking (ICOIN)*, pp. 248–253, IEEE, Da Nang, Vietnam, January 2017.
- [13] M. Piskozub, R. Spolaor, and I. Martinovic, “MalAlert: detecting malware in large-scale network traffic using statistical features,” *ACM Sigmetrics Performance Evaluation Review*, vol. 46, no. 3, pp. 151–154, 2019.

- [14] N. Moustafa, B. Turnbull, and K. K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2019.
- [15] A. Liu, Z. Chen, S. Wang, L. Peng, C. Zhao, and Y. Shi, "A fast and effective detection of mobile malware behavior using network traffic," in *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*, pp. 109–120, Springer, Guangzhou, China, November 2018.
- [16] M. Yeo, Y. Koo, Y. Yoon et al., "Flow-based malware detection using convolutional neural network," in *Proceedings of the 2018 International Conference on Information Networking (ICOIN)*, pp. 910–913, IEEE, Chiang Mai, Thailand, January 2018.
- [17] H. Zhang, D. D. Yao, and N. Ramakrishnan, "Detection of stealthy malware activities with traffic causality and scalable triggering relation discovery," in *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (ASIA CCS '14)*, pp. 39–50, Kyoto, Japan, June 2014.
- [18] H. Zhang, D. D. Yao, and N. Ramakrishnan, "Causality-based sensemaking of network traffic for android application security," in *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security (ALSec '16)*, pp. 47–58, Madrid, Spain, April 2016.
- [19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [20] Y. Zhang, H. Mekky, Z.-L. Zhang et al., "Detecting malicious activities with user-agent-based profiles," *International Journal of Network Management*, vol. 25, no. 5, pp. 306–319, 2015.
- [21] M. Grill and M. Rehak, "Malware detection using http user-agent discrepancy identification," in *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 221–226, Atlanta, Georgia, December 2014.
- [22] K. Li, R. Chen, L. Gu et al., "A method based on statistical characteristics for detection malware requests in network traffic," in *Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 527–532, Guangzhou, China, June 2018.
- [23] J. Zhang, S. Saha, G. Gu et al., "Systematic mining of associated server herds for malware campaign discovery," in *Proceedings of the 2015 IEEE 35th International Conference on Distributed Computing Systems*, pp. 630–641, Columbus, OH, USA, June 2015.
- [24] H. Mekky, R. Torres, Z. L. Zhang et al., "Detecting malicious http redirections using trees of user browsing activity," in *Proceedings of the IEEE Conference on Computer Communications (IEEE INFOCOM 2014)*, pp. 1159–1167, Toronto, Canada, April 2014.
- [25] L. Liu, S. Saha, R. Torres et al., "Detecting malicious clients in isp networks using http connectivity graph and flow information," in *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pp. 150–157, Beijing, China, August 2014.
- [26] K. Cabaj, M. Gregorczyk, and W. Mazurczyk, "Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics," *Computers & Electrical Engineering*, vol. 66, pp. 353–368, 2018.
- [27] S. Mizuno, M. Hatada, T. Mori, and S. Goto, "Botdetector: a robust and scalable approach toward detecting malware-infected devices," in *Proceedings of the 2017 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, Paris, France, May 2017.
- [28] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, New York, NY, USA, August 2016.
- [29] K. M. Kumar and A. R. M. Reddy, "A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method," *Pattern Recognition*, vol. 58, pp. 39–48, 2016.
- [30] A. Malhotra and K. Bajaj, "A hybrid pattern based text mining approach for malware detection using DBScan," *CSI Transactions on ICT*, vol. 4, no. 2–4, pp. 141–149, 2016.
- [31] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "SCIKIT-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [32] D. Rendell, "Understanding the evolution of malware," *Computer Fraud & Security*, vol. 2019, no. 1, pp. 17–19, 2019.
- [33] H. Huang, H. Deng, J. Chen, L. Han, and W. Wang, "Automatic multi-task learning system for abnormal network traffic detection," *International Journal of Emerging Technologies in Learning (ijET)*, vol. 13, no. 4, 2018.

## Research Article

# VPN Traffic Detection in SSL-Protected Channel

**Muhammad Zain ul Abideen , Shahzad Saleem , and Madiha Ejaz**

*School of Electrical Engineering and Computer Science (SEecs), National University of Sciences and Technology (NUST), Islamabad, Pakistan*

Correspondence should be addressed to Muhammad Zain ul Abideen; [mabideen.msis18seecs@seecs.edu.pk](mailto:mabideen.msis18seecs@seecs.edu.pk)

Received 24 May 2019; Revised 4 September 2019; Accepted 16 September 2019; Published 29 October 2019

Guest Editor: Rajkumar Soundrapandiyan

Copyright © 2019 Muhammad Zain ul Abideen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent times, secure communication protocols over web such as HTTPS (Hypertext Transfer Protocol Secure) are being widely used instead of plain web communication protocols like HTTP (Hypertext Transfer Protocol). HTTPS provides end-to-end encryption between the user and service. Nowadays, organizations use network firewalls and/or intrusion detection and prevention systems (IDPS) to analyze the network traffic to detect and protect against attacks and vulnerabilities. Depending on the size of organization, these devices may differ in their capabilities. Simple network intrusion detection system (NIDS) and firewalls generally have no feature to inspect HTTPS or encrypted traffic, so they rely on unencrypted traffic to manage the encrypted payload of the network. Recent and powerful next-generation firewalls have Secure Sockets Layer (SSL) inspection feature which are expensive and may not be suitable for every organizations. A virtual private network (VPN) is a service which hides real traffic by creating SSL-protected channel between the user and server. Every Internet activity is then performed under the established SSL tunnel. The user inside the network with malicious intent or to hide his activity from the network security administration of the organization may use VPN services. Any VPN service may be used by users to bypass the filters or signatures applied on network security devices. These services may be the source of new virus or worm injected inside the network or a gateway to facilitate information leakage. In this paper, we have proposed a novel approach to detect VPN activity inside the network. The proposed system analyzes the communication between user and the server to analyze and extract features from network, transport, and application layer which are not encrypted and classify the incoming traffic as malicious, i.e., VPN traffic or standard traffic. Network traffic is analyzed and classified using DNS (Domain Name System) packets and HTTPS- (Hypertext Transfer Protocol Secure-) based traffic. Once traffic is classified, the connection based on the server's IP, TCP port connected, domain name, and server name inside the HTTPS connection is analyzed. This helps in verifying legitimate connection and flags the VPN-based traffic. We worked on top five freely available VPN services and analyzed their traffic patterns; the results show successful detection of the VPN activity performed by the user. We analyzed the activity of five users, using some sort of VPN service in their Internet activity, inside the network. Out of total 729 connections made by different users, 329 connections were classified as legitimate activity, marking 400 remaining connections as VPN-based connections. The proposed system is lightweight enough to keep minimal overhead, both in network and resource utilization and requires no specialized hardware.

## 1. Introduction

To enable the communication between the computers, TCP/IP stack was implemented. The stack was implemented without the consideration of security of information being transferred in the communication [1]. This issue raised a lot of security concerns which are constantly managed by different security services [2]. Secure Sockets Layer (SSL) is

commonly used to provide authentication and encryption security service in TCP/IP stack [3].

The trend of encrypted traffic in the network has largely increased in the last decade due to security concerns in general and privacy concerns in specific [4]. The encryption has provided a lot of benefits for the user ensuring end-to-end secrecy and data confidentiality. The need to inspect the traffic originating or destined for the organization's network

has immensely increased for many security reasons. One of the reasons may be to simply validate parties involved in the communication [5].

Simple firewalls are generally not equipped with SSL inspection or off-loading which allows encrypted traffic to pass without any inspection [6]. This allows malicious traffic inside the network over covert channels that are not inspected by the firewall [7]. There is a dire need to detect legitimate and illegitimate traffic with minimal network overhead and overall system cost. This will allow any scale organization to better govern their organizational policies.

Virtual private network (VPN) service may be used to hide the real traffic in the network which may be otherwise not allowed or may be monitored [8]. A user using VPN service connects to a VPN server using normal Transport Layer Security (TLS) connection outside the network. Once connected, it requests the website or service from the server [9, 10]. The VPN server originates the request on behalf of the user to the server requested. The encrypted response is sent to the user on already established channel; as a result, the whole activity passes any filter on the network firewall.

Such techniques may be used by the users which aim to hide from or deceive the organization of their Internet activity [9]. This paper proposes a novel technique to detect VPN traffic inside a network. The proposed technique extracts the network traffic features and classifies the traffic to indicate if the traffic is legitimate or not. Key features are extracted from the network traffic and are compared against the already identified features of traffic found to be illegitimate or VPN traffic.

The system is also able to classify the traffic which is not following the pattern of normal traffic or normal user activity and flags that particular traffic stream to be invalid. We tested our system against five well-known freely available web-based VPN service providers; the proposed system was able to classify all of them correctly. More traffic-characterizing features may be added to identify more applications.

## 2. Related Work and Comparison

Multiple VPN services like TOR [11], Hotspot Shield, and other services have unique fingerprints, and not all the services can be distinguished using a similar criterion. Yamada et al. discussed a technique that uses statistical analysis on the encrypted traffic [12]. The scheme discussed, uses data size of network packets and performs timing analysis on the received packets to detect malicious traffic inside an encrypted channel. This technique is very useful for Web service providers to analyze the traffic coming to their servers and detect any malicious activity coming from outside the network.

A study on android-based applications which use VPN services [13] to show that these VPN services may use third-party trackers to track user behavior, and some may be used to bypass android sandbox environment. Once a malware or virus is delivered to the device inside the network, the whole network is vulnerable to attacks [14].

VPN clients inside the network act as a proxy, which connect to the respective VPN server. Once the connection

is established, the VPN service provider is able to change or eavesdrop on the information and network traffic as required [15, 16]. This attracts many third-party advertisement or tracking entities [17, 18]. Any malicious entity can read, save, and/or modify our request and the related information to and from the destined service.

VPN services can change the data as they are in control of incoming and outgoing traffic from network to device. VPN services are also able to perform TLS interception [19] by using their own certificates which is trusted locally by the system, for VPN service to work properly. This leads to a more potentially risky situation when the device connected contains sensitive data [13, 20]. One of the countermeasures to this issue is certificate pinning [13, 21]. So, detecting such VPN services inside your network can save you from huge losses in terms of the information lost.

Goh et al. [22] proposes a man-in-the-middle approach to detect VPN traffic in the network. The article puts forward a solution that uses *secret-sharing* scheme which involves a massive key management overhead using public key infrastructure (PKI) technique. The paper assumes that the traffic coming to the system is unencrypted and the data are available in plain form for the system to analyze and detect VPN traffic. This is achieved by using application layer proxy which generates the copy of unencrypted traffic against each connection which is then sent to the system for further analysis. This technique approximately doubles the network traffic and computational resources of existing system while increasing the memory requirements to decrypt and re-encrypt the web traffic.

Another solution that uses *Deep Packet Inspection* technique [23] uses multiple sensors throughout the network to get the unencrypted traffic from the end hosts and send it back to snort-based IDS [24] to detect unusual behavior in traffic. It increases the overall network traffic because a sensor is to be installed on each network machine to be able to detect any unusual activity. Another technique is to copy the entire connection traffic and use pre-shared secret to analyze any malicious traffic [25].

To identify applications being run inside the network, network analysis is used extensively. The work discussed by He et al. [26] uses basic yet one of the most effective and used techniques in network traffic analysis for traffic classification. Based on *five-tuple connection classification*, the technique uses connection characteristics like packet size, their interarrival time, and the direction and order of the packets to identify the network signature of any android application. The scheme provides basic understanding of traffic classification. However, network traffic generated by web-based VPN services will have no major difference or identifying characteristics, different to a standard HTTPS connection.

The use of unencrypted traffic to manage, analyze, and categorize encrypted traffic is an exciting concept, discussed by Niu et al. [27]. The schemes use labelled *DNS-based data set* to identify malicious command and control traffic and label the traffic as suspicious or normal. The concept provides a unique prospective to analyze the network traffic beyond five-tuple/ current connection technique discussed

previously [26]. Table 1 provides basic attributes of already discussed techniques. The techniques discussed pave the path of our proposed scheme.

Our proposed system analyzes *DNS records* to identify malicious or illegitimate VPN server names. Connection features are extracted using *five-tuple approach*. Five-tuple approach classifies each new connection by five attributes listed below:

- (i) Source IP
- (ii) Destination IP
- (iii) Protocol (TCP/UDP)
- (iv) Source port
- (v) Destination port

DNS-based traffic analysis and connection management were done using five-tuple techniques; our proposed system goes a step further to analyze *HTTPS handshake*. This is done to verify the server name used in the connection with the DNS activity which the user has generated by his network activity. Using this novel approach of managing a connection by using the activity preceding the current connection, we are able to detect and identify VPN traffic inside the network.

### 3. Forensic Analysis of VPN Services Client

To detect the network activity of VPN services, we carried out the forensic analysis of VPN services. For this purpose, we choose top five freely available web-based VPN services listed below:

- (i) TOR browser
- (ii) Hotspot shield free
- (iii) Browsec VPN
- (iv) ZenMate VPN
- (v) Hoxx VPN

For each of these VPN services, we analyzed the network traffic, generated by their clients, installed on a user PC. The initial analysis was performed using Wireshark [28] and NetworkMiner [29]. Detailed analysis of each VPN service is discussed below.

**3.1. Hotspot Shield.** Hotspot shield [30] developed by AnchorFree is one of the leading free VPN services used. We tested its two versions:

- (i) Client application for windows desktop
- (ii) Firefox add-on

**3.1.1. Client Application for Windows Desktop.** In client version of the abovementioned VPN service, it was observed that once enabled, the service uses standard port 443 for HTTPS connections but generally connects to only one server. All the traffic may it be multisite traffic uses the same active connection. Figure 1 shows the connection details for current user activity against Hotspot Shield. Hotspot Shield

uses fake well-known server name in SSL certificate to bypass the traffic from server name-based filters over the network, if any, as shown in Figure 2 below.

It can be seen that the used server name is *twitter.com*. It does not generate any DNS entry for such server name. The NetworkMiner tool shows us the connection details in Figure 3. We can see that eight unique connections were made; in this case, it generally means eight unique web pages were open. Requests of all these web pages were managed by the server whose IP is *136.0.99.219*. Certificate details can also be seen against this server IP which were received. Total 20,708 packets were sent in this activity, and 116,84 packets were received.

Figure 4 shows that no DNS activity for such host name was found during the communication. We can see all the DNS generated by the user while using Hotspot Shield client.

**3.1.2. Firefox Add-On.** Hotspot Shield in add-on uses standard https port along with standard DNS queries. The only way to detect Hotspot Shield inside the network is to identify the domain names used by Hotspot Shield. Shown below in Figure 5 is the network traffic generated by Hotspot Shield captured using Wireshark.

It can be seen in Figure 6 that the domain name is *ext-mi-ex-nl-ams-pr-p-1.northghost.com* for which the connection is established.

We observed that Hotspot Shield domain name consists of two main parts:

- (i) Server identifier
- (ii) Domain name

This can also be seen in certificate details in Figure 7, analyzed by NetworkMiner tool:

It is clearly observed that the domain name is *\*.northghost.com* and the other part is some server identifier as it may change once you reinstate the connection. It can be seen that the connections for Hotspot Shield were established against only one server with IP address *216.162.47.67*. Total connections established were 35, and a total of 207,08 packets were sent in this activity, and 11,684 packets were received.

The add-on also generates standard DNS activity as shown in Figure 8.

Changing the VPN locations from add-on's option has no effect on the server being connected by the client as the server identifier in the same activity does not change.

**3.2. ZenMate.** ZenMate [31] developed by ZenGuard is also very popular free VPN service used. We analyzed the chrome-based add-on of ZenMate. It uses standard https port along with standard DNS queries. The only way to detect ZenMate inside the network is to identify the domain names used by ZenMate VPN. Shown below in Figure 9 is the network traffic generated by ZenMate VPN captured using Wireshark.

It can be seen in Figure 10 that the domain name is *63.ayala-maroon.ga* for which the connection is established.



TABLE 1: Attributes of related techniques.

Research techniques	Strengths	Limitations
NIDS-based technique [22]	(1) Complete architecture to handle encrypted traffic-based intrusion detection (2) Protection against remote access and evasion techniques	(1) Multiple devices to be added in the network (2) Increased bandwidth inside the network due to traffic duplication
DNS-based technique [27]	(1) Introduces the concept of DNS scoring and analysis. Helpful in detecting malicious CNC based on DNS	(1) All CNC may not use only DNS based implementation
Connection-based technique [26]	(1) Five-tuple-based connection management. Helpful in identifying different protocol and application behavior	(1) Traffic generated by HTTPS based VPN will generally look like standard HTTPS streams

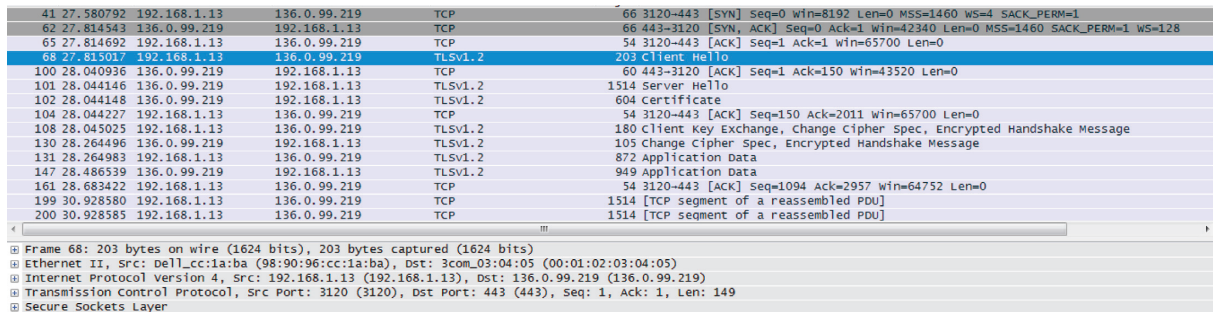


FIGURE 1: Wireshark: Hotspot Shield client.

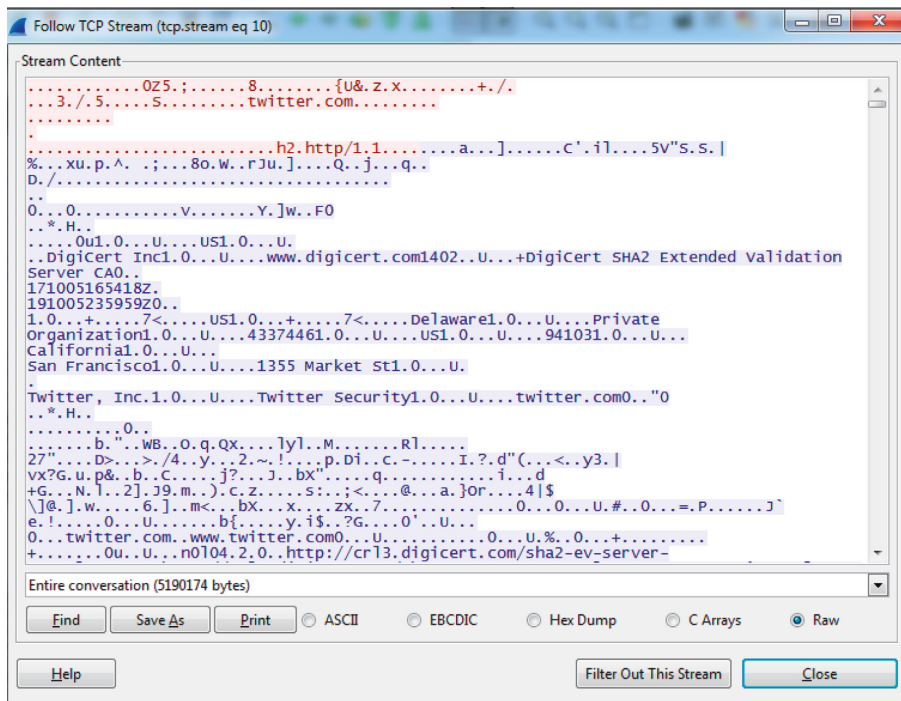


FIGURE 2: Wireshark: Hotspot Shield TCP stream.

Like Hotspot Shield, ZenMate’s domain name also consists of two main parts:

- (i) Server identifier
- (ii) Domain name

This can also be seen in certificate details in Figure 11, analyzed by NetworkMiner tool:

It is clearly observed that the domain name is *\*.ayala-maroon.ga*, and the number part is some server identifier. ZenMate is unique from other VPN services as it constantly

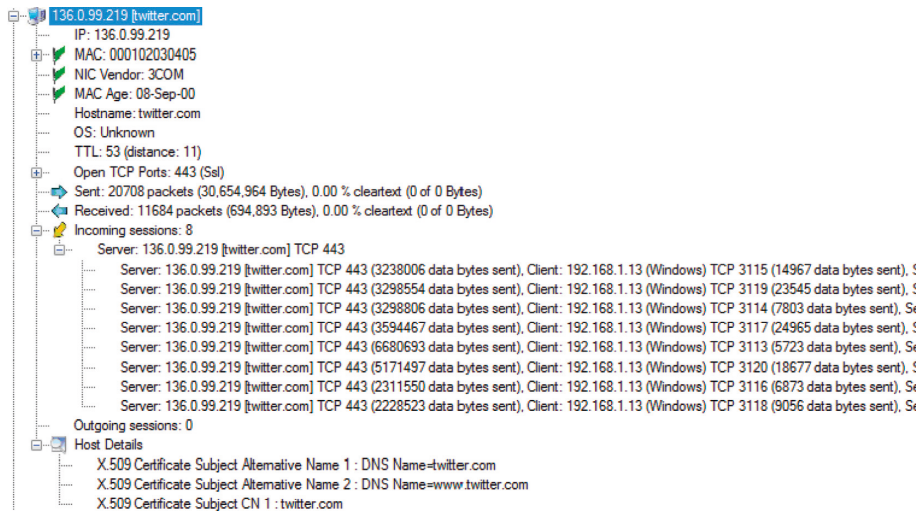


FIGURE 3: NetworkMiner: Hotspot Shield connection details.

Frame nr.	Timestamp	Client	Client Port	Server	Server Port	IP TTL	DNS TTL (time)	Transaction ID	Type	DNS Query	DNS Answer
3	2018-09-26 10:14:28 UTC	192...	49283	192...	53	61	00:01:00	0xD25A	0x0005 (CNAME)	geo.hotspotshield.com	us.hotspotshield.com
3	2018-09-26 10:14:28 UTC	192...	49283	192...	53	61	00:01:00	0xD25A	0x0001 (Host Address)	us.hotspotshield.com	74.115.0.53
32524	2018-09-26 10:15:31 UTC	192...	51671	192...	53	61	00:01:00	0x0337	0x0001 (Host Address)	djh90c9110en8.cloudfront.net	143.204.208.138
32524	2018-09-26 10:15:31 UTC	192...	51671	192...	53	61	00:01:00	0x0337	0x0001 (Host Address)	djh90c9110en8.cloudfront.net	143.204.208.6
32524	2018-09-26 10:15:31 UTC	192...	51671	192...	53	61	00:01:00	0x0337	0x0001 (Host Address)	djh90c9110en8.cloudfront.net	143.204.208.122
32524	2018-09-26 10:15:31 UTC	192...	51671	192...	53	61	00:01:00	0x0337	0x0001 (Host Address)	djh90c9110en8.cloudfront.net	143.204.208.156

FIGURE 4: NetworkMiner: no DNS information found for 136.0.99.219.

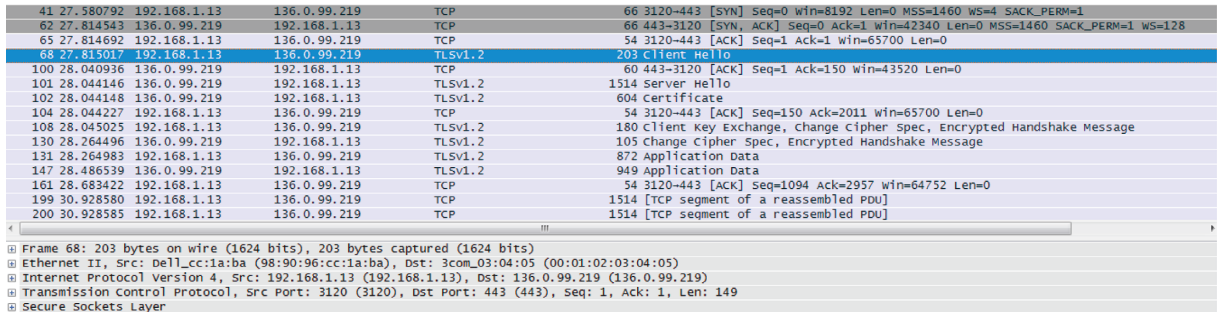


FIGURE 5: Wireshark: Hotspot Shield add-on.

changes the servers being connected by a user. So, any suspicious or long activity with one server cannot be identified by automated tools. As seen in Figure 11, multiple host names against the same domains are listed in SSL certificate provided by the VPN server. These servers/hosts may be used randomly to request multiple resources over the Internet. It is clearly shown in the figure that the number of connections against this server is only five, which is less than other VPN servers' connection discussed in the paper.

Another unique feature that ZenMate offers is that it changes the domain name as well once the location of the VPN server is changed from the settings of add-on. As shown in Figure 12, the server name is changed to *34.lutz-obrien-olive.ga* once the user has changed the location.

ZenMate changes domain names against region selected by the user, but for the same region, the server identifier of domain name may change but domain remains the same. If a

user is constantly changing the locations, after some time when all locations available are exhausted, the domains for each location could be identified. As shown is Figure 13, multiple domains for ZenMate service used by this user are as follows:

- (i) lutz-obrien-olive.ga
- (ii) ayala-maroon.ga
- (iii) hall-silver.ga
- (iv) young-purple.ga

This information can now be used to prepare a filter to identify ZenMate VPN inside the network. One can also notice that the last part of domain is always a *color* and ends with *.ga*. So, if we received DNS request or response and the domain name ends with *.ga* with “-” (*dash*) in the query, it could be separated on “-.” Once separated, if the last string

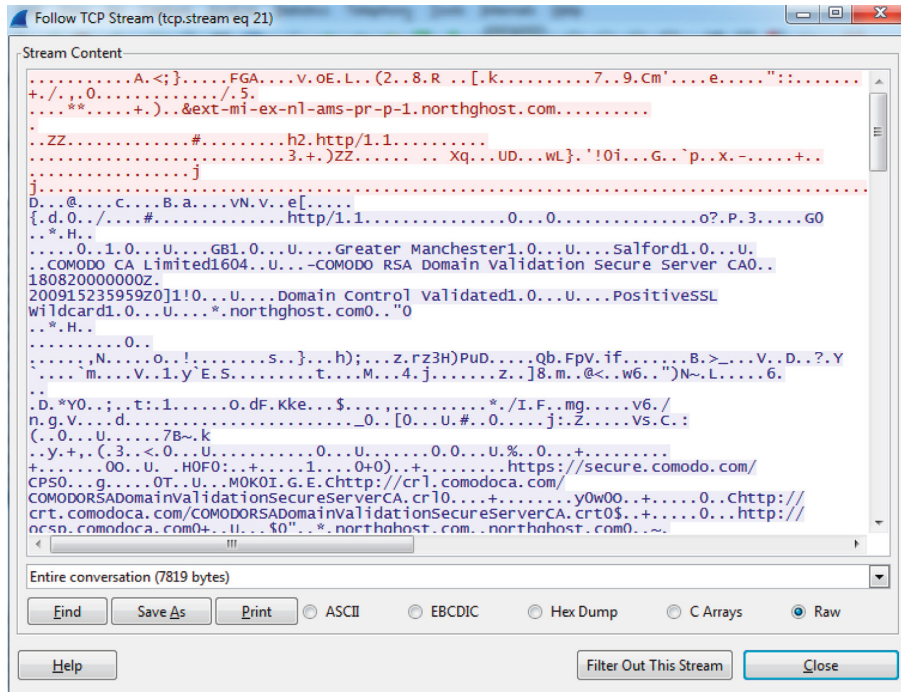


FIGURE 6: Wireshark: Hotspot Shield add-on TCP stream.

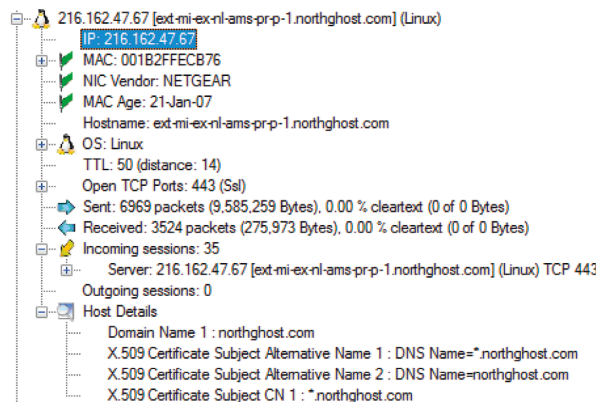


FIGURE 7: NetworkMiner: Hotspot Shield add-on connection details.

2983	2019-05-13 07:13:26 UTC	192...	60552	192...	53	64	00:04:17	0x35...	0x0005 (CNAME)	fonts.gstatic.com	gstaticadsll.google.com
2983	2019-05-13 07:13:26 UTC	192...	60552	192...	53	64	00:04:17	0x35...	0x0001 (Host Address)	gstaticadsll.google.com	172.217.19.3
3201	2019-05-13 07:13:26 UTC	192...	62215	192...	53	64	02:49:29	0x3A...	0x0005 (CNAME)	www.google-analytics.com	www.google-analytics.l.google.com
3201	2019-05-13 07:13:26 UTC	192...	62215	192...	53	64	00:04:39	0x3A...	0x0001 (Host Address)	www.google-analytics.l.google.com	172.217.19.14
5200	2019-05-13 07:13:30 UTC	192...	61388	192...	53	64	00:05:00	0xE9...	0x0001 (Host Address)	ext-mi-ex-nl-ams-prp-1.northghost.com	216.162.47.67
14297	2019-05-13 07:13:43 UTC	192...	59456	192...	53	64	00:24:24	0x25...	0x0005 (CNAME)	edge-chat.facebook.com	star.c10.facebook.com
14297	2019-05-13 07:13:43 UTC	192...	59456	192...	53	64	00:00:07	0x25...	0x0001 (Host Address)	star.c10.facebook.com	157.240.24.20
14927	2019-05-13 07:13:44 UTC	192...	51326	192...	53	64	00:00:47	0xA2...	0x0001 (Host Address)	star-mini.c10.facebook.com	157.240.24.35
14927	2019-05-13 07:13:44 UTC	192...	51326	192...	53	64	00:04:26	0xA2...	0x0005 (CNAME)	www.facebook.com	star-mini.c10.facebook.com

FIGURE 8: NetworkMiner: DNS information for 136.0.99.219.

contains any well-known color name, we can classify it as ZenMate DNS server. As shown in Figure 14, the domain name analysis was done by NetworkMiner, we can see the same pattern discussed above.

3.3. TOR Browser. TOR Browser [11] is used generally by users to hide their Internet activity and to access resources

on dark web. TOR browser uses a concept of onion routing to hide user’s activity. We installed TOR browser to analyze the network traffic generated by the browser. It uses a nonstandard port for communication over Internet. It uses HTTPS over 9001 TCP Port initially for circuit connection. After the circuit connection is established, TOR may use 443 for normal Internet or any other port as configured. TOR

508	44.562205	192.168.100.3	193.176.86.50	TCP	66	3921-443 [SYN] Seq=0 win=17520 Len=0 MSS=1460 WS=256 SACK_PERM=1
509	44.723596	193.176.86.50	192.168.100.3	TCP	66	443-3921 [SYN, ACK] Seq=0 Ack=1 win=29200 Len=0 MSS=1412 SACK_PERM=1 WS=512
510	44.723755	192.168.100.3	193.176.86.50	TCP	54	3921-443 [ACK] Seq=1 Ack=1 win=17408 Len=0
511	44.724659	192.168.100.3	193.176.86.50	TLSv1.2	571	Client Hello
513	44.884046	193.176.86.50	192.168.100.3	TCP	54	443-3921 [ACK] Seq=1 Ack=518 win=29696 Len=0
514	44.896032	193.176.86.50	192.168.100.3	TLSv1.2	1466	server Hello
515	44.897054	193.176.86.50	192.168.100.3	TCP	1466	[TCP segment of a reassembled PDU]
516	44.897060	193.176.86.50	192.168.100.3	TLSv1.2	1466	certificate
517	44.897079	193.176.86.50	192.168.100.3	TLSv1.2	160	server Key Exchange
518	44.897184	192.168.100.3	193.176.86.50	TCP	54	3921-443 [ACK] Seq=518 Ack=4343 win=17408 Len=0
519	44.915463	192.168.100.3	193.176.86.50	TLSv1.2	180	client Key Exchange, change cipher Spec, Hello Request, Hello Request
523	45.074344	193.176.86.50	192.168.100.3	TLSv1.2	296	New Session Ticket, change cipher Spec, Encrypted Handshake Message
524	45.078336	192.168.100.3	193.176.86.50	TLSv1.2	308	Application Data
532	45.246899	193.176.86.50	192.168.100.3	TLSv1.2	102	Application Data
533	45.248158	192.168.100.3	193.176.86.50	TLSv1.2	660	Application Data

FIGURE 9: Wireshark: ZenMate add-on.

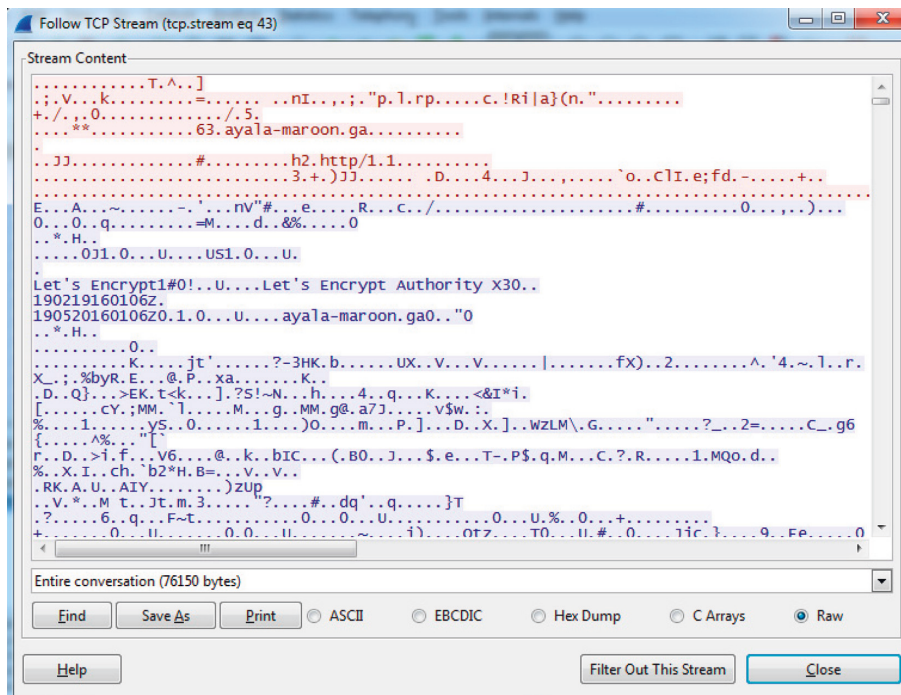


FIGURE 10: Wireshark: ZenMate add-on TCP stream.

will generally not generate any DNS traffic. A normal TOR stream viewed in Wireshark is shown in Figure 15.

Opening of each website may create new connection to server and server name along with their IP addresses which are communicated to TOR browser during circuit establishment process and are encrypted. Figure 16 shows a TOR-based TCP stream analyzed in Wireshark.

Connection details of a TOR connection analyzed by NetworkMiner are shown in Figure 17. It shows that, against server IP 5.9.42.230, a total of 639 packets were sent and 586 packets were received by the user.

Complete activity of the user for the session being discussed is also shown in Figure 18. It is interesting to mention here that no DNS activity was found for TOR browser.

3.4. *Browsec VPN*. Browsec VPN [32] is another freely available VPN. We used it as Firefox add-on. It uses standard

HTTPS port along with standard DNS queries. The only way to detect Browsec VPN inside the network is to identify the domain names used by it. Shown below in Figure 19 is the network traffic generated by Browsec VPN captured using Wireshark.

It can be seen in Figure 20 that the domain name is *nl30.tcdn.me* for which the connection was established. Like other VPN services, the domain name of Browsec VPN can also be further divided for better analysis. It consists of three main parts; it can also be seen in certificate details in Figure 21, analyzed by NetworkMiner tool:

- (i) Country code
- (ii) Server identifier
- (iii) Domain name

It is clearly observed that the domain name is *\*.tcdn.me* and the other part consists of some server identifier and location identifier. In Figure 21, the location identifier is *nl*,

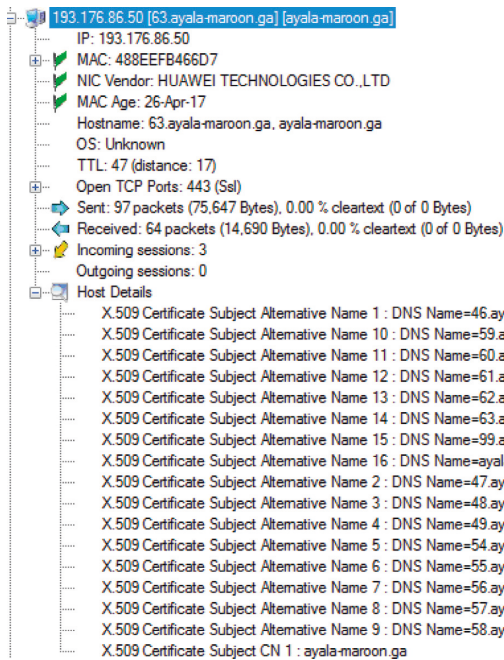


FIGURE 11: NetworkMiner: ZenMate VPN add-on connection details.

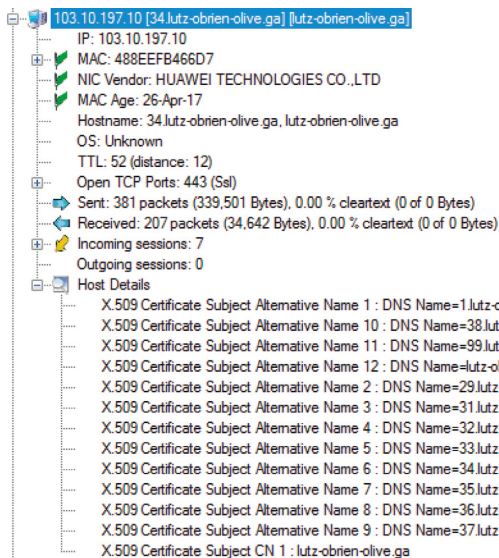


FIGURE 12: NetworkMiner: ZenMate VPN add-on connection details—changed location.

which means Netherlands, and in Figure 22, we can see the country is United Kingdom.

Like ZenMate VPN, Browsec VPN also changes its DNS information when changing the location, but unlike ZenMate, the domain name is not changed rather only the server qualifier is changed. Figure 23 shows the DNS traffic generated by user's activity.

**3.5. Hoxx VPN.** Hoxx VPN [33] is another freely available VPN. We used it as Firefox add-on. It uses standard HTTPS port along with standard DNS queries. We can detect Hoxx

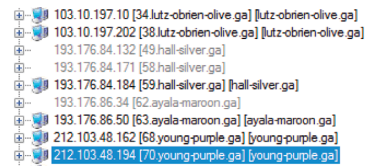


FIGURE 13: NetworkMiner: ZenMate VPN add-on connection details—all locations.

VPN inside the network by identifying the domain names used by the VPN service. Shown below in Figure 24 is the network traffic generated by Hoxx VPN captured using Wireshark.

It can be seen in Figure 25 that the domain name is *dyn-146-185-141-219-5871-b377a.klafive.com* for which the connection is established. Like other VPN services, the domain name of Hoxx VPN server can also be further divided for better analysis. It consists of two main parts:

- (i) Server identifier
- (ii) Domain name

This division can also be seen in certificate details in Figure 26, analyzed by NetworkMiner tool. It is clearly observed that domain name is *\*.klafive.com* and the other part consists of some server identifier. Figure 27 shows the DNS traffic generated by user's activity.

## 4. Proposed System

The proposed system distinguishes the normal flow of an Internet activity or session from an abnormal one. Normally, when a user wants to connect to a website a DNS request is made to translate the web name to IP address [34]. After successful name resolution, against the IP, a TCP (Transmission Control Protocol) session is initiated and required security associations are established. This behavior may be used to monitor and analyze different features of network traffic. [35–37].

The proposed system classifies any incoming data into multiple categories depending on the current state of connection; in addition to that, Internet activity preceding the connection is also monitored to identify the traffic as VPN or simple Internet traffic. The process of detecting any illegitimate traffic is further classified into two main processes:

- (i) Feature extraction
- (ii) Traffic classification

**4.1. Feature Extraction.** To classify traffic as normal or VPN, we have to extract different traits of the network traffic. Now, most of these traits can be found in current traffic stream while some of them are collected before the actual stream starts. Figure 28 shows the basic flow of network traffic feature extraction module of the system. The analyzer extracts the following information to be used for traffic categorization.

5438	2019-04-01 19:32:38 UTC	192....	53872	192....	53	64	00:01:08	0x5F28	0x0001 (Host Address)	34.lutz-obrien-olive.ga	103.10.197.10
634	2019-04-01 19:30:33 UTC	192....	62279	192....	53	64	00:00:19	0xCE61C	0x0001 (Host Address)	38.lutz-obrien-olive.ga	103.10.197.202
4705	2019-04-01 19:31:16 UTC	192....	63787	192....	53	64	00:05:00	0x1FCC	0x0001 (Host Address)	49.hall-silver.ga	193.176.84.132
4592	2019-04-01 19:31:00 UTC	192....	60082	192....	53	64	00:05:00	0x743A	0x0001 (Host Address)	58.hall-silver.ga	193.176.84.171
4980	2019-04-01 19:31:53 UTC	192....	55896	192....	53	64	00:05:00	0xCE24	0x0001 (Host Address)	59.hall-silver.ga	193.176.84.184
356	2019-04-01 19:30:06 UTC	192....	61995	192....	53	64	00:04:39	0xE773	0x0001 (Host Address)	62.ayala-maroon.ga	193.176.86.34
507	2019-04-01 19:30:30 UTC	192....	49346	192....	53	64	00:04:21	0xF496	0x0001 (Host Address)	63.ayala-maroon.ga	193.176.86.50
5270	2019-04-01 19:32:18 UTC	192....	60309	192....	53	64	00:03:43	0xC40C	0x0001 (Host Address)	68.young-purple.ga	212.103.48.162
5153	2019-04-01 19:32:07 UTC	192....	65171	192....	53	64	00:00:09	0x713C	0x0001 (Host Address)	70.young-purple.ga	212.103.48.194

FIGURE 14: NetworkMiner: DNS information for ZenMate servers.

No.	Time	Source	Destination	Protocol	Length	Info
850	9.961776	172.16.0.6	5.9.42.230	TCP	74	41744 → 9001 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
851	10.111696	5.9.42.230	172.16.0.6	TCP	82	9001 → 41744 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0
852	10.111732	172.16.0.6	5.9.42.230	TCP	66	41744 → 9001 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSv=
853	10.111875	172.16.0.6	5.9.42.230	TLSv1.2	260	Client Hello
860	10.260446	5.9.42.230	172.16.0.6	TCP	74	9001 → 41744 [ACK] Seq=1 Ack=195 Win=30080 Len=0
861	10.264325	5.9.42.230	172.16.0.6	TLSv1.2	1079	Server Hello, Certificate, Server Key Exchange, Se
862	10.264349	172.16.0.6	5.9.42.230	TCP	66	41744 → 9001 [ACK] Seq=195 Ack=1006 Win=32128 Len=0
863	10.265052	172.16.0.6	5.9.42.230	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encryptec
869	10.414827	5.9.42.230	172.16.0.6	TLSv1.2	125	Change Cipher Spec, Encrypted Handshake Message [E
870	10.415039	172.16.0.6	5.9.42.230	TLSv1.2	106	Application Data
871	10.567576	5.9.42.230	172.16.0.6	TLSv1.2	679	[TCP Previous segment not captured] , Ignored Unknr
872	10.567604	172.16.0.6	5.9.42.230	TCP	78	[TCP Window Update] 41744 → 9001 [ACK] Seq=361 Ack
873	10.568823	5.9.42.230	172.16.0.6	TCP	1522	[TCP Out-Of-Order] 9001 → 41744 [ACK] Seq=1057 Ack

> Frame 851: 82 bytes on wire (656 bits), 82 bytes captured (656 bits)  
 > Ethernet II, Src: Netgear\_fc:cb:76 (00:1b:2f:fe:cb:76), Dst: Tp-LinkT\_1c:2a:63 (60:e3:27:1c:2a:63)  
 > Internet Protocol Version 4, Src: 5.9.42.230, Dst: 172.16.0.6  
 > Transmission Control Protocol, Src Port: 9001, Dst Port: 41744, Seq: 0, Ack: 1, Len: 0

FIGURE 15: Wireshark: TOR traffic.

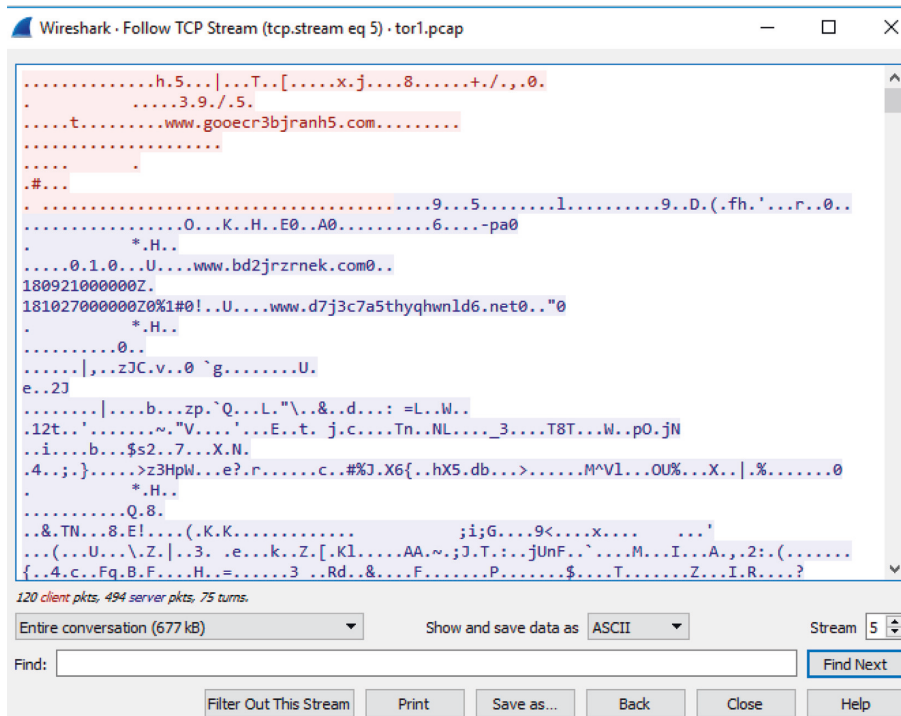


FIGURE 16: Wireshark: TOR browser TCP stream.

4.1.1. *Basic Feature Extraction.* Server IP of the server and user is extracted at the first step. This information is extracted from IPv4 Protocol fields, source IP and destination IP [38]. Depending upon the transport layer protocol, the source port and destination ports are also extracted [39].

4.1.2. *Domain Name Server Analysis.* Unencrypted traffic information is as important in traffic characterization and behavior analysis of users as the encrypted traffic. For any web request, generated by a user, a DNS request is initiated by the user’s browser to request the IP information of the

5.9.42.230 [www.goocrc3bjranh5.com] [www.d73c7a5thyqhwld6.net] [www.b6enz3fiouyfmptwa.com] (Linux)  
 IP: 5.9.42.230  
 MAC: 001B2FFECB76  
 NIC Vendor: NETGEAR  
 MAC Age: 1/21/2007  
 Hostname: www.goocrc3bjranh5.com, www.d73c7a5thyqhwld6.net, www.b6enz3fiouyfmptwa.com  
 OS: Linux  
 TTL: 51 (distance: 13)  
 Open TCP Ports: 9001 (Ssl)  
 Sent: 639 packets (779,519 Bytes), 0.00% cleartext (0 of 0 Bytes)  
 Received: 586 packets (207,195 Bytes), 0.00% cleartext (0 of 0 Bytes)  
 Incoming sessions: 2  
 Outgoing sessions: 0  
 Host Details  
 X.509 Certificate Subject CN 1 : www.d73c7a5thyqhwld6.net

FIGURE 17: NetworkMiner: TOR browser connection details.

5.9.42.230 [www.goocrc3bjranh5.com] [www.d73c7a5thyqhwld6.net] [www.b6enz3fiouyfmptwa.com] (Linux)  
 95.130.12.119 [www.e6c3r3ntbd4fsfrenypdf7o.com] [www.i72ed2gr6vpyztcx.net] (Linux)  
 109.236.90.209 [www.77bahgmj.com] [www.2u7hg4dwgcg2vkbxfk5.net]  
 164.132.77.175  
 172.16.0.1  
 172.16.0.6 (Linux)  
 185.13.39.197 [www.x5o244h62yix23cdfvcuhso.com] [www.hbafmp5y3bdww.net] (Linux)  
 195.228.75.149 [www.lehip.com] [www.uocook7z3eae.net] (Linux)

FIGURE 18: NetworkMiner: TOR browser user activity details.

6195	194.862092	192.168.100.3	198.16.66.139	TCP	66	4069 → 443 [SYN] Seq=0 Win=17520 Len=0 MSS=1460 WS=256 SACK_PERM=1
6196	195.012806	198.16.66.139	192.168.100.3	TCP	66	443 → 4069 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1412 SACK_PERM=1
6197	195.012912	192.168.100.3	198.16.66.139	TCP	54	4069 → 443 [ACK] Seq=1 Ack=1 Win=17408 Len=0
6198	195.013664	192.168.100.3	198.16.66.139	TLV1.2	571	Client Hello
6200	195.208806	198.16.66.139	192.168.100.3	TCP	54	443 → 4069 [ACK] Seq=1 Ack=518 Win=30336 Len=0
6201	195.759557	198.16.66.139	192.168.100.3	TLV1.2	1466	Server Hello
6202	195.760354	198.16.66.139	192.168.100.3	TLV1.2	1466	Certificate [TCP segment of a reassembled PDU]
6203	195.760357	198.16.66.139	192.168.100.3	TLV1.2	270	Server Key Exchange, Server Hello Done
6204	195.760426	192.168.100.3	198.16.66.139	TCP	54	4069 → 443 [ACK] Seq=518 Ack=3041 Win=17408 Len=0
6205	195.785770	192.168.100.3	198.16.66.139	TLV1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
6206	195.935194	198.16.66.139	192.168.100.3	TCP	54	443 → 4069 [ACK] Seq=3041 Ack=644 Win=30336 Len=0
6207	196.250670	198.16.66.139	192.168.100.3	TLV1.2	296	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
6208	196.251400	192.168.100.3	198.16.66.139	TLV1.2	308	Application Data

> Frame 6195: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)  
 > Ethernet II, Src: IntelCor\_77:5c:13 (34:e6:ad:77:5c:13), Dst: HuaweiTe\_b4:66:d7 (48:8e:ef:b4:66:d7)  
 > Internet Protocol Version 4, Src: 192.168.100.3, Dst: 198.16.66.139  
 > Transmission Control Protocol, Src Port: 4069, Dst Port: 443, Seq: 0, Len: 0

FIGURE 19: Wireshark: Browsec VPN add-on.

```

.....A...i...H2:r...b...{!.~`& .}7.....
.LQ].w...#Y...@s'.....+./.,0...../.5.
.....:.....nl30.tcdn.me.....
.....#.....h2.http/1.1.....
.....3.+.).....h}.j..^...b."...z+...".(.....+..
ZZ.....E...A..f...=..
..?08w...C.kl..3b.....a.Y..0.....#.....
6..
2.
/...0...0...../..Z.p...Dzo.0
..*..H..
.....0L1.0 ..U...BE1.0...U.
..GlobalSign nv-sal*0 ..U...AlphaSSL CA - SHA256 - G20..
181012080114Z.
    
```

14 client pkts, 274 server pkts, 17 turns.  
 Entire conversation (356 kB) Show and save data as ASCII Stream 184  
 Find: Find Next

FIGURE 20: Wireshark: Browsec VPN TCP stream.

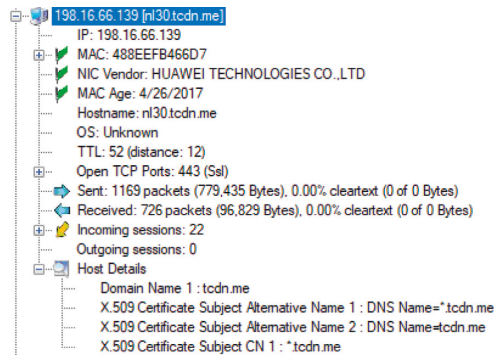


FIGURE 21: NetworkMiner: Browsec VPN connection details—NL.

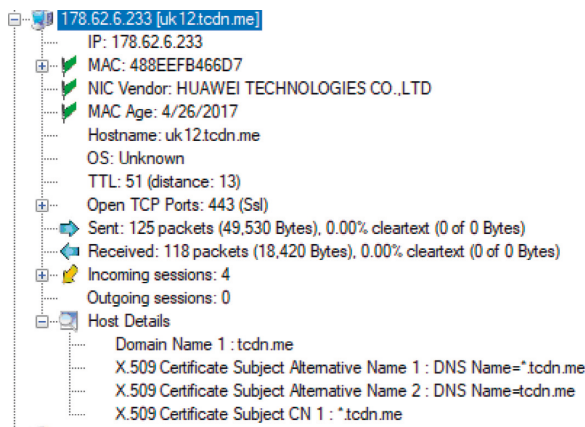


FIGURE 22: NetworkMiner: Browsec VPN connection details—UK.

Frame nr.	Timestamp	Client	Client Port	Ser...	Server Port	IP TTL	DNS TTL (time)	Transaction ID	Type	DNS Query	DNS Answer
6194	2019-04-01 19:33:01 UTC	192....	50050	192....	53	64	17:24:48	0x3D78	0x0001 (Host Address)	nl30.tcdn.me	198.16.66.139
7425	2019-04-01 19:33:15 UTC	192....	59982	192....	53	64	17:20:27	0xDA3C	0x0001 (Host Address)	nl10.tcdn.me	198.16.66.123
8497	2019-04-01 19:33:39 UTC	192....	53446	192....	53	64	11:18:17	0x7234	0x0001 (Host Address)	sg17.tcdn.me	178.128.57.177
8770	2019-04-01 19:34:00 UTC	192....	61604	192....	53	64	12:19:49	0x7957	0x0001 (Host Address)	sg25.tcdn.me	178.128.117.77
8980	2019-04-01 19:34:02 UTC	192....	64325	192....	53	64	17:14:48	0x173F	0x0001 (Host Address)	uk1.tcdn.me	178.62.34.82
9282	2019-04-01 19:34:23 UTC	192....	56276	192....	53	64	17:19:22	0x19B4	0x0001 (Host Address)	uk9.tcdn.me	46.101.16.229
9762	2019-04-01 19:34:44 UTC	192....	49190	192....	53	64	17:24:24	0x86B9	0x0001 (Host Address)	uk12.tcdn.me	178.62.6.233

FIGURE 23: NetworkMiner: DNS information for Browsec VPN.

No.	Time	Source	Destination	Protocol	Length	Info
252	28.811041	192.168.1.2	149.28.168.15	TCP	66	9687 → 443 [SYN] Seq=0 Win=17520 Len=0 MSS=1460 WS=256 SACK_PERM=1
260	29.010811	149.28.168.15	192.168.1.2	TCP	66	443 → 9687 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1
261	29.010891	192.168.1.2	149.28.168.15	TCP	54	9687 → 443 [ACK] Seq=1 Ack=1 Win=17408 Len=0
262	29.013443	192.168.1.2	149.28.168.15	TLsv1.2	571	Client Hello
266	29.214909	149.28.168.15	192.168.1.2	TCP	54	443 → 9687 [ACK] Seq=1 Ack=518 Win=30336 Len=0
267	29.218401	149.28.168.15	192.168.1.2	TLsv1.2	1514	Server Hello
268	29.219246	149.28.168.15	192.168.1.2	TLsv1.2	1514	Certificate [TCP segment of a reassembled PDU]
269	29.219250	149.28.168.15	192.168.1.2	TLsv1.2	137	Server Key Exchange, Server Hello Done
270	29.219319	192.168.1.2	149.28.168.15	TCP	54	9687 → 443 [ACK] Seq=518 Ack=3004 Win=17408 Len=0
271	29.232412	192.168.1.2	149.28.168.15	TLsv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
284	29.428043	149.28.168.15	192.168.1.2	TLsv1.2	105	Change Cipher Spec, Encrypted Handshake Message
287	29.439411	192.168.1.2	149.28.168.15	TLsv1.2	354	Application Data
289	29.645919	149.28.168.15	192.168.1.2	TCP	1514	443 → 9687 [ACK] Seq=3055 Ack=944 Win=31360 Len=1460 [TCP segment of a

> Frame 252: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)

> Ethernet II, Src: IntelCor\_77:5c:13 (34:e6:ad:77:5c:13), Dst: Netgear\_fe:cb:76 (00:1b:2f:fe:cb:76)

> Internet Protocol Version 4, Src: 192.168.1.2, Dst: 149.28.168.15

> Transmission Control Protocol, Src Port: 9687, Dst Port: 443, Seq: 0, Len: 0

FIGURE 24: Wireshark: Hoxx VPN add-on.



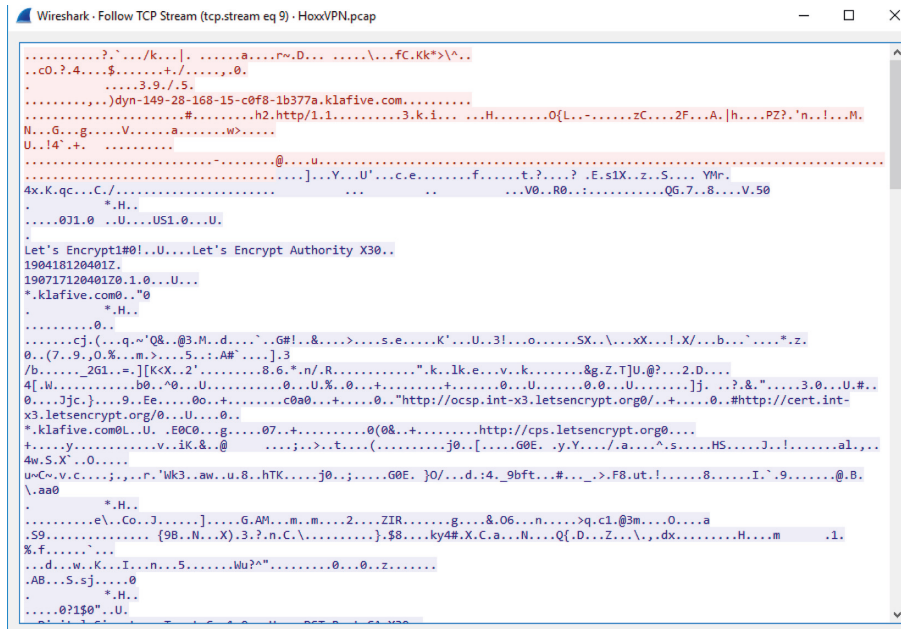


FIGURE 25: Wireshark: Hoxx VPN TCP stream.

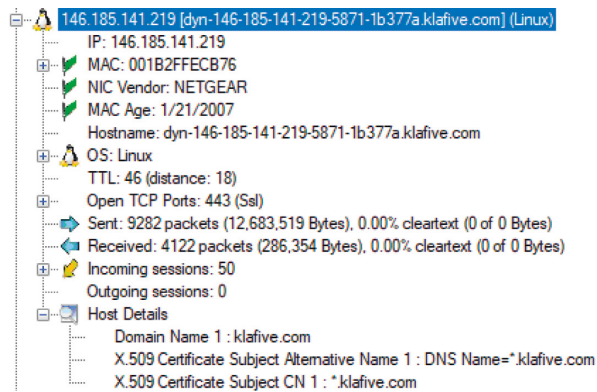


FIGURE 26: NetworkMiner: Hoxx VPN connection details.

250	2019-05-13 07:37:01 UTC	192...	51103	192...	53	64	1.00:00:00	0xCFD8	0x0001 (Host ...	dyn-149-28-168-15-c0f8-1b377a.k1a5ive.com	149.28.168.15
251	2019-05-13 07:37:01 UTC	192...	51103	192...	53	64	1.00:00:00	0xCFD8	0x0001 (Host ...	dyn-149-28-168-15-c0f8-1b377a.k1a5ive.com	149.28.168.15
254	2019-05-13 07:37:01 UTC	192...	62867	192...	53	64	1.00:00:00	0xC6BD	0x0001 (Host ...	dyn-149-28-168-15-c0f8-1b377a.k1a5ive.com	149.28.168.15
263	2019-05-13 07:37:01 UTC	192...	61758	192...	53	64	00:00:00	0x621C	0x0000	dyn-149-28-168-15-c0f8-1b377a.k1a5ive.com	No error condition (flags 0x8180)
264	2019-05-13 07:37:01 UTC	192...	61758	192...	53	64	00:00:00	0x621C	0x0000	dyn-149-28-168-15-c0f8-1b377a.k1a5ive.com	No error condition (flags 0x8180)
895	2019-05-13 07:37:14 UTC	192...	64210	192...	53	64	00:00:00	0x6FAE	0x0000	dyn-149-28-168-15-c0f8-1b377a.k1a5ive.com	No error condition (flags 0x8180)
18562	2019-05-13 07:37:48 UTC	192...	54745	192...	53	64	1.00:00:00	0xCE67	0x0001 (Host ...	dyn-146-185-141-219-5871-1b377a.k1a5ive.com	146.185.141.219
18563	2019-05-13 07:37:48 UTC	192...	54745	192...	53	64	1.00:00:00	0xCE67	0x0001 (Host ...	dyn-146-185-141-219-5871-1b377a.k1a5ive.com	146.185.141.219
18572	2019-05-13 07:37:48 UTC	192...	50164	192...	53	64	23:59:59	0x5EB6	0x0001 (Host ...	dyn-146-185-141-219-5871-1b377a.k1a5ive.com	146.185.141.219
19170	2019-05-13 07:37:49 UTC	192...	55650	192...	53	64	00:00:00	0xCC57	0x0000	dyn-146-185-141-219-5871-1b377a.k1a5ive.com	No error condition (flags 0x8180)
19173	2019-05-13 07:37:49 UTC	192...	55650	192...	53	64	00:00:00	0xCC57	0x0000	dyn-146-185-141-219-5871-1b377a.k1a5ive.com	No error condition (flags 0x8180)

FIGURE 27: NetworkMiner: DNS information for Hoxx VPN.

server name. A response is sent to the user from DNS server containing IP information of the server [34]. This information is stored by our system to verify the DNS server name vs. HTTPS certificate's server name to see for any inconsistencies.

4.1.3. *HTTPS Protocol Detection.* Incoming traffic is then passed to HTTPS detection module. The system looks for HTTPS other than port 443. This is done by looking for HTTPS headers on streams which are TCP-based connections but the server port number is other than 443. A lot of

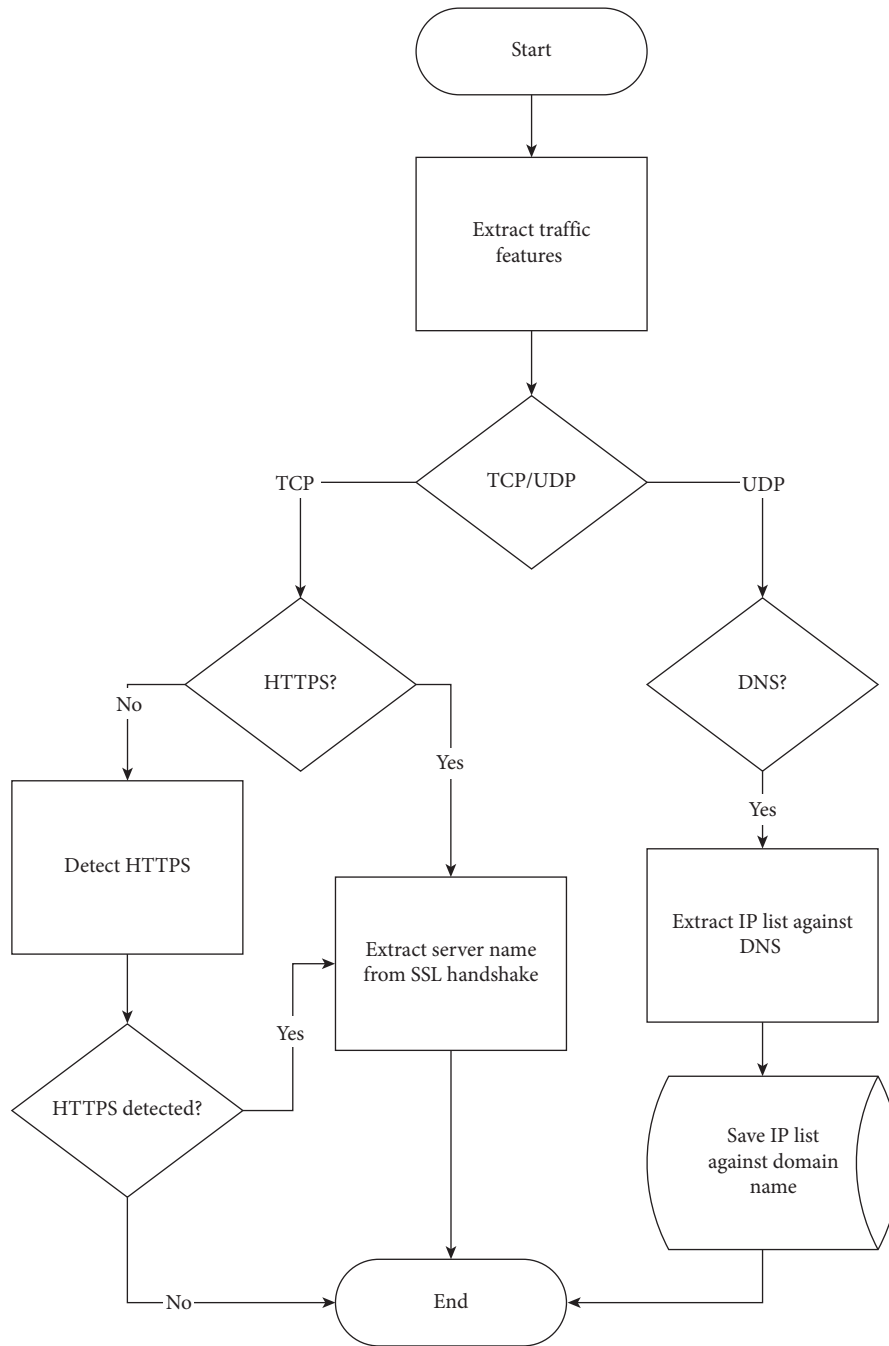


FIGURE 28: Feature extraction.

applications and services use the technique to change the server port. This allows them to pass through network firewall and is not labelled as encrypted payload.

4.1.4. *SSL Analysis.* The proposed system decodes SSL certificates [40] once HTTPS is detected. There are 4 basic types of messages in SSL:

- (i) Handshake
- (ii) Change Cipher Spec
- (iii) Application data

(iv) Alert

From the Handshake messages, we extract the server information such as name of the server to which the connection is made. This is used to verify or detect the DNS activity versus server name.

These features once extracted are used by traffic classifier to classify each connection to VPN or normal traffic.

4.2. *Traffic Classification.* After features are extracted, we can classify the incoming traffic as normal traffic or VPN traffic only for the TCP-based connections. TCP connection states

are stored for every new connection. Once the connection is established, it is classified as legitimate or VPN traffic based on extracted features of previous network traffic and new connection. This classification may be as legitimate traffic or VPN traffic. The proposed scheme classifies the incoming connections as shown in Figure 29 and is discussed below.

**4.2.1. IP-Based Classification.** Server IP of each new connection is looked up in an already populated IP-based hash table. This hash table contains the IP list of TOR's exit nodes [11] along with the server IP that were previously classified by the system as VPN servers. This is done to minimize the *resource utilization* against already classified VPN server. If server IP of the current connection is found in this IP-based hash, then the traffic is classified as VPN traffic.

**4.2.2. Server Name-Based Classification.** If the connection is not classified by VPN IP-based hash table, the server name specified in *HTTPS Client Hello* message is used to classify the connection. In a normal TCP/IP-based communication, whenever a service or website needs to be accessed, first its domain name is converted into IP address. This is done to access the resources over the Internet [41]. An IP address at a given time is bound to a specific domain. Using this technique, we classify the normal domains against the domains responsible for VPN Services. This classification can be further divided into two steps.

**4.2.3. No Server Name Analysis.** Against the current server name extracted from the connection, we look up our self-maintained DNS list, populated by network traffic. If no DNS entry is present for that server name in the list or the server IP of the connection is not associated against the given server name, such traffic is classified as VPN traffic. Mostly, inside the initial connection to VPN server, these IPs against DNS are shared with the client's application in SSL-protected channel as to avoid any DNS-based filtering.

**4.2.4. Server Name Analysis.** The server name or the domain name of the current connection is looked up against the well-known VPN server's domain names. The list is maintained to look up the server name; if found, the connection is classified as VPN-based connection. The list is generated by the traffic analysis of these VPN servers, and some unique strings are extracted specific to that VPN service as discussed previously in Section 3.

## 5. System Evaluation

The deployment of our proposed solution, if used only for detection, can be passive as well. Passive deployment will result in *lower latency* as the traffic is being mirrored by the switch or gateway itself. For passive deployment, all the traffic destined outside the network and DNS traffic must pass through the tapped interface as shown in Figure 30.

We analyzed the traffic pattern of well-known available VPN services which use HTTPS protocol for communication. These servers are listed below:

- (i) TOR browser
- (ii) Hotspot Shield free
- (iii) Browsec VPN
- (iv) ZenMate VPN
- (v) Hoxx VPN

The traffic of these VPN services was analyzed, and a selection criterion was built based on the pattern emerging from the analysis. The key features for each VPN service are shown in Table 2. In case of TOR, we see nonstandard HTTPS behavior which means that it may not be on default port 443. We can also detect TOR by *TOR nodes* list populated and updated by community.

In case of Hotspot Shield, we tested two variants of its client. One was the add-on of Firefox web browser, and the other client was desktop application. In case of web browser extension or add-on, Hotspot Shield uses special domain names which are used to uniquely classify the service. In case of desktop application, the client uses nonstandard port for HTTPS with no DNS activity. *Browsec and Hoxx* VPNs both were tested as add-on to the browser, and they are uniquely classified using the domain names the servers use.

All three services discussed above use the same type of domain names across multiple geolocations, e.g., any traffic may be classified as traffic of Hoxx VPN if its domain name contains *\*.klafive.com*. This is not the case for ZenMate VPN. It changes domain names with respect to geolocations chosen by the user. The list of these domain names is communicated during initial connection setup and is updated frequently. This allows VPN services like ZenMate and others to work over a network which uses DNS-based filters, if these filters are not updated frequently.

**5.1. Traffic Generation.** Across multiple systems inside the network, multiple clients of the abovementioned VPN services were installed and configured. These clients were enabled, and network activity was generated by surfing the Internet. The activity was monitored by VPN detector, and alerts were generated once the VPN activity was detected.

**5.2. Traffic Classification Alert.** The alerts generated above for different VPN services were of different types depending upon the activities performed by the users. The generated alerts by five of these users are shown in Table 3.

The alerts shown in Table 3 show the traffic classification of each type of VPN service used with respect to its unique characteristics as discussed in Table 2. Mostly, VPNs may be classified with the help of DNS activity which enable the user to access such services.

The results shown in Table 3 show that the system classified 400 out of 729 active connections as potential VPN connections. Once the system is deployed, any new connection activity in the network is monitored. Each system connected to Internet manages its on DNS cache to reuse

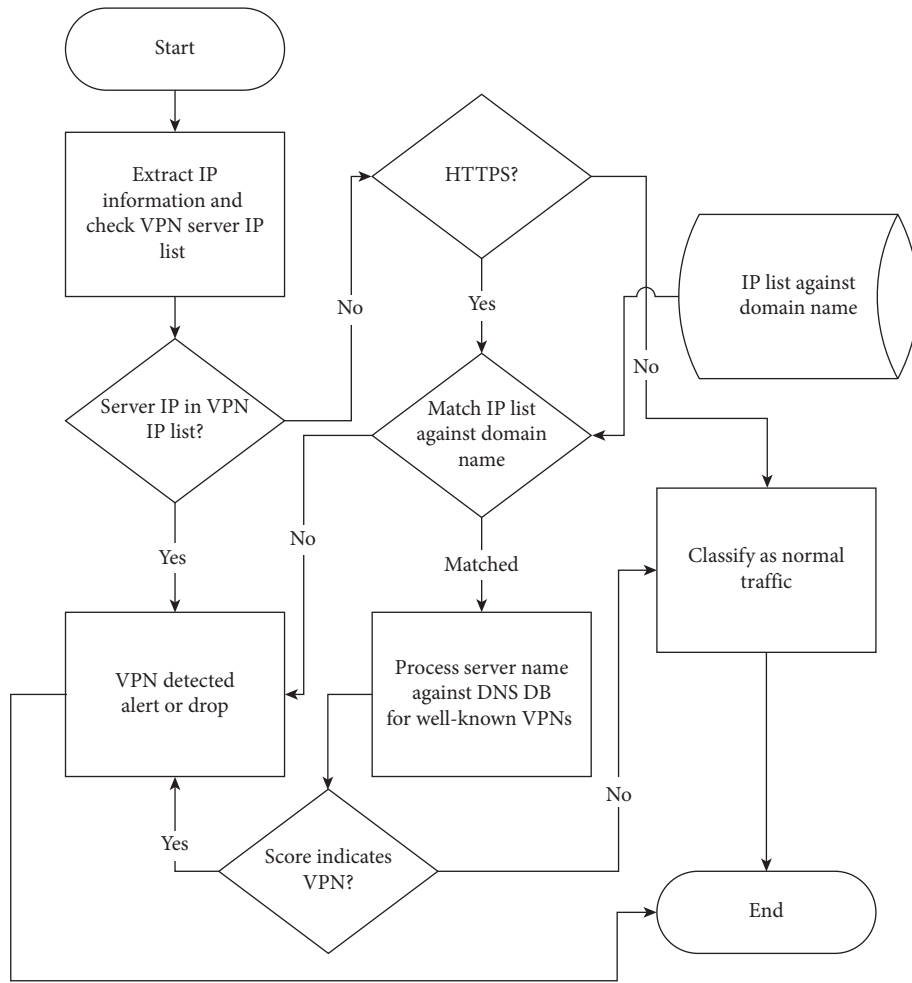


FIGURE 29: Traffic classification.

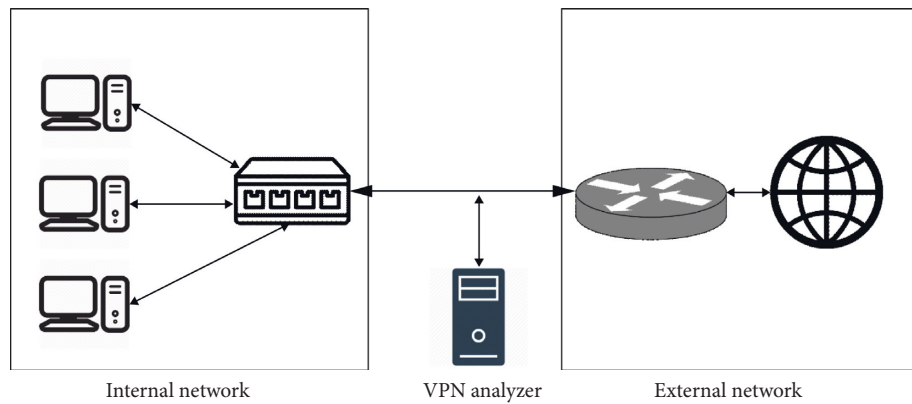


FIGURE 30: Deployment model.

DNS information. If a new connection is made and no DNS activity is present in the system for the server, the system will flag it as potential VPN traffic. To improve system’s precision, the system ignores the already established connections.

VPN classification based on IP and DNS activity may need periodic updates to the lists maintained by the system. Updating this information will increase the overall accuracy of the system and result in less false positives and negatives. Our test shows that, in case of TOR IP analysis, the IP

TABLE 2: Forensic analysis of freely available VPN services.

VPN services	Classifiers for forensic analysis			
	IP	Host name	Nonstandard HTTPS	DNS activity
TOR browser	✓	✗	✓	✓
Hotspot Shield free	✗	✓	✓	✓
Browsec VPN	✗	✓	✗	✗
ZenMate VPN	✗	✓	✗	✗
Hoxx VPN	✗	✓	✗	✗

TABLE 3: Alerts generated for the user activity.

User details	Alerts classification (connection based)				
	Total	Legitimate activity	IP-based VPN	DNS-based VPN	NO DNS
User 1	178	59	4	109	6
User 2	85	50	0	35	0
User 3	250	114	0	135	1
User 4	71	24	2	41	4
User 5	145	82	0	63	0

information should be populated in real time to get better results.

## 6. Conclusion

A VPN service inside an organization may generally be used by an individual to hide the real communication. This communication may be harmful or damage the organization, and the organization may not allow such communication over its monitored network. An organization may not be able to invest heavily on SSL-based proxies to manage its network. This paper proposes a lightweight approach to detect and block unwanted VPN clients inside the organizational network responsible for some illegitimate activity.

Our proposed technique focuses on the information available in plain, which means there is no need to decrypt or decode any network communication. This helps in low *resource utilization*. The proposed solution not only focuses on the current connection but also keeps track of the network activity responsible for this communication, i.e., DNS activity. Such mapping of DNS with its next stream helps identify the normal behavior of the TCP/IP network stack. If no Domain Name information is available for current connection, it may not be normal traffic flow. The scheme also analyzes nonstandard use of HTTPS and detects this anomaly as it is largely used to hide such communication from HTTPS-based filters in firewall.

Results show that our proposed system is able to identify and classify such trends in network traffic and classify the network traffic. The analysis of the VPN services discussed in Table 2 is crucial to detect these services. These service providers keep changing the traffic characteristics for their service. Active analysis of these services must be carried out to keep VPN detector up to date with latest traffic trends.

## Data Availability

The data used to support the findings of this study are provided within the article.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

- [1] B. Harris and R. Hunt, "Tcp/ip security threats and attack methods," *Computer Communications*, vol. 22, no. 10, pp. 885–897, 1999.
- [2] X. Li, M. Wang, H. Wang, Y. Ye, and C. Qian, "Toward secure and efficient communication for the internet of things," *IEEE/ACM Transactions on Networking*, vol. 27, no. 2, pp. 621–634, 2019.
- [3] E. Rescorla, *SSL and TLS: Designing and Building Secure Systems*, vol. 1, Addison-Wesley, Boston, MA, USA, 2001.
- [4] A. P. Felt, R. Barnes, A. King, C. Palmer, C. Bentzel, and P. Tabriz, "Measuring HTTPS adoption on the web," in *Proceedings of the 26th USENIX Security Symposium (USENIX Security 17)*, pp. 1323–1338, USENIX Association, Vancouver, BC, Canada, August 2017.
- [5] J. Clark and P. C. Van Oorschot, "SoK: SSL and HTTPS: revisiting past challenges and evaluating certificate trust model enhancements," in *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, pp. 511–525, IEEE, Berkeley, CA, USA, May 2013.
- [6] C. Paya and O. Dubrovsky, "Inspecting encrypted communications with end-to-end integrity," US Patent 7562211, 2009.
- [7] V. Lifliand and A. Michael Ben-Menahem, "Encrypted network traffic interception and inspection," US Patent 8578486, 2013.
- [8] N. Leavitt, "Anonymization technology takes a high profile," *Computer*, vol. 42, no. 11, pp. 15–18, 2009.
- [9] Z. Zhang, S. Chandel, J. Sun, S. Yan, Y. Yu, and J. Zang, "VPN: a boon or trap?: a comparative study of MPLS, IPSec, and SSL virtual private networks," in *Proceedings of the 2018 2nd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 510–515, IEEE, Erode, India, February 2018.
- [10] K. Karuna Jyothi and B. I. Reddy, "Study on virtual private network (VPN), VPN's protocols and security," *International*

- Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 3, no. 5, 2018.
- [11] D. Roger, N. Mathewson, and S. Paul, "TOR: the second-generation onion router," Technical report, Naval Research Laboratory, Washington, DC, USA, 2004.
  - [12] A. Yamada, Y. Miyake, K. Takemori, A. Studer, and A. Perrig, "Intrusion detection for encrypted web accesses," in *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, vol. 1, pp. 569–576, Niagara Falls, Ont., Canada, May 2007.
  - [13] M. Ikram, N. Vallina-Rodriguez, S. Seneviratne, M. A. Kaafar, and V. Paxson, "An analysis of the privacy and security risks of android VPN permission-enabled apps," in *Proceedings of the 2016 Internet Measurement Conference*, pp. 349–364, ACM, Santa Monica, CA, USA, November 2016.
  - [14] S. Sudin, R. B. Ahmad, and S. Z. Syed Idrus, "A model of virus infection dynamics in mobile personal area network," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 2–4, pp. 197–201, 2018.
  - [15] N. Weaver, C. Kreibich, M. Dam, and V. Paxson, "Here be web proxies," in *Proceedings of the International Conference on Passive and Active Network Measurement*, pp. 183–192, Springer, Los Angeles, CA, USA, March 2014.
  - [16] C. Reis, S. D. Gribble, T. Kohno, and N. C. Weaver, "Detecting in-flight page changes with web tripwires," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, vol. 8, pp. 31–44, San Francisco, CA, USA, April 2008.
  - [17] N. Vallina-Rodriguez, S. Sundaresan, C. Kreibich, and V. Paxson, "Header enrichment or ISP enrichment?: emerging privacy threats in mobile networks," in *Proceedings of the 2015 ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, pp. 25–30, ACM, London, UK, August 2015.
  - [18] N. Weaver, C. Kreibich, and V. Paxson, "Redirecting DNS for ads and profit," in *Proceedings of the UNISEX Workshop on Free and Open Communications on the Internet 2011*, vol. 2, no. 2–3, San Francisco, CA, USA, August 2011.
  - [19] N. Vallina-Rodriguez, J. Amann, C. Kreibich, N. Weaver, and V. Paxson, "A tangled mass: the android root certificate stores," in *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, pp. 141–148, ACM, Sydney, Australia, December 2014.
  - [20] Y. Song and U. Hengartner, "Privacyguard: a VPN-based platform to detect information leakage on android devices," in *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*, pp. 15–26, ACM, Denver, CO, USA, October 2015.
  - [21] S. Fahl, M. Harbach, T. Muders, L. Baumgärtner, B. Freisleben, and M. Smith, "Why eve and mallory love android: an analysis of android ssl (in) security," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, pp. 50–61, ACM, Raleigh, NC, USA, October 2012.
  - [22] V. T. Goh, J. Zimmermann, and M. Looi, "Towards intrusion detection for encrypted networks," in *Proceedings of the 2009 International Conference on Availability, Reliability and Security*, pp. 540–545, IEEE, Fukuoka, Japan, March 2009.
  - [23] A. A. Abimbola, J. M. Munoz, and W. J. Buchanan, "Nethost-sensor: investigating the capture of end-to-end encrypted intrusive data," *Computers & Security*, vol. 25, no. 6, pp. 445–451, 2006.
  - [24] R. Martin, "Snort—lightweight intrusion detection for networks," in *Proceedings of the 13th USENIX Conference on System Administration, LISA '99*, pp. 229–238, USENIX Association, Seattle, WA, USA, November 1999.
  - [25] X. Li, S. G. Karanvir, G. H. Cooper, and J. R. G., "Encrypted data inspection in a network environment," US Patent 9176838, 2013.
  - [26] G. He, B. Xu, and H. Zhu, "AppFA: a novel approach to detect malicious android applications on the network," *Security and Communication Networks*, vol. 2018, Article ID 2854728, 15 pages, 2018.
  - [27] W. Niu, X. Zhang, G. W. Yang, J. Zhu, and Z. Ren, "Identifying APT malware domain based on mobile DNS logging," *Mathematical Problems in Engineering*, vol. 2017, Article ID 4916953, 9 pages, 2017.
  - [28] A. Nath, *Packet Analysis with Wireshark*, Packt Publishing Ltd., Birmingham, UK, 2015.
  - [29] Netressec, Network miner.
  - [30] AnchorFree, Hoptspot Shield VPN.
  - [31] ZenGuard, ZenMate VPN.
  - [32] Browsec LLC, Browsec VPN Your Personal Privacy and Security Online.
  - [33] VPN1.com, Lightning Fast VPN Service | Hoxx VPN | VPN Service for Everyone.
  - [34] P. V. Mockapetris, "Domain names: implementation specification," Technical report, USC/Information Sciences Institute, Marina del Rey, CA, USA, 1983.
  - [35] L. Deri, R. Carbone, and S. Suin, "Monitoring networks using ntop," in *Proceedings of the 2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium (Cat. No. 01EX470)*, pp. 199–212, IEEE, Seattle, WA, USA, May 2001.
  - [36] B. Paul, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement*, pp. 71–82, ACM, New York, NY, USA, 2002.
  - [37] Mohammed Abdul Qadeer, A. Iqbal, M. Zahid, and M. Rahman Siddiqui, "Network traffic analysis and intrusion detection using packet sniffer," in *Proceedings of the 2010 Second International Conference on Communication Software and Networks*, pp. 313–317, IEEE, Singapore, February 2010.
  - [38] J. Postel, "Internet protocol," Technical report, DARPA, Arlington County, VA, USA, 1981.
  - [39] J. Postel, "Transmission control protocol," Technical report, DARPA, Arlington County, VA, USA, 1981.
  - [40] T. Dierks and E. Rescorla, "The transport layer security (TLS) protocol version 1.2," Technical report, 2008.
  - [41] A. F. Behrouz, *TCP/IP Protocol Suite*, McGraw-Hill, Inc., New York, NY, USA, 2nd edition, 2002.

## Research Article

# Outsourcing Hierarchical Threshold Secret Sharing Scheme Based on Reputation

En Zhang , Jun-Zhe Zhu, Gong-Li Li, Jian Chang, and Yu Li

*College of Computer and Information Engineering, Henan Normal University, Xixiang 453007, China*

Correspondence should be addressed to En Zhang; zhangenzdrj@163.com

Received 11 April 2019; Accepted 24 August 2019; Published 10 October 2019

Guest Editor: Mehdi Hussain

Copyright © 2019 En Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Secret sharing is a basic tool in modern communication, which protects privacy and provides information security. Among the secret sharing schemes, fairness is a vital and desirable property. To achieve fairness, the existing secret sharing schemes either require a trusted third party or the execution of a multiround protocol, which are impractical. Moreover, the classic scheme requires expensive computing in the secret verification phase. In this work, we provide an outsourcing hierarchical threshold secret sharing (HTSS) protocol based on reputation. In the scheme, participants from different levels can fairly reconstruct the secret, and the protocol only needs to run for one round. A cloud service provider (CSP) uses powerful computing resources to help participants complete homomorphic encryption and complex verification operations, and the CSP cannot be aware of any valuable information. The participants can obtain the secret with a small number of operations. To avoid collusion, we suppose that participants have their own reputation value, and they are punished or rewarded according to their behavior. The reputation value of a participant who deviates from the protocol will decrease; therefore, the participant will choose a cooperative strategy to obtain better payoffs. Lastly, our scheme is proved to be secure, and experiments indicate that our scheme is feasible and efficient.

## 1. Introduction

Secret sharing is an important cryptographic primitive and has a widespread application in secure multiparty computation, image encryption, and attribute-based encryption. Secret sharing, an idea proposed by Shamir [1] and Blakley [2], allows a dealer to distribute different shares among a set of participants. The method guarantees any authorized subsets of  $t$  or more participants can reconstruct the secret. However, it is hard to guarantee that the dealer and participants are absolutely honest. To address this problem, verifiable secret sharing (VSS) schemes [3–5] guarantee additionally any cheating behavior can be detected, which can check the validity of shares. Subsequently, a series of protocols [6–9] is studied sharing multiple secrets at a time. In these schemes, participants only need to submit a pseudoshare rather than a real share to recover multiple secrets. Secret sharing has become an important research topic, and a large quantity of studies have been proposed. A multistage secret sharing scheme was introduced by Pilaram

and Eghlidis [10], which was based on Lattice and could resist quantum attacks. Zhang et al. [11] presented an outsourcing secret sharing scheme based on homomorphic encryption, but the scheme could not effectively resist collusion. Recently, secret sharing has stronger privacy requirements. Although information about shares is leaked, the adversary still has no access to information about secret. Fehr and Yuan [12] constructed a robust secret sharing scheme with security against a rushing adversary. Benhamouda et al. studied leakage resilience of the MPC protocol [13]. A nonmalleable scheme concerning secret sharing was presented by Goyal and Kumar [14]. The scheme can resist adversary of someone who arbitrarily tampers with shares. Later, Goyal and Kumar [15] proposed nonmalleable secret sharing schemes for more general access structures.

In real life, everyone is not exactly equal in status or privileges. It would be an endless task to cite such living examples. For example, in a research and development department of a company, the shares of the private key of confidential files may be distributed among employees.

Some are accountants, and some are department managers. The company's policy requires 3 employees to be in attendance at the same time to open confidential files, but at least one of them must be a department manager. Such a setting requires a special secret sharing method. Therefore, the concept of HTSS was proposed. Tassa [16] introduced the structure of HTSS. In the scheme, a secret is shared among participants that are divided into different levels. Only participants who meet a certain level can reconstruct the secret. If the specific level is not met, the participants learn nothing about the secret. Later, Traverso et al. [17] proposed an HTSS scheme that supports verifiability and dynamics, which can add, remove, and renew shares. Recently, Mohamed and Arockia [18] introduced an HTSS scheme for color images. Bhattacharjee et al. [19] presented a hierarchical image scheme for bandwidth efficient transmission and offered a great degree of robustness in compressed sensing.

In the classic secret sharing scheme, fairness is a desirable property that guarantees each participant can gain the secret simultaneously. For the purpose of the goal, Tompa and Woll [20] firstly introduced a fair reconstruction scheme. The main idea of the scheme is to hide the real secret value, and the cheater has to guess the secret location. However, it is impractical for all participants to release their shares synchronously. A novel fair threshold scheme was presented by Tian et al. [21]. In the work, the real secret value was hidden in the sequence for the sake of decreasing the probability of the cheater achieving a successful guess. Combining the approach with game theory, Halpern and Teague [22] introduced a rational cryptographic protocol. In the rational scheme, the participants are rational players whose behavior aims are to maximize their profit. To achieve fairness, existing schemes require either a trusted third party or the execution of a multiround protocol, which are impractical.

The reputation system plays a key role in the online community, such as auction markets, trusted content delivery, and e-commerce. By publicizing the reputation value, participants can choose trusted peers with whom to cooperate. Reputation systems can effectively combat selfish, dishonest, and malicious behavior. Xiong and Liu [23] presented a detailed explanation. Combining with reputation systems, Zhang et al. [24] proposed a PSI protocol against social rational participants in which the parties who defect the PSI protocol will be penalized. Nojournian and Stinson [25] introduced a socio-rational protocol. In this paper, participants are invited to execute an unknown number of protocols based on their reputation. Recently, a series of works were proposed. Litos and Zindros [26] created a reputation network in which the reputation value is quantifiable and expressed in monetary terms. Clark et al. [27] presented a dynamic, privacy-preserving decentralized reputation system.

At present, the vast amount of data stored in the cloud has led to explosive growth in the data volume. People are entering the era of big data, and everything will be digitized. According to the statistics of the Millet cloud storage service, the number of customers at the end of 2015 reached

97 million, with 46.5 billion photos and 504 million videos. It is estimated that by 2020, the global data volume will reach 44 ZB. At the same time, cloud outsourcing computing is also very common. More and more devices with poor computing power such as smart phones, pads, and sensors can outsource computing to a CSP with powerful computing power so that users can enjoy unlimited computing resources. However, in the face of outsourcing computing, users are reluctant to disclose their personal sensitive data. Therefore, we need to find a practical approach to implement an HTSS scheme.

*1.1. Our Contribution.* We provide an outsourcing HTSS protocol based on reputation, as is demonstrated in Figure 1. In this protocol, secret shares are distributed to different levels of participants. The participants can obtain the secret fairly with a small quantity of operations. Expensive computing is outsourced to a CSP, and the CSP can gain nothing about the secret. Moreover, the reputation system can effectively prevent participants from colluding with the server. Compared with previous schemes, our scheme has the following advantages:

- (1) The participants are not required to always be online, which avoids multiple interactions between the participants and the server.
- (2) The protocol could accurately check the malicious behavior of the participants or the server.
- (3) Expensive computing is outsourced to a CSP. With the CSP's computing power, the CSP can execute homomorphic encryption and complex verification operations, and the server can gain nothing about the secret.
- (4) Through a combination with the reputation system, we design a social game model for the hierarchical secret sharing scheme, which can resist collusion between the participant and the server. Assuming that participants have their own reputation value, they are punished or rewarded according to their behavior. Moreover, all participants are rational players whose behavior aims are to maximize their profit. The reputation value of a participant deviating from the protocol will decrease. In our model, a participant who chooses a cooperative strategy can obtain better payoffs. Therefore, each participant will honestly abide by the protocol.

We formally describe preliminaries in Section 2. We construct an outsourcing HTSS scheme based on reputation in Section 3. We indicate the security of the scheme in Section 4 and compare our scheme with previous schemes in Section 5. Finally, the conclusion of our paper is presented in Section 6.

## 2. Preliminary

*2.1. Secret Sharing Homomorphisms.* Benaloh [28] described the homomorphic property of secret sharing. For example, consider two secrets  $k_1$  and  $k_2$ , which are shared by



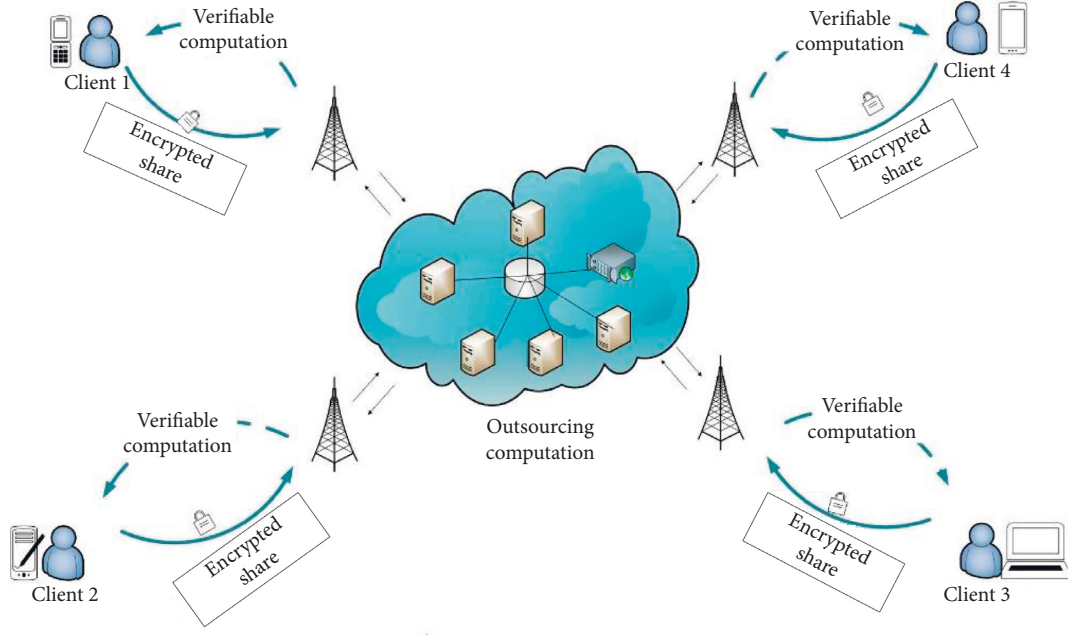


FIGURE 1: Outsourcing the hierarchical secret sharing scheme based on reputation.

polynomials  $\varphi(x)$  and  $\phi(x)$ . If we add the shares  $f(i) = \varphi(i) + \phi(i)$ ,  $1 \leq i \leq n$ , each of  $f(i)$  can be viewed as a subshare of secret  $k_1 + k_2$ . Suppose that  $K$  is defined as the secret domain, and  $\Sigma$  is defined as the share domain. A set of functions  $F_I: \Sigma^t \rightarrow K$  can be determined, where  $I \subseteq \{1, 2, \dots, n\}$  and  $|I| = t$ . Given any set of  $t$  values  $k_{i_1}, \dots, k_{i_t}$ , the following equation can define the secret  $k$ :

$$k = F_I(k_{i_1}, \dots, k_{i_t}), \quad \text{for } I = \{i_1, \dots, i_t\}. \quad (1)$$

*Definition 1.* Suppose  $\oplus$  and  $\otimes$  are two operations on the secret domain  $K$  and share domain  $\Sigma$ , respectively. There are

$$\begin{aligned} k &= F_I(k_{i_1}, \dots, k_{i_t}), \\ k' &= F_I(k'_{i_1}, \dots, k'_{i_t}), \end{aligned} \quad (2)$$

then,

$$k \oplus k' = F_I(k_{i_1} \otimes k'_{i_1}, \dots, k_{i_t} \otimes k'_{i_t}). \quad (3)$$

From the above definition, Shamir's polynomial is  $(+, +)$ -homomorphic, which implies that the sum of the shares is equivalent to shares of the sum.

**2.2. Tassa's  $(\mathbf{t}, n)$  Hierarchical Threshold Scheme.** In HTSS, a set of participants  $P = \{P_1, \dots, P_n\}$  are split into multiple levels  $U_0, U_1, \dots, U_m$ , where  $U_0$  is the highest level and  $U_m$  is the lowest level. For all  $0 \leq i < j \leq m$ , there is  $P = \cup_{i=0}^m U_i$ , where  $U_i \cap U_j = \emptyset$ . Supposing that  $n_h$  is the number of participants associated with level  $U_h$ , we can obtain  $n = |P| = \sum_{h=0}^m n_h$ . Then, we define a threshold  $t_h$  associated with level  $U_h$ , for  $h = 0, \dots, m$ , which satisfies  $0 < t_0 < \dots < t_m$ . In addition, we set  $\mathbf{t} = \{t_h\}_{h=0}^m$ ,  $t = t_m$ , and

$t_{-1} = 0$ . Therefore, the  $(\mathbf{t}, n)$  hierarchical access structure  $\Gamma$  is described as follows:

$$\Gamma = \left\{ A \subset P : \left| A \cap \left( \bigcup_{j=0}^i U_j \right) \right| \geq t_i, \quad \forall i \in \{0, 1, \dots, m\} \right\}. \quad (4)$$

Next, we describe in detail how the Birkhoff interpolation reconstructs the secret.

The Birkhoff interpolation problem is to find a polynomial  $F(x) = \sum_{r=0}^{t-1} a_r x^r \in \mathbb{R}_{t-1}[x]$  that satisfies the equalities  $F(i) = \sigma_{i,j}$ , where  $F^j(i)$  is the  $j$ -th derivative of  $F(x)$  at position  $i$ . Suppose that an authorized subset  $R \in \Gamma \subset P$  can reconstruct the secret.  $E$  associated with  $R$  is a matrix with binary entries. If there is participant  $p_{i,j}$  with share  $\sigma_{i,j}$ , then the entry  $e_{i,j}$  is set to "1". In addition, we set  $\varphi = \{1, x, \dots, x^t\} = \{\omega_0, \omega_1, \dots, \omega_{t-1}\}$  and define  $\omega_r^j$  as the  $j$ -th derivative of  $\omega_r$ . The matrix  $A(E, X, \varphi)$  can be expressed as follows:

$$A(E, X, \varphi) = \begin{pmatrix} \omega_0^{j_1}(i_1) & \omega_1^{j_1}(i_1) & \cdots & \omega_{t-1}^{j_1}(i_1) \\ \omega_0^{j_2}(i_2) & \omega_1^{j_2}(i_2) & \cdots & \omega_{t-1}^{j_2}(i_2) \\ \vdots & \vdots & \vdots & \vdots \\ \omega_0^{j_r}(i_r) & \omega_1^{j_r}(i_r) & \cdots & \omega_{t-1}^{j_r}(i_r) \end{pmatrix}, \quad (5)$$

where  $r = 0, \dots, t-1$ .

The polynomial  $F(x)$  can be reconstructed:

$$F(x) = \sum_{r=0}^{t-1} \frac{|A(E, X, \varphi_r)|}{|A(E, X, \varphi)|} x^r, \quad (6)$$

in which we can obtain  $A(E, X, \varphi_r)$  by replacing the  $(r+1)$ -th column with the shares  $\sigma_{i,j}$  in the lexicographic order.

*Definition 2.* Let  $M$  be a message space,  $\Sigma$  be a share space, and  $\Gamma$  be an access structure where  $t_h$  is the threshold associated with level  $U_h$ . Suppose that the pair  $(i, j)$  is the identity of participant  $p_{i,j} \in U_h$ . Then, an HTSS scheme contains the *share phase* and *reconstruction phase*.

*Share Phase.* A dealer outputs  $n$  shares  $\sigma_{i,j} \in \Sigma$  that is distributed to participant  $p_{i,j} \in U_h$ .

*Reconstruction Phase.* An authorized subset  $R$  of  $t$  participants, which satisfies  $R \in \Gamma$ , can reconstruct the secret  $k \in M$  using Birkhoff interpolation.

**2.3. Social Game Model of Secret Sharing.** Reputation systems can provide an incentive for honest behavior and help people decide who is trustworthy. Several reputation systems have been deployed in practical applications, such as encouraging compliance with e-commerce contracts. Next, we briefly review the related concepts and methods in [25].

*Definition 3.* Let  $T_i^j(p)$  be the trust value assigned by participant  $P_j$  to  $P_i$  during period  $p$ . Let  $T_i : \mathbb{N} \mapsto \mathbb{R}$  be the trust function computing the reputation of  $P_i$ :

$$T_i(p) = \frac{1}{n-1} \sum_{j \neq i} T_i^j(p), \quad \text{where } -1 \leq T_i(p) \leq +1 \text{ and } T_i(0) = 0. \quad (7)$$

The monotonically increasing function  $\mu(x)$  and the monotonically decreasing function  $\mu'(x)$  are used to update reputation values recursively, that is, computing  $T_i(p)$  by  $T_i(p-1)$ . If participant  $P_i$  has a choice of cooperating during period  $p$ , then  $T_i(p) = T_i(p-1) + \mu(x)$ . If participant  $P_i$  has a choice of defecting during period  $p$ , then  $T_i(p) = T_i(p-1) - \mu'(x)$ .

Subsequently, we review the payoff assumption. Let  $u_i(a)$  be  $P_i$ 's payoff by considering future action, let  $\mu'_i(a)$  be  $P_i$ 's payoff by considering current action, let  $l_i(a) \in \{0, 1\}$  define whether the participant is aware of secret during period  $p$ , and define  $\text{num}(a) = \sum l_i(a)$ . The generalized payoff assumptions of social games are as follows:

- (A)  $l_i(a) = l_i(a')$  and  $T_i^{a'}(p) > T_i^a(p) \implies u_i(a) > u_i(a')$
- (B)  $l_i(a) > l_i(a') \implies u_i(a) > u_i(a')$
- (C)  $l_i(a) = l_i(a')$  and  $\text{num}(a) < \text{num}(a') \implies u_i(a) > u_i(a')$

*Remark 1.* A, B, and C have impact factors  $\rho_1, \rho_2$ , and  $\rho_3$ , respectively, where  $\rho_1 \gg \rho_2 \geq \rho_3$ .

Let

$$\omega_i(a) = \frac{3}{2 - T_i^a(p)}. \quad (8)$$

We can obtain the current payoff  $u'_i(a)$  and the future payoff  $u_i(a)$  as follows:

$$u'_i(a) = \rho_2 l_i(a) + \rho_3 \frac{l_i(a)}{\text{num}(a) + 1}, \quad (9)$$

$$u_i(a) = \rho_1 \frac{|T_i^a(p) - T_i^a(p-1)|}{T_i^a(p) - T_i^a(p-1)} \times \omega_i(a) + u'_i(a).$$

### 3. The HTSS Scheme Based on Reputation

In this section, combining an outsourcing computation and the reputation system, we propose a novel outsourcing HTSS protocol based on reputation. In the protocol,  $t$  or more parties from different levels can recover the secret. The scheme contains five phases: an initialization phase, a secret distribution phase, an outsourcing phase, a reconstruction phase, and a reputation update phase. We formally defined some parameters during the initialization phase. In the secret distribution phase, a dealer distributes encrypted shares and broadcasts verification information and participants receive a random value and encrypted shares. Then, the participants send shares to a CSP, and the CSP returns the results to the participants where the CSP cannot be aware of any valuable information about the secret. Next, the participants can obtain the secret fairly in the reconstruction phase. Finally, we can update the participant's reputation value. To avoid collusion, participants have their own reputation value and they are punished or rewarded according to their behavior. For example, if a participant wants to collude with the CSP and sends a collusion invitation to the CSP, then we can penalize the participant according to the reputation system.

**3.1. Initialization Phase.** Let  $p$  and  $q$ , such as  $(q | p-1)$ , be two large primes,  $g$  be a generator of the  $q$ -th order subgroup  $\mathbb{F}_q^*$  of  $\mathbb{F}_p^*$ , and  $H(x)$  be a collision-resistant hash function.

A secret  $k$  is shared among  $n$ -parties, and a set of parties denoted by  $P = \{P_1, \dots, P_n\}$  are split into multiple levels  $U_0, U_1, \dots, U_m$ .  $n_h$  is the number of participants associated with level  $U_h$ , and  $t_h$  is the threshold associated with level  $U_h$ , for  $h = 0, \dots, m$ . The pair  $(i, j)$  is the identity of participant  $p_{i,j} \in U_h$ , for  $i = 1, \dots, n_h$ ,  $j = t_{h-1}$ , and  $t_{-1} = 0$ .

**3.2. Secret Distribution Phase.** The trusted dealer distributes shares by performing the following stages:

*Step 1.* The dealer randomly chooses  $t-1$  coefficients  $a_1, \dots, a_{t-1} \in \mathbb{F}_q$  and generates a polynomial with  $t-1$  degree:

$$f(x) = \sum_{r=0}^{t-1} a_r x^r \text{ mod } q, \quad (10)$$

where  $a_0$  is a secret value, i.e.,  $k = a_0$ . The corresponding shares are  $\sigma_{i,j} = f^j(i)$ , where  $f^j(i)$  is the  $j$ -th derivative of the polynomial  $f(x)$  at position  $i$ .

*Step 2.* The dealer randomly chooses  $t - 1$  coefficients  $a'_1, \dots, a'_t \in \mathbb{F}_q$  and generates a polynomial with  $t - 1$  degree:

$$f'(x) = \sum_{r=0}^{t-1} a'_r x^r \text{ mod } q, \quad (11)$$

where  $a'_0$  distributed to all participants is a random value. The corresponding shares are  $\sigma'_{i,j} = f'^j(i)$ .

*Step 3.* According to the  $(+, +)$ -homomorphic property, the sum of the shares is equivalent to the shares of the sum, and the dealer performs the following operation:

$$\xi_{i,j} = \sigma_{i,j} \otimes \sigma'_{i,j} = f^j(i) \otimes f'^j(i). \quad (12)$$

*Step 4.* The dealer distributes  $(\xi_{i,j}, H(a_0))$  to participant  $p_{i,j} \in U_h$ , for  $i = 1, \dots, n_h$ ,  $j = t_{h-1}$ , and  $h = 0, \dots, m$ .

*Step 5.* The dealer broadcasts verification information:

$$c_r = g^{a_r \otimes a'_r} \text{ mod } p, \quad r = 0, \dots, t - 1. \quad (13)$$

**3.3. Outsourcing Phase.** Suppose that  $t$  or more participants from different levels commit their shares, and then they will perform the following stages:

*Step 1.* An authorized subset of  $t$  participants sent  $(\xi_{i,j}, c_r)$  to the CSP.

*Step 2.* According to following equation, the CSP checks whether  $(\xi_{i,j}, c_r)$  is correct:

$$g^{\xi_{i,j}} \equiv \prod_{r=j}^{t-1} c_r^{(r!/(r-j)!)i^{r-j}} = g^{f^j(i)} \text{ mod } p, \quad (14)$$

where  $r = 0, \dots, t - 1$ . The CSP performs Step 3 if the above equation is held; otherwise, the protocol is terminated and the deception of participant  $p_{i,j}$  will be disclosed.

*Step 3.* The CSP uses Birkhoff interpolation to reconstruct  $f(x)$  with  $\xi_{i,j}$ :

$$f(x) = \sum_{r=0}^{t-1} \frac{|A(E, X, \varphi_r)|}{|A(E, X, \varphi)|} x^r. \quad (15)$$

According to the above equation, the CSP can learn  $k' = F(0) = k \oplus a'_0$  and send  $k'$  to  $t$  participants.

**3.4. Reconstruction Phase.** Each participant can obtain the secret with a small amount of computation according to the following steps:

*Step 1.* The participant can obtain the secret  $k$  by  $k = k' \oplus a'_0$ .

*Step 2.* The participant can verify secret  $k$  according to the following equation:

$$H(a_0) = H(k). \quad (16)$$

If the equation is true, CSP's calculation is correct; otherwise, it is wrong.

**3.5. Reputation Update Phase.** The reputation value updates as follows:

*Case 1.* If  $P_k (1 \leq k \leq n + 1)$  sends a collusion to  $P_{j \neq k} (1 \leq k \leq n + 1)$  and  $P_j$  has a choice of colluding with  $P_k$ , then the colluder earns  $\rho_4$ , where  $\rho_1 \gg \rho_2 \geq \rho_3 \geq \rho_4$  and  $P_{n+1} = \text{CSP}$ .

*Case 2.* If  $P_j$  has a choice of not to collude with  $P_k$  and broadcasts his malicious behavior, then  $P_j$ 's reputation value will increase. In contrast,  $P_k$ 's reputation value will decrease.

*Case 3.* If each participant has a choice of cooperating, then the reputation value will increase; otherwise, the reputation value will decrease.

## 4. Security Analysis

In the section, we give the analysis of the protocol.

**Theorem 1.** *The outsourcing HTSS scheme is secure and any  $t - 1$  or fewer participants get nothing about the secret.*

*Proof.* (a) Any  $t - 1$  or fewer participants get nothing about the secret.

In the scheme, any  $t - 1$  or fewer participants' collusion from different levels cannot obtain the secret with their subshares  $\xi_{i,j}$  for  $i = 1, \dots, n_h$ ,  $j = t_{h-1}$ , and  $h = 0, \dots, m$  because the Birkhoff interpolation requires  $t$  values to determine the unique solution.

(b) The CSP cannot be aware of any valuable information about the secret.

The scheme protects the participant's privacy, and the CSP does not know the participant's input and output. An authorized subset of  $t$  participants sends encrypted share  $\xi_{i,j}$  to the CSP. Therefore, the CSP cannot be aware of any valuable information about the secret.  $\square$

**Theorem 2.** *The outsourcing HTSS scheme can verify malicious behavior, and the malicious behavior can be detected in time.*

*Proof.* (a) The participants and the CSP can check invalid shares.

The public verification information  $c_r = g^{a_r \otimes a'_r}$  can check shares whether is correct, and a commitment to the  $\xi_{i,j}$  can be expressed by the following equation:

$$\begin{aligned}
g^{\xi_{i,j}} &= g^{\sigma_{i,j} \otimes a'_{i,j}} = g^{(a_0 + a_1 i + \dots + a_{t-1} i^{t-1})^{(j)} \otimes (a'_0 + a'_1 i + \dots + a'_{t-1} i^{t-1})^{(j)}} \\
&= g^{(a_0 \otimes a'_0) + (a_1 \otimes a'_1) i^{(j)} + \dots + (a_{t-1} \otimes a'_{t-1}) i^{t-1(j)}} \\
&= (\alpha_0)^{(j)} (\alpha_1) i^{(j)} \dots (\alpha_{t-1}) i^{t-1(j)}.
\end{aligned} \tag{17}$$

Thus, the validity of  $\xi_{i,j}$  can be checked:

$$g^{\xi_{i,j}} \equiv \prod_{r=j}^{t-1} c_r^{(r!/(r-j)!)i^{r-j}} = g^{f^j(i)} \pmod{p}, \tag{18}$$

and the malicious behavior can be detected in time.

(b) The participants can verify the CSP's calculation result.

The participants can verify the calculation result by a collision-resistant hash function. If  $H(a_0) = H(k)$ , the participants can confirm that the CSP's calculation is correct; otherwise, the result is incorrect. Moreover, the participants can detect the CSP's malicious behavior in time.  $\square$

**Theorem 3.** *The scheme is a social Nash equilibrium and collusion-free if the rational participant chooses a cooperation strategy.*

*Proof.* (a) The scheme is not secure if the participants collude with the CSP.

The scheme cannot resist collusion between the server and other participants. In the scheme, if  $P_i$  receives the CSP's collusion invitation and sends  $a'_0$  to the CSP, then the CSP can obtain the real secret  $k$  instead of  $k' = k \oplus a'_0$ .

(b) Following the method in [25], we consider all participants are rational. Let  $\text{Coop}_j$  define that participant  $P_j$  chooses a cooperation strategy where  $1 \leq j \leq n+1$  and  $P_{n+1} = \text{CSP}$ , let  $\text{Coll}_j$  define that  $P_j$  chooses a collusion strategy, let  $\text{Coop}_{-j}$  denote that all participants choose a cooperation strategy except for  $P_j$ , and let  $\text{Coop}_{-i||j}$  denote that all the participants choose a cooperation strategy except for  $P_i$  and  $P_j$ .

If all the participants have a choice of cooperating denoted by  $(\text{Coop}_j, \text{Coop}_{-j})$ , then the payoff functions for choosing cooperation strategy are  $u_i^{(\text{Coop}_j, \text{Coop}_{-j})} = \Omega(\rho_1 \omega_i + \rho_2 + (\rho_3/(n+1)))$  and  $u_{n+1}^{(\text{Coop}_j, \text{Coop}_{-j})} = \rho_1 \omega_{n+1}$  where  $\omega_i = 3/(2 - T_i(p))$ .

If  $P_i$  invites CSP to collude and the CSP has a choice of colluding with  $P_i$  with a probability of 0.5, then the payoff functions for choosing colluding strategy are  $u_i^{(\text{Coll}_i, \text{Coll}_{n+1}, \text{Coop}_{-i||n+1})} = \rho_1 \omega_i + \rho_2 + (\rho_3/(n+2)) + \rho_4$  and  $u_{n+1}^{(\text{Coll}_i, \text{Coll}_{n+1}, \text{Coop}_{-i||n+1})} = \rho_1 \omega_{n+1} + \rho_2 + (\rho_3/(n+2)) + \rho_4$  where  $\rho_1 \gg \rho_2 \geq \rho_3 \geq \rho_4$ ; otherwise, if CSP has a choice of not to collude with  $P_i$  with a probability of 0.5 and publishes his malicious behavior, then  $u_i^{(\text{Coll}_i, \text{Coop}_{-i})} = -\rho_1 \omega_i$  and  $u_{n+1}^{(\text{Coll}_i, \text{Coop}_{-i})} = \rho_1 \omega'_{n+1}$ , where  $\omega'_i = 3/(2 - (T_i(P) + \mu(x)))$ . If the CSP invites  $P_i$  to collude and  $P_i$  has a choice of colluding with CSP, then  $u_{n+1}^{(\text{Coll}_{n+1}, \text{Coll}_i, \text{Coop}_{-i||n+1})} = \rho_1 \omega_{n+1} + \rho_2 + (\rho_3/(n+2)) + \rho_4$  and  $u_i^{(\text{Coll}_{n+1}, \text{Coll}_i, \text{Coop}_{-i||n+1})} = \rho_1 \omega_i + \rho_2 + (\rho_3/(n+2)) + \rho_4$ ; otherwise, if  $P_i$  has a choice of not to collude

TABLE 1: Computation time (in ms).

Client	$t=3$	$t=4$	$t=5$	$t=6$
Time of secret verification	791.52	868.21	1103.42	7370.42
Time of secret reconstruction	2.17	2.19	2.31	2.74

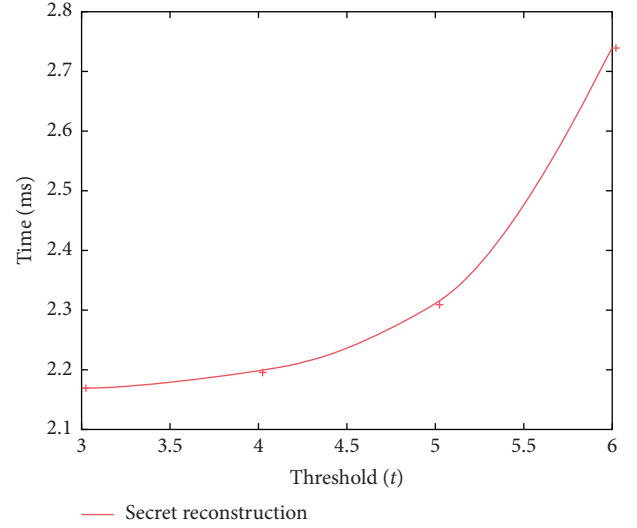


FIGURE 2: Secret reconstruction time.

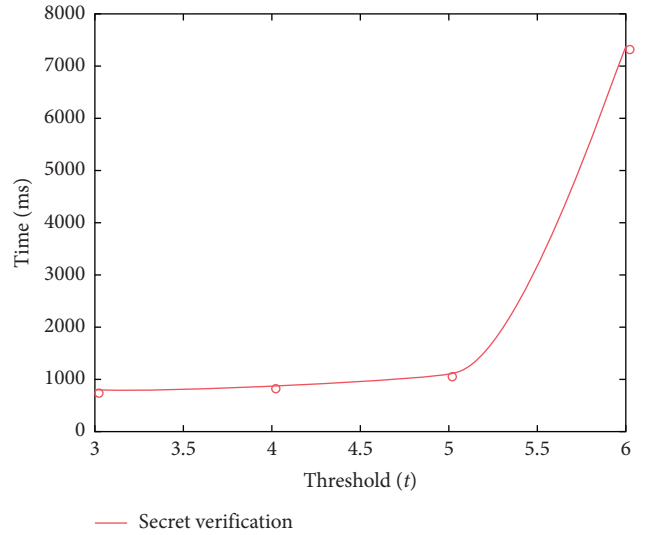


FIGURE 3: Secret verification time.

with CSP and publishes his malicious behavior, then  $u_{n+1}^{(\text{Coll}_{n+1}, \text{Coop}_{-(n+1)})} = -\rho_1 \omega_{n+1}$  and  $u_i^{(\text{Coll}_{n+1}, \text{Coop}_{-(n+1)})} = \rho_1 \omega'_i$ . The payoff function of  $P_i$  choosing a collusive strategy is  $u_i = 1/2(\rho_2 + (\rho_3/(n+2)) + \rho_4)$ , and the payoff function of  $P_i$  choosing a cooperative strategy is  $u_i = 1/2(\rho_1 \omega_i + \rho_2 + (\rho_3/(n+2)) + \rho_1 \omega'_i)$ . The payoff function of the CSP choosing a collusive strategy is  $u_{n+1} = 1/2(\rho_2 + (\rho_3/(n+2)) + \rho_4)$ , and the payoff function of the CSP choosing a cooperative strategy is  $u_{n+1} = 1/2(\rho_1 \omega'_{n+1} + \rho_1 \omega_{n+1})$ . The payoff function of cooperative strategy is larger than that of collusive strategy. From the above statements, we can conclude that choosing cooperation is the optimal strategy.  $\square$

TABLE 2: Feature comparison of schemes.

	Maleka et al. [29]	Harn et al. [30]	Pilaram and Eghlidis [10]	Traverso et al. [17]	Our scheme
Fairness	No	Yes	Yes	No	Yes
Number of rounds	Multiple rounds	Multiple rounds	One round	One round	One round
Trusted third party	No	No	Yes	Yes	No
Interactive	Yes	Yes	No	Yes	No
Computation	User	User	User	User	CSP
Communication cost	$O(tk)$	$O(t)$	$O(1)$	$O(t)$	$O(1)$

## 5. Performance Analysis

We evaluated the prototype on a PC which has an Intel Core i7-6700 CPU (4-core 2.60 GHz) and 8 GB of RAM. To ignore network latency, we run the server and all clients on the same host. The times of the secret verification and secret reconstruction are given in Table 1. In Figure 2, the curve shows the reconstruction time of the scheme. According to the test results, the time varies from 2.17 ms to 2.74 ms. Figure 3 shows the time of the verification, and as the number of participant increases, the verification time increases exponentially. According to the test results, the time varies from 791.52 ms to 7370.42 ms. We conclude that the secret reconstruction requires less time than the verification algorithm.

In addition, we listed our comparison results in Table 2. Maleka et al. [29] analyzed a finite repeated game and an infinite repeated game, but the scheme could not effectively guarantee fairness. Traverso et al. [17] proposed an HTSS scheme that supports verifiability and dynamics, which can add, remove, and renew shares. Although the scheme can check invalid shares, the scheme cannot effectively guarantee fairness. A multistage secret sharing scheme was introduced by Pilaram and Eghlidis [10], which was based on Lattice and could resist quantum attacks. But this scheme requires a trusted third party. In order to achieve desire of fairness, Harn et al. [30] proposed a fair secret sharing scheme, but the scheme requires multiple protocol rounds and cannot be effectively applied to devices with poor computing capabilities.

In contrast, our scheme only needs to execute the protocol once. The participants only need to perform the decryption operation, and the communication cost is  $O(1)$ . In the proposed scheme, complex operations such as homomorphic encryption and verification are outsourced to the CSP. Moreover, our scheme does not require participants to always be online.

## 6. Conclusion

Combining outsourcing computation and a reputation system, we provide an outsourcing HTSS protocol based on reputation. The participants can obtain the secret fairly with a small number of operations in this work. Expensive computing is outsourced to a CSP, and the CSP could learn nothing about the secret. The reputation system can effectively prevent participants from colluding with the server. Participants have their own reputation value, and they are punished or rewarded according to their behavior. Moreover, our protocol could accurately check the malicious

behavior of the participants or the server and does not require multiple interactions between the participants and the server, which applies to cloud computing environments and mobile networks.

## Data Availability

All data generated or analyzed during this study are included in this published article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (U1604156, 61772176, and 61602158) and Science and Technology Research Project of Henan Province (172102210045 and 192102210131).

## References

- [1] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.
- [2] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the American Federation of Information Processing Societies (AFIPS'79) National Computer Conference*, vol. 48, pp. 313-317, CA, USA, February 1979.
- [3] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults," in *Proceedings of the 26th Annual Symposium on Foundations of Computer Science*, pp. 383-395, IEEE, Portland, OR, USA, October 1985.
- [4] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, pp. 427-438, IEEE, Los Angeles, CA, USA, October 1987.
- [5] T. P. Pedersen, "Distributed provers with applications to undeniable signatures," in *Advances in Cryptology-EUROCRYPT*, pp. 221-242, Springer, Berlin, Germany, 1991.
- [6] C. Blundo, A. De Santis, and U. Vaccaro, "Efficient sharing of many secrets," in *Proceedings of the Annual Symposium on Theoretical Aspects of Computer Science*, pp. 692-703, Springer, Würzburg, Germany, February 1993.
- [7] H.-Y. Chien, J.-K. Jan, and Y.-M. Tseng, "A practical  $(t, n)$  multi-secret sharing scheme," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 83, no. 12, pp. 2762-2765, 2000.
- [8] L.-J. Pang and Y.-M. Wang, "A new  $(t, n)$  multi-secret sharing scheme based on Shamir's secret sharing," *Applied Mathematics and Computation*, vol. 167, no. 2, pp. 840-848, 2005.

- [9] C.-C. Yang, T.-Y. Chang, and M.-S. Hwang, "A  $(t, n)$  multi-secret sharing scheme," *Applied Mathematics and Computation*, vol. 151, no. 2, pp. 483–490, 2004.
- [10] H. Pilaram and T. Eghlidos, "An efficient lattice based multi-stage secret sharing scheme," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 1, pp. 2–8, 2017.
- [11] E. Zhang, J. Peng, and M. Li, "Outsourcing secret sharing scheme based on homomorphism encryption," *IET Information Security*, vol. 12, no. 1, pp. 94–99, 2018.
- [12] S. Fehr and C. Yuan, "Towards optimal robust secret sharing with security against a rushing adversary," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Darmstadt, Germany, May 2019.
- [13] F. Benhamouda, A. Degwekar, Y. Ishai, and T. Rabin, "On the local leakage resilience of linear secret sharing schemes," in *Proceedings of the Annual International Cryptology Conference*, pp. 531–561, Springer, Santa Barbara, CA, USA, August 2018.
- [14] V. Goyal and A. Kumar, "Non-malleable secret sharing," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pp. 685–698, ACM, Los Angeles, CA, USA, June 2018.
- [15] V. Goyal and A. Kumar, "Non-malleable secret sharing for general access structures," in *Proceedings of the Annual International Cryptology Conference*, pp. 501–530, Springer, Santa Barbara, CA, USA, August 2018.
- [16] T. Tassa, "Hierarchical threshold secret sharing," *Journal of Cryptology*, vol. 20, no. 2, pp. 237–264, 2007.
- [17] G. Traverso, D. Demirel, and J. Buchmann, "Dynamic and verifiable hierarchical secret sharing," in *Proceedings of the International Conference on Information Theoretic Security*, pp. 24–43, Springer, Tacoma, WA, USA, August 2016.
- [18] F. P. Mohamed and R. J. P. Arockia, "Hierarchical threshold secret sharing scheme for color images," *Multimedia Tools and Applications*, vol. 76, no. 4, pp. 5489–5503, 2017.
- [19] T. Bhattacharjee, S. P. Maity, and S. R. Islam, "Hierarchical secret image sharing scheme in compressed sensing," *Signal Processing: Image Communication*, vol. 61, pp. 21–32, 2018.
- [20] M. Tompa and H. Woll, "How to share a secret with cheaters," *Journal of Cryptology*, vol. 1, no. 3, pp. 133–138, 1989.
- [21] Y. Tian, C. Peng, Q. Jiang, and J. Ma, "Fair  $(t, n)$  threshold secret sharing scheme," *IET Information Security*, vol. 7, no. 2, pp. 106–112, 2013.
- [22] J. Halpern and V. Teague, "Rational secret sharing and multiparty computation," in *Proceedings of the Thirty-Sixth Annual ACM symposium on Theory of computing*, pp. 623–632, ACM, Chicago, IL, USA, June 2004.
- [23] L. Xiong and L. Liu, "Peertrust: supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.
- [24] E. Zhang, F. Li, B. Niu, and Y. Wang, "Server-aided private set intersection based on reputation," *Information Sciences*, vol. 387, pp. 180–194, 2017.
- [25] M. Nojournian and D. R. Stinson, "Socio-rational secret sharing as a new direction in rational cryptography," in *Proceedings of the International Conference on Decision and Game Theory for Security*, pp. 18–37, Springer, Budapest, Hungary, November 2012.
- [26] O. S. T. Litos and D. Zindros, "Trust is risk: a decentralized financial trust platform," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 340–356, Springer, Sliema, Malta, April 2017.
- [27] M. R. Clark, K. Stewart, and K. M. Hopkinson, "Dynamic, privacy-preserving decentralized reputation systems," *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2506–2517, 2017.
- [28] J. C. Benaloh, "Secret sharing homomorphisms: keeping shares of a secret," in *Proceedings of the Conference on the Theory and Application of Cryptographic Techniques*, pp. 251–260, Springer, Santa Barbara, CA, USA, August 1986.
- [29] S. Maleka, A. Shareef, and C. P. Rangan, "Rational secret sharing with repeated games," in *Proceedings of the International Conference on Information Security Practice and Experience*, pp. 334–346, Springer, Sydney, Australia, April 2008.
- [30] L. Harn, C. Lin, and Y. Li, "Fair secret reconstruction in  $(t, n)$  secret sharing," *Journal of Information Security and Applications*, vol. 23, pp. 1–7, 2015.

## Research Article

# Generative Reversible Data Hiding by Image-to-Image Translation via GANs

Zhuo Zhang <sup>1,2</sup>, Guangyuan Fu,<sup>1</sup> Fuqiang Di <sup>2</sup>, Changlong Li <sup>3</sup> and Jia Liu<sup>2</sup>

<sup>1</sup>*Xi'an Research Institute of High Technology, Xi'an 710086, China*

<sup>2</sup>*Key Lab of Networks and Information Security of PAP, Xi'an 710086, China*

<sup>3</sup>*The General Staff of PAP, Beijing 100000, China*

Correspondence should be addressed to Fuqiang Di; [wgd\\_dfq@sina.com](mailto:wgd_dfq@sina.com)

Received 25 April 2019; Accepted 2 July 2019; Published 11 September 2019

Guest Editor: Ki-Hyun Jung

Copyright © 2019 Zhuo Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The traditional reversible data hiding technique is based on cover image modification which inevitably leaves some traces of rewriting that can be more easily analyzed and attacked by the warder. Inspired by the cover synthesis steganography-based generative adversarial networks, in this paper, a novel generative reversible data hiding (GRDH) scheme by image translation is proposed. First, an image generator is used to obtain a realistic image, which is used as an input to the image-to-image translation model with CycleGAN. After image translation, a stego image with different semantic information will be obtained. The secret message and the original input image can be recovered separately by a well-trained message extractor and the inverse transform of the image translation. The experimental results have verified the effectiveness of the scheme.

## 1. Introduction

Information hiding [1–5], also called data hiding, is an important information security technique widely used in secret transmission [6], digital copyright protection [7], and other scenarios. If we classify data hiding by the reversibility of cover image, data hiding can generally be divided into two types: irreversible data hiding (IDH) [8, 9] and reversible data hiding (RDH) [10–12]. The traditional data hiding methods can be classified into the former type, while the latter type is a special technique which is mainly applied to medical, judicial, and military fields.

With the emergence and development of artificial intelligence [13–16] and other new techniques, the IDH methods using deep learning models have achieved a series of breakthroughs in methods and performance and become the trend of development in this field [17–20]. Among these new methods, secret data can be hidden and extracted well without any modification in the original cover image and cannot be detected by the warder (steganalysis algorithm). Comparatively, RDH with deep learning has received less attention. As far as our best knowledge, currently, no RDH

method can hide data without modification. At present, RDH methods can be divided into two types: RDH in unencrypted images [21–23] and RDH in encrypted images [24, 25]. Among these RDH methods, data hiding is based on cover image modification, which is more and more easily to be detected by increasingly advanced machine learning detection tools.

In [18], a new image steganography method via deep convolutional generative adversarial networks (DCGANs) is proposed. In this method, a mapping from the secret data to random noise is designed. With this mapping, a corresponding relationship between secret data and the stego image generated by the DCGAN model is obtained, and an extractor used to extract secret data is trained. This method has a strong ability to resist state-of-the-art detection tools, and it has provided great inspiration for the RDH method without modification.

Cycle-consistent generative adversarial network (CycleGAN) [26] is a newly proposed image-to-image translate model, which learns to automatically translate an image from a source domain into a target domain in the absence of paired examples. In this generative adversarial network (GAN) model,

there are two generators and two discriminators. Cycle-consistency loss is defined to train the CycleGAN model. Using CycleGAN, one type of picture can be transformed into another, and this transformation is reversible. Obviously, this kind of technique can be applied to the RDH field.

In [27], a framework for RDH in encrypted images based on reversible image transformation is proposed. At first, a cover image is transformed into another target image by image transformation. Then, secret data are embedded into the transformed image, which is regarded as the encrypted image. There have been many similar methods [28–30]. In this type of method, the image transformation is regarded as a special type of image encryption. And the embedding method belongs to the traditional method and essentially relies on pixel modification. However, the generative model uses neural networks to learn the data distribution rule of samples, and the generated image has strong randomness, which enhances the security of the data hiding algorithms. This advantage makes it far superior to traditional methods.

In this paper, the generative reversible data hiding (GRDH) method based on the GAN model is proposed. Learning the secret data mapping method in [18] and the image recovery method in CycleGAN, a new GRDH framework is proposed. In this framework, a cover image is generated by a noise vector, which is transformed by the secret data. Then, the cover image is transformed into a marked image by the CycleGAN model. Similar to the frame in [27], the transformed image can be regarded as a special encrypted image. In addition, a new extractor is trained to extract the secret data, which make the data hiding framework reversible. The experimental results have proved the feasibility of the proposed GRDH method.

## 2. DCGAN and CycleGAN

DCGAN [31] is an upgraded version of GAN. In the DCGAN model, a convolutional neural network is introduced to design a generator and discriminator. To improve the quality of generated samples and the speed of the convergence process, some changes to the structure of the original convolutional neural network have been made. With the powerful feature extraction ability of the convolutional neural network, the learning effect of the generative adversarial network has been significantly improved. The illustration of the DCGAN model is shown in Figure 1. The fake image is generated by random noise and a generator. The discriminator is designed in order to judge whether the generated image is a real image or fake image. The goal of the generator is to generate real images to deceive the discriminator. The goal of the discriminator is to separate the fake image from the real one as far as possible. In this way, the generator and discriminator constitute a dynamic game process.

CycleGAN [26] is essentially a ring network which is made up of two symmetrical GAN models. The illustration of the CycleGAN model is shown in Figure 2. On the one hand, two GAN models share two generators. On the other hand, each GAN model contains a discriminator. Thus, there are two generators and two discriminators in a CycleGAN model. Figure 3 shows the illustration of a one-way GAN model. Real

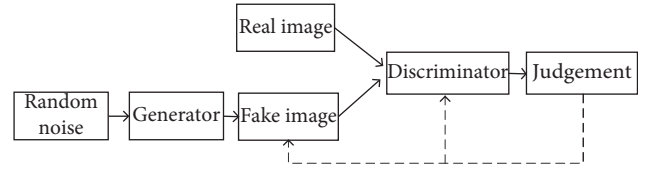


FIGURE 1: Illustration of the DCGAN model.

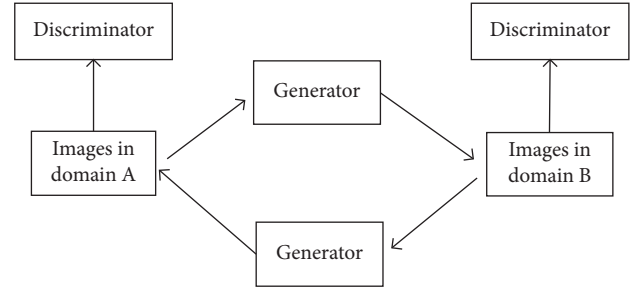


FIGURE 2: Illustration of the CycleGAN model.

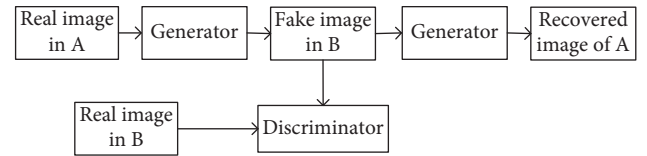


FIGURE 3: Illustration of a one-way GAN model in CycleGAN.

images in domain A can be transformed to fake images in domain B based on a discriminator and then transformed to the recovered image of domain A by another generator.

## 3. Generative Reversible Data Hiding

The illustration of GRDH is shown in Figure 4.

There are two processes in GRDH: preparation process and implementation process, consisting of the following 4 phases:

*Phase 1 (CycleGAN training).* A generator  $G_1$  and a restorer  $F$  are generated by the CycleGAN method. With two discriminators  $D_1$  and  $D_2$ , two image mapping goals are achieved:  $X \rightarrow Y$  and  $Y \rightarrow X$ , where  $X$  and  $Y$  are image collections.

*Phase 2 (generator training).* A generator  $G_2$  is obtained by a GAN method (e.g., DCGAN or BEGAN) with the help of a discriminator  $D_3$ .

*Phase 3 (extractor training).* In this phase based on the two discriminators  $G_1$  and  $G_2$  obtained earlier, we can achieve the transformation from random noise to image collection  $Y$ . Then, we train a new extractor  $E$  based on the GAN technique and ensure that the generated output  $Z_2$  is as close as possible to the input  $Z_1$ .

*Phase 4 (send and receive).* Before data hiding, the sender sends an extractor  $E$  and a restorer  $F$  to the receiver. Both



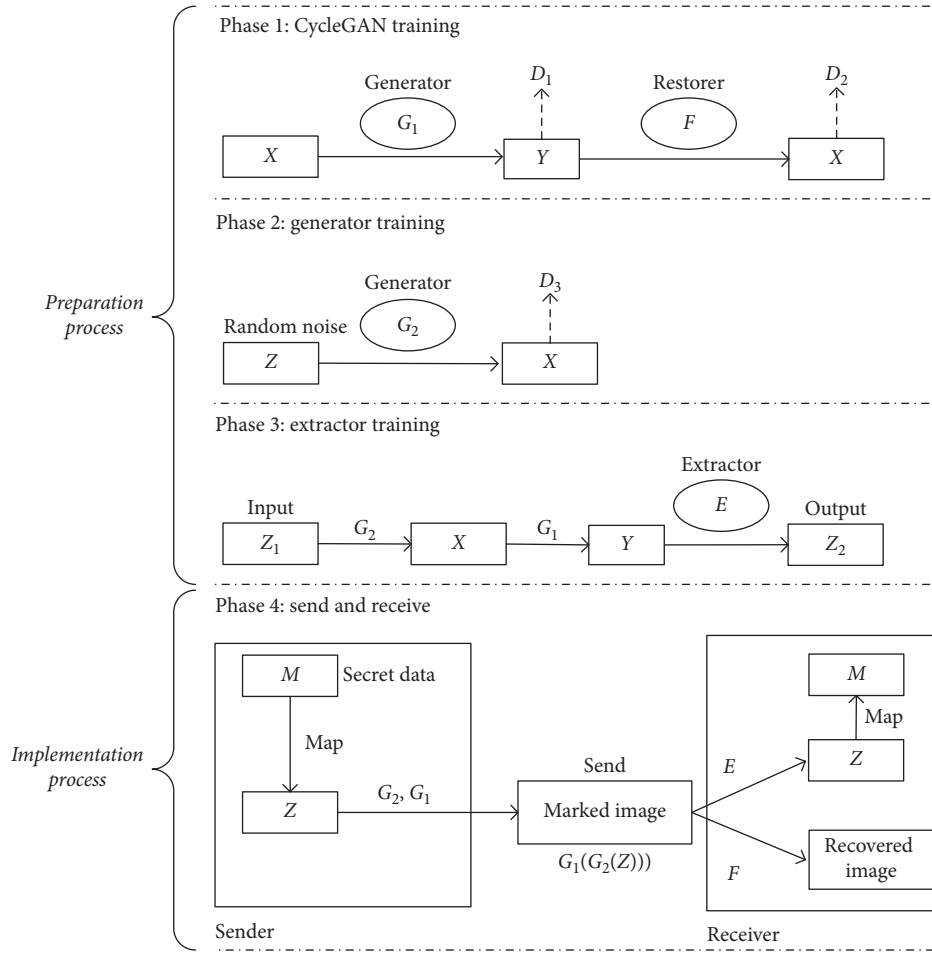


FIGURE 4: Illustration of the proposed GRDH method.

sides learn a mapping from secret data  $M$  to noise  $Z$ . Corresponding to the traditional RDH methods, the image generated by  $G_1$  and  $G_2$  can be regarded as the cover image and marked image. Then, the sender sends the marked image  $G_1(G_2(Z))$  to the receiver. At the receiver side, recovered image can be obtained and the embedded data can be extracted. We will go into detail about various phases in the following.

**3.1. Preparation Process.** The purpose of the preparation process is to train related models and prepare for data hiding, including three phases: CycleGAN training, generator training, and extractor training. The detailed steps can be described as follows:

**Phase 1 (CycleGAN training).** Our goal of this phase is to generate a marked image generator  $G_1$  and a recovered image restorer  $F$ . Without loss of generality, we choose the original CycleGAN model to train. Assume  $X$  and  $Y$  denote two image collections, corresponding to the cover image and marked image, respectively. Firstly, two image databases are built: a cover image database (CDB) and a marked image database (MDB). Each database contains one type of image. For example, the CDB contains images of normal horses and the MDB contains images of zebras. Then, the cover image in

the following phases will be an image of a normal horse, and the marked image to be sent to the receiver can be an image of a zebra. In Phase 1, the training process is based on the original CycleGAN model. We apply the two adversarial losses:  $L_{\text{GAN}}(G_1, D_1, X, Y)$  and  $L_{\text{GAN}}(F, D_2, Y, X)$ , for mapping  $X \rightarrow Y$  and  $Y \rightarrow X$  defined in [26], and the full objective function can be described as follows:

$$L(G_1, F, D_1, D_2) = L_{\text{GAN}}(G_1, D_1, X, Y) + L_{\text{GAN}}(F, D_2, Y, X) + \lambda L_{\text{cyc}}(G_1, F), \quad (1)$$

where  $L_{\text{cyc}}(G_1, F)$  denotes the cycle-consistency loss.

**Phase 2 (generator training).** Our goal of this phase is to generate a cover image generator  $G_2$ . Without loss of generality, we can choose the original DCGAN model to train. According to the principle of the DCGAN model, a generator  $G_2$  can be generated by the cover image database CDB and the discriminator  $D_3$ . Then, the mapping from random noise  $z$  to image collection  $X$  can be learned. The structures of the DCGAN model are introduced in [31]. Both the generator  $G_2$  and the discriminator  $D_3$  are CNN structures. Denote  $x$  and  $P_{\text{data}}$  as the real image and its

distribution from the cover image database CDB, and then the objective function to be optimized is as follows:

$$\min \max V(G_2, D_3) = E_{x \sim P_{\text{data}}(x)} [\log D_3(x)] + E_{Z \sim P_z(z)} \cdot [\log(1 - D_3(G_2(z)))] \quad (2)$$

Other unsupervised GAN models (e.g., BEGAN) can also be used for generator training.

*Phase 3 (extractor training).* After Phase 1 and Phase 2, two generators  $G_1$  and  $G_2$  have been trained. Based on these two generators and input noise  $z_1$ , the marked image can be generated by  $G_1(G_2(z_1))$ . Our goal of Phase 3 is to train an extractor  $E$  for secret data. We draw on Hu's method [18]. The construction method of  $E$  is similar to that of the discriminator in the DCGAN model, which has four convolutional layers and a fully connected layer. A leak-Relu activation function and batch normalization are used in each layer. Different from conventional CNN models, there is no pooling layer or dropout operation in the extractor. The illustration of the extractor in the GRDH method is shown in Figure 5.

If the input of  $E$  is the marked image with the size  $64 \times 64 \times 3$ , the output size after the first layer is  $32 \times 32 \times 64$ . In the following layers, the image dimensions are halved, and the number of channels is doubled from that in the previous layer. The final output is a noise vector of 100 dimensions. Each noise value in the vector is between  $-1$  and  $1$ . The defined loss function for extractor training can be described as follows:

$$L(E) = \sum_{i=1}^n (z_i - E(G_1(G_2(z_i))))^2 \quad (3)$$

We use this loss function to train the extractor as much as possible so that its output is as close as possible to the input noise  $z_1$ .

*3.2. Implementation Process.* After the preparation process, two generators  $G_1$  and  $G_2$ , an extractor  $E$ , and a restorer  $F$  are obtained. As shown in Figure 4, the sender holds  $G_1$  and  $G_2$  and sends  $E$  and  $F$  in advance by a secure channel. The above process is similar to that of key distribution in public key cryptography.

According to the steps of the preparation process, the noise vector  $Z$  is transformed into an image  $G_2(Z)$  by  $G_2$  at first and then transformed into another image  $G_1(G_2(Z))$  by  $G_1$ . From the view of the RDH technique, the first image  $G_2(Z)$  can be seen as the cover image, and the second image  $G_1(G_2(Z))$  can be seen as the marked image. In the implementation process, the only thing that the image owner needs to do is sending the marked image  $G_1(G_2(Z))$  to the receiver. At the receiving end, the receiver can recover the cover image by the restorer  $F$  and extract the noise vector by the extractor  $E$ . From the view of RDH in encrypted images, the above process at the receiving end belongs to a separable

scheme. It means that the receiver can not only recover the image before data extraction but also recover the image after data extraction. Beyond that, the mapping method proposed in [18] is used to realize the mapping from the secret binary bits to the noise vector. The mapping method can be described as follows.

At first, the secret binary bits are divided into several groups. Each group contains  $k$  bits. For example,  $\{110101100\}$  is divided into the three groups  $\{110\}$ ,  $\{101\}$ , and  $\{100\}$  when  $k = 3$ . Then, each group is mapped to a random noise  $r$  with a given interval according to the following equation:

$$r = \text{random}\left(\frac{m}{2^{k-1}} - 1 + \delta, \frac{m+1}{2^{k-1}} - 1 - \delta\right), \quad (4)$$

where  $m$  denotes the decimal value of the group to be mapped and  $\delta$  denotes the gap between the divided intervals. For example,  $k = 3$  and  $\delta = 0.001$ . We map every three secret bits into a random noise with the value between  $-1$  and  $1$ . The mapping from the group to the interval is shown in Table 1. At last, all the mapped noise is packaged into a vector. The above mapping method allows a deviation tolerance in data extraction and ensures the extraction accuracy of the secret data during the implementation process. This mapping method will be shared by both the sender and the receiver. The sender maps the secret data to the noise vector, and the receiver maps the noise vector to the secret data.

## 4. Experimental Results

In this section, a group of experiments is conducted to verify the effectiveness of the GRDH method proposed in this paper. These experiments consist of two parts. First, we train the GAN models and the extractor for GRDH preparation. Then, we use these trained models for GRDH to verify the feasibility of the method. In all the experiments, we generate random bits (using `random.randint` function in NumPy) as secret information. All images in the datasets were resized to  $64 \times 64$  in advance for model training. All experimental results are obtained by the Lenovo graphic workstation ThinkStation P500 with NVIDIA GeForce GTX 1080 Ti GPU and 128 GB of memory.

### 4.1. Experimental for GRDH Preparation

*4.1.1. CycleGAN Model Training.* We select the image database of horses and zebras in [32] as training samples. In the CycleGAN training stage, we set the initial learning rate to 0.0002 and the least batch size to 100. Besides, the stochastic gradient descent (SGD) [33] is selected as the optimization algorithm in model training. Some visual experimental results of CycleGAN training are shown in Figure 6.

The first line stands for the mapping from the  $X$ -domain image (horse) into the  $Y$ -domain image (zebra). The images from left to right are, respectively, the original image, the transformed image, and the reconstructed image. The second line stands for the mapping from the  $Y$ -

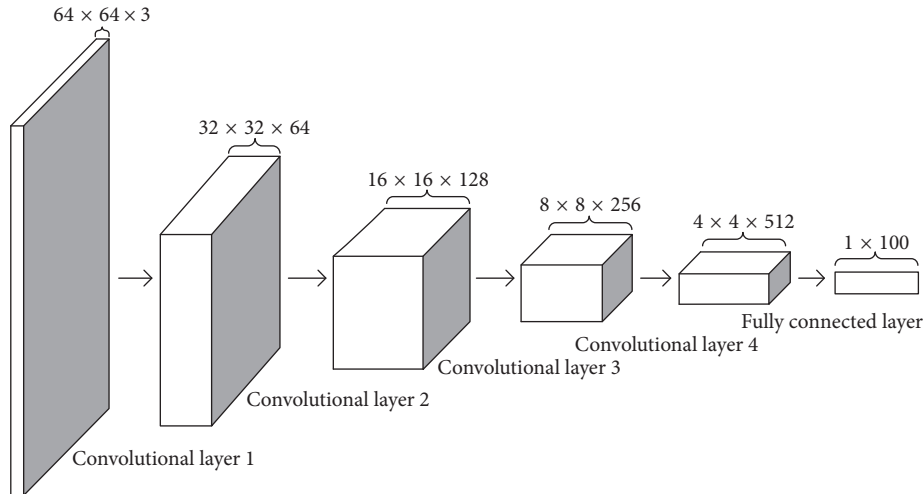


FIGURE 5: Illustration of the extractor in the GRDH method.

TABLE 1: Mapping from the group to the interval.

Group	Interval
000	(−0.999, −0.751)
001	(−0.749, −0.501)
010	(−0.499, −0.251)
011	(−0.249, −0.001)
100	(0.001, 0.249)
101	(0.251, 0.499)
110	(0.501, 0.749)
111	(0.751, 0.999)

domain image (zebra) into the  $X$ -domain image (horse). From Figure 6, it can be seen that when the model is trained for 210000 steps, the visual results of image transformation and image recovery can be acceptable in some special scenes when the demand for reversibility is not high. The visual results are related to the chosen image database and sample size.

We also use the man2woman image set [34] to train the model. The batch size is set to 100, and the random number seed is set to 1234; the initial learning rate value is set to 0.002, and the rate remains the same in the first 10,000 steps and then decays every 10,000 steps until it decays to zero (one step represents a batch of image training). Assuming that the Man image set is  $X$  and the Woman image set is  $Y$ , the adjustment parameters are set to 10.0. In training in both directions  $X \rightarrow Y$  and  $Y \rightarrow X$ , the first-moment parameter of the gradient descent optimizer is set to 0.5 and the number of filters of the first convolutional layer is set to 64. Figure 7 shows the image quality using the CycleGAN model after different training steps. The CycleGAN model is trained for 600,000 steps and takes about 52 hours, that is, about 11,500 steps per hour. It can be seen from the figure that the quality of the gender-converted image after the number of trainings more than 100,000 has reached an acceptable level. As the number of trainings continues to increase, the image quality of the CycleGAN model for gender conversion and image restoration is gradually improved.

**4.1.2. Generator Training.** Because of the high image quality of BEGAN's production, we directly use the BEGAN model for generator training. We use the CelebA image library [35]; the batch size is set to 16; the initial base learning rate is set to 0.001, and the learning rate is gradually attenuated. Furthermore, the initial value of the parameter  $k_0 = 0$ , and the initial value of the parameter  $\gamma = 2$ . The quality of the image generated by the BEGAN model after different training steps is shown in Figure 8. It can be seen from the figure that the quality of the generated image gradually increases with the increase of the number of training steps, and the number of steps is gradually increased to a more realistic level with the natural image.

It is worth noting that one of the important factors affecting the quality of the generated image is the type of the GAN model selected. The BEGAN model is relatively simple, the training time is low, and the image quality is relatively acceptable. Compared with the BEGAN model, although the training of the StyleGAN model takes a lot of time, the generated image is more realistic and closer to the natural image. Figure 9 shows a partial generation of the StyleGAN model based on the FFHQ image library. It can be seen that it is very close to the natural image, but even if the GPU of the NVIDIA Tesla V100 model needs to be trained for about five weeks.

**4.1.3. Extractor Training.** First, we generate 10,000 marked images by the generator from the trained BEGAN (random noise as the input) and CycleGAN. Then, we used these 10,000 images as training sets to train the extractor with a large number of random noise vectors. In the training procedure, the minibatch is set to 100, Adam optimization is used for training, and the learning rate is set to 0.0002. We train the extractor for 200,000 steps. The loss function value of the extractor is shown in Figure 10.

From the experimental results, it can be seen that the extractor converges rapidly and will eventually converge to a small value, which means that the output of the extractor is very close to the input noise vector. This means that we can



FIGURE 6: Partial experimental results of CycleGAN training. (a) Step = 200. (b) Step = 86000. (c) Step = 160000. (d) Step = 210000.



FIGURE 7: Continued.



FIGURE 7: CycleGAN model training results. (a) Step = 5000. (b) Step = 200000. (c) Step = 400000. (d) Step = 600000.

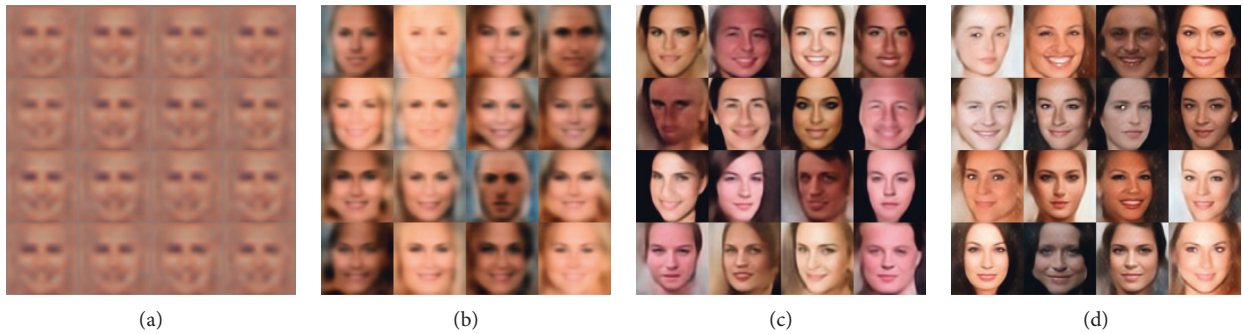


FIGURE 8: BEGAN model training results. (a) Step = 2000. (b) Step = 20000. (c) Step = 80000. (d) Step = 400000.



FIGURE 9: StyleGAN model training results.

reverse map the output of the extractor back to the secret message using data in Table 1.

*4.2. Experiments for GRDH Implementation.* First, the secret information bit string (generated using `random.randint` in NumPy) is divided into several segments (3 bits/segment, i.e.,  $k = 3$ ), and then the binary bit form segment is converted into a random noise form of the range  $[-1, 1]$  by formula (4). Image generation, image transformation, and

image restoration were performed using the trained CycleGAN model and BEGAN model. Among them, CycleGAN model training image sets are the Man image set and Woman image set, and the training frequency is 600,000 steps; the BEGAN model training image set is CelebA image library, and the training frequency is 400,000 steps; the noise dimension is set to 100, so the amount of embedded data in the experiment was set to 300 bits (i.e.,  $100 \times 3$ ). The experimental results of image generation, image transformation, and image restoration

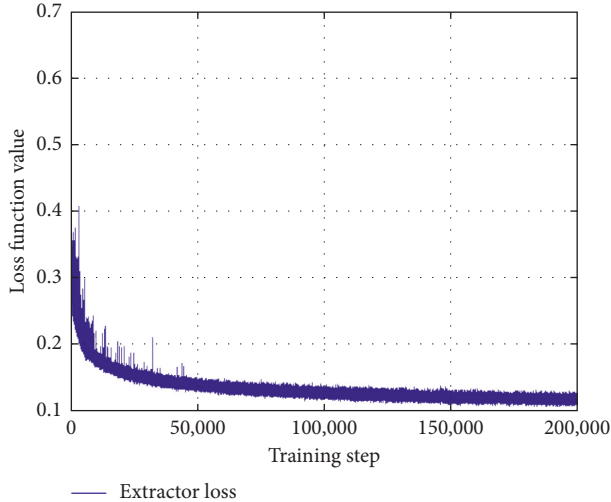


FIGURE 10: Loss function value of the extractor in training.

in some samples are shown in Figure 11(a). Because of the limitations of image quality generated by BEGAN, the visual quality of gender conversion and image restoration is relatively low. Figure 11(b) shows some experimental results when the StyleGAN model is adopted in the image generation process.

In addition, we calculated the average value of peak signal-to-noise ratio (PSNR) for 1,000 recovered images from two GAN models separately, as shown in Table 2.

Although the average PSNR value of restored images is not high, the subjective visual effect is acceptable. Meanwhile, it is easy to see from the figure that the visual quality of gender conversion and image restoration is relatively high. Therefore, whether in the preparation phase or the implementation phase, the image visual quality is mainly affected by the type of the selected GAN model.

To test the extractor, we measured the accuracy of the extraction of secret information from 1,000 marked images. The average extraction accuracy of the extractor is 88.7%. In addition, to test the effects of parameters  $k$  and  $\delta$  on the extractor recovery accuracy rate  $R$ , some further experiments are carried out. The effects of parameters  $k$  and  $\delta$  on recovery accuracy  $R$  are shown in Tables 3 and 4, respectively.

From the tables, it can be seen the recovery accuracy  $R$  significantly decreases with the rise of parameter  $k$  and slightly increases with the rise of parameter  $\delta$ . It is just because the smaller the parameter  $k$  is, the better the correction capability of the algorithm will be. Although the extractor recovery accuracy is not perfect, this problem can be resolved by including error-correction codes in the input noise.

Because the generative steganography is to fully fit the data distribution in the natural image dataset through the training generator, this technique is very resistant to the machine learning-based steganographic analyzer, which was mentioned in [18]. In other words, it is safer than traditional modification-based steganography. Besides, because of the



(a)



(b)

FIGURE 11: Results of image generation, image conversion, and image restoration. (a) BEGAN. (b) StyleGAN.

TABLE 2: Average values of PSNRs of recovered images.

GAN model	PSNR value (dB)
BEGAN	22.665
StyleGAN	28.333

TABLE 3: Effects of parameter  $k$  on recovery accuracy  $R$  when  $\delta = 0.01$ .

$k$ value	1	2	3	4	5
$R$ value	0.951	0.937	0.891	0.765	0.657

data hiding principle, the embedding capacity is very limited now. In the future, with the progress of the GAN model, the steganographic capacity of this scheme will gradually increase.

## 5. Conclusions

In this paper, a novel RDH scheme named “GRDH” based on the GAN model is proposed. Firstly, we use the GAN model to train a powerful image generator to get realistic images. This image is imported into the CycleGAN model to obtain images with different semantic information. In order

TABLE 4: Effects of parameter  $\delta$  on recovery accuracy  $R$  when  $k = 3$ .

$\delta$ value	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
$R$ value	0.883	0.889	0.894	0.887	0.889	0.893	0.897	0.886	0.894	0.898

to achieve message embedding, we then establish a mapping relationship between noise and messages. The extraction of the message is achieved by training an extractor to remove noise from the final dense image. The experimental results have demonstrated the effectiveness of the proposed method. Although the 100 percent reversibility cannot be achieved because of the existing performance of CycleGAN, the proposed method can achieve the first RDH scheme without cover modification. However, compared with those of the traditional methods, the embedding capacity of the proposed method is very limited, but the security is higher. In the future work, we will try to experiment with some new generative models to improve the quality of restored images and to improve the steganographic capacity of the method.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (nos. 61379152, 61403417, 61402530, and 61872384) and Shaanxi Provincial Natural Science Foundation (2014JQ8301).

## References

- [1] P. Moulin and J. A. O'Sullivan, "Information-theoretic analysis of information hiding," in *Proceedings of the 2000 IEEE International Symposium on Information Theory (Cat. No. 00CH37060)*, vol. 49, no. 3, pp. 563–593, Sorrento, Italy, August 2000.
- [2] Y. Ke, J. Liu, M.-Q. Zhang, T.-T. Su, and X.-Y. Yang, "Steganography security: principle and practice," *IEEE Access*, vol. 6, pp. 73009–73022, 2018.
- [3] A. Yassine, N. Joglekar, B. Dan, S. Eppinger, and D. Whitney, "Information hiding in product development: the design churn effect," *Research in Engineering Design*, vol. 14, no. 3, pp. 145–161, 2003.
- [4] M. Hussain, A. W. A. Wahab, N. Javed, and K.-H. Jung, "Recursive information hiding scheme through LSB, PVD shift, and MPE," *IETE Technical Review*, vol. 35, no. 1, pp. 53–63, 2018.
- [5] W. Mazurczyk and S. Wendzel, "Information hiding: challenges for forensic experts," *Communications of the ACM*, vol. 61, no. 1, pp. 86–94, 2017.
- [6] H. Boche, M. Cai, C. Deppe, and J. Nötzel, "Classical-quantum arbitrarily varying wiretap channel: secret message transmission under jamming attacks," *Journal of Mathematical Physics*, vol. 58, no. 10, article 102203, 2017.
- [7] B. P. Devi, K. M. Singh, and S. Roy, "New copyright protection scheme for digital images based on visual cryptography," *IETE Journal of Research*, vol. 63, no. 6, pp. 870–880, 2017.
- [8] T. Sarkar and S. Sanyal, "Reversible and irreversible data hiding technique," 2014, <https://arxiv.org/abs/1405.2684>.
- [9] P. Praveenkumar, N. K. Devi, K. Thenmozhi, J. B. B. Rayappan, and R. Amirtha, "Yet another but an incremental reversible data hiding- for an Irreversible data safety," in *Proceedings of the International Conference on Computer Communication & Informatics*, Coimbatore, India, January 2016.
- [10] Y.-Q. Shi, X. L. Li, X. P. Zhang, H.-T. Wu, and B. Ma, "Reversible data hiding: advances in the past two decades," *IEEE Access*, vol. 4, pp. 3210–3237, 2016.
- [11] F. Di, F. Huang, M. Zhang, L. Jia, and X. Yang, "Reversible data hiding in encrypted images with high capacity by bit-plane operations and adaptive embedding," *Multimedia Tools and Applications*, vol. 77, no. 16, pp. 20917–20935, 2018.
- [12] B. Ma and Y. Q. Shi, "A reversible data hiding scheme based on code division multiplexing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 1914–1927, 2016.
- [13] G. Pigozzi, A. Tsoukiàs, and P. Viappiani, "Preferences in artificial intelligence," *Annals of Mathematics & Artificial Intelligence*, vol. 20, no. 3, pp. 1–41, 2016.
- [14] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, "Neuroscience-inspired artificial intelligence," *Neuron*, vol. 95, no. 2, pp. 245–258, 2017.
- [15] A. Bundy, "Preparing for the future of artificial intelligence," *AI & Society*, vol. 32, no. 2, pp. 285–287, 2017.
- [16] M. Moravcik, M. Schmid, N. Burch et al., "DeepStack: expert-level artificial intelligence in heads-up no-limit poker," *Science*, vol. 356, no. 6337, pp. 508–513, 2017.
- [17] Y. Ke, M. Zhang, J. Liu, T. Su, and X. Yang, "Generative steganography with Kerckhoffs' principle based on generative adversarial networks," 2018, <https://arxiv.org/abs/1711.04916>.
- [18] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li, "A novel image steganography method via deep convolutional generative adversarial networks," *IEEE Access*, no. 6, pp. 38303–38314, 2018.
- [19] M. Liu, M. Zhang, J. Liu et al., "Coverless information hiding based on generative adversarial networks," *Journal of Applied Sciences*, vol. 36, no. 2, pp. 371–382, 2018.
- [20] C. Chu, A. Zhmoginov, and M. Sandler, "CycleGAN, a master of steganography," 2017, <https://arxiv.org/abs/1712.02950>.
- [21] C.-C. Lin, X.-L. Liu, W.-L. Tai, and S.-M. Yuan, "A novel reversible data hiding scheme based on ambtc compression technique," *Multimedia Tools and Applications*, vol. 74, no. 11, pp. 3823–3842, 2015.
- [22] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.
- [23] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.
- [24] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 553–562, 2013.

- [25] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 7, no. 2, pp. 826–832, 2012.
- [26] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, December 2017.
- [27] W. Zhang, H. Wang, D. Hou, and N. Yu, "Reversible data hiding in encrypted images by reversible image transformation," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1469–1479, 2016.
- [28] Y.-L. Lee and W.-H. Tsai, "A new secure image transmission technique via secret-fragment-visible mosaic images by nearly reversible color transformations," *IEEE Transactions Circuits and Systems for Video Technology*, vol. 24, no. 4, pp. 695–703, 2014.
- [29] D. Hou, W. Zhang, and N. Yu, "Image camouflage by reversible image transformation," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 225–236, 2016.
- [30] D. Hou, C. Qin, W. Zhang, and N. Yu, "Reversible visual transformation via exploring the correlations within color images," *Journal of Visual Communication and Image Representation*, vol. 53, pp. 134–145, 2018.
- [31] R. Alec, M. Luke, and C. Soumith, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2016, <https://arxiv.org/abs/1511.06434>.
- [32] <https://pan.baidu.com/s/1Zf6hvoDMsMi51WIPEOoqzg>.
- [33] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, <http://arxiv.org/abs/1412.6980>.
- [34] <https://pan.baidu.com/s/1i5qY3yt>.
- [35] Large-Scale CelebFaces Attributes (CelebA) Dataset, <http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>.



## Research Article

# Linear $(t, n)$ Secret Sharing Scheme with Reduced Number of Polynomials

Kenan Kingsley Phiri<sup>1</sup> and Hyunsung Kim <sup>1,2</sup>

<sup>1</sup>Department of Mathematical Sciences, University of Malawi, P.O. Box 280, Zomba, Malawi

<sup>2</sup>Department of Cyber Security, Kyungil University, Kyongsan, Kyungbuk 38428, Republic of Korea

Correspondence should be addressed to Hyunsung Kim; kim@kiu.ac.kr

Received 18 March 2019; Revised 13 June 2019; Accepted 16 July 2019; Published 4 August 2019

Guest Editor: Mehdi Hussain

Copyright © 2019 Kenan Kingsley Phiri and Hyunsung Kim. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Threshold secret sharing is concerned with the splitting of a secret into  $n$  shares and distributing them to some persons without revealing its information. Any  $t \leq n$  persons possessing the shares have the ability to reconstruct the secret, but any persons less than  $t$  cannot do the reconstruction. Linear secret sharing scheme is an important branch of secret sharing. The purpose of this paper is to propose a new polynomial based linear  $(t, n)$  secret sharing scheme, which is based on Shamir's secret sharing scheme and ElGamal cryptosystem. Firstly, we withdraw some required properties of secret sharing scheme after reviewing the related schemes and ElGamal cryptosystem. The designed scheme provides the properties of security for the secret, recoverability of the secret, privacy of the secret, and cheating detection of the forged shares. It has half computation overhead than the previous linear scheme.

## 1. Introduction

Information security has been a major concern over the past years in communication technology. The main concern has been how to make information confidential, authenticating and protecting it from being altered before reaching the receiver. Cryptography is part of the answer to these concerns [1]. The idea in cryptography is to prevent unauthorized use or alteration of information using mathematical tools. In a symmetric cryptosystem, this is achieved by using a shared secret key and in asymmetric cryptosystem it is achieved by using a pair of keys, public and private. The use of secret key or public and private key pair raised another problem of securely storing it. To address this problem, secret sharing schemes allow reliable storage without any risk [2, 3].

Threshold secret sharing is a method of splitting a secret  $s$  into  $n$  shares and distributing them to users such that the shares do not reveal any information about the secret [4]. In this case, the secret is a secret key, which needs to be stored securely. The secret is reconstructed easily if the authorized number of users combines their shares together. A subset of unauthorized users cannot reconstruct the secret or gain any

information about the secret. The shares are sent to users using private channels so that each user should not have information of shares of the other users before reconstruction is done. Schemes that achieve this are called secret sharing schemes. Some schemes reveal the shares of all users who take part in secret reconstruction. In such schemes, users know all shares after the secret is reconstructed. This is called open reconstruction. While other schemes do not reveal the shares even after reconstruction is done for the reason that the shares may be reused. This is called closed reconstruction. Such schemes use a trusted third party to take the role of secret reconstruction as discussed by Martin [5].

Shamir introduced a  $(t, n)$  threshold secret sharing scheme, which provides secure way of sharing a secret [6]. The scheme starts with secret information, which is divided into  $n$  pieces of information called shares. The shares are distributed by a dealer to  $n$  individuals, called users; each user gets at least a share. These shares do not reveal any information about the secret. The dealer is entrusted with sharing of the secret to  $n$  users using share generation and distribution algorithms. No user knows the share of the others because distribution is done through secure channel. The

secret is reconstructed by any authorized subset of users with cardinality  $t$  using Lagrange interpolation.

Tompa and Woll exposed the weakness of Shamir's scheme by introducing a cheating concept [7]. Malicious users present forged shares during reconstruction so that the honest users get an invalid secret. The cheaters will be able to reconstruct a valid secret since they know all correct shares. Therefore, Shamir's scheme cannot withstand this attack even if there is only one cheater. Furthermore, they proposed an improved scheme which uses redundant shares to detect cheating so that malicious users are prevented. Redundant shares are extra shares used in reconstructing the secret other than the required threshold. Many schemes also solve the same problem of cheating detection [8–14].

Apart from cheating detection, there are schemes that identify cheaters in [15–20]. These schemes provide the method of identifying any forged share once it has been detected that cheating takes place. Identification is also good since it helps to recognize who the cheater is and can be removed from the system during the next sharing. Other schemes reconstruct the secret even though there are forged shares called robust secret sharing scheme (RSS) [21–23]. RSS prevents cheating by allowing the reconstruction of the correct secret even if some participants submit forged shares. However, the probability of recovering the secret in RSS depends on the number of forged shares submitted during reconstruction phase. Furthermore, some schemes verify shares and are called verifiable secret sharing schemes (VSS) [24, 25]. VSS also prevents cheating by verifying the shares received from the dealer. In VSS, users assume that the adversary may corrupt the dealer, as a result it is no longer trusted. Once user receives the share from the dealer, he (or she) shares it to other users and creates a check vector, which helps to identify cheaters. The user rejects the share if it does not agree to the check vector, otherwise accepts it.

To achieve cheating prevention, users are given a share of the secret plus additional information, which is used for cheating detection. This makes share size  $|v_i|$ ,  $\forall i \leq t - 1$ , to increase greatly as compared to the secret size. However, it is shown by Carpentier et al. that a lower bound for the problem should be given as  $|v_i| \geq |s| / \epsilon$ , where  $\epsilon$  is the cheating probability [26]. The lower bound is based on the assumption that  $t - 1$  cheaters somehow know the secret before they cheat a user  $U_i$ . This is called Carpentier, De Santis, and Vaccaro (CDV) assumption. However, Ogata and Kurosawa proposed a scheme that detects cheating based on the assumption that no cheating user knows the secret [27]. The scheme's share size reaches the lower bound of  $|v_i| \geq (|s| - 1) / \epsilon + 1$ . This is called Ogata, Kurosawa, and Stinson (OKS) assumption.

Linear secret sharing schemes have been studied because of their application in multiparty computation and function sharing [28, 29]. The schemes are able to detect cheating behavior of malicious users during reconstruction of the secret; hence, an honest user cannot be fooled. Liu et al.'s scheme could be applied if system needs to share more than one secret [28]. Cramer et al.'s scheme depends its security on the universal hash function, which means that it is not unconditionally secure, where Lin and Harn's scheme has been proved to be easily broken by a simple attack as pointed

out by Ghodosi [30, 31]. Liu et al.'s scheme uses two polynomials to detect cheating during secret reconstruction and reduce the share size given to a user. Use of two polynomials increases the number of computations the scheme undergoes. As a result, there is an increased computation overhead.

This paper proposes a new linear  $(t, n)$  threshold secret sharing scheme based on a polynomial called polynomial based linear scheme (PBLs), which optimizes the number of computation overhead while maintaining security and privacy concerns. The goals that achieve security and privacy concerns (SP) of PBLs are

- (i) SP1: provide security of the secret.
- (ii) SP2: provide recoverability of the secret once shared.
- (iii) SP3: provide privacy of the secret and shares.
- (iv) SP4: provide cheating detection feasibility not only for one cheater but also  $t - 1$  cheaters so that any malicious behavior could be detected.

The goal that achieves computation overhead (CO) of PBLs is

- (i) CO: reduce the number of polynomials used so that computational overhead is reduced as compared to Liu et al.'s scheme.

To achieve these goals, PBLs uses ElGamal cryptosystem and Shamir's scheme as core operations. ElGamal cryptosystem helps to design a basic scheme, which is the initialization of PBLs. The basic scheme aims at hiding the secret during share generation phase, which can be revealed during cheating detection. PBLs applies Shamir's secret sharing scheme to share the secret, which uses the polynomial  $f(x)$  such that the element hiding the secret becomes the coefficient of  $x$ . Therefore, reconstruction of the secret in PBLs uses Lagrange interpolation, which comes up with the polynomial  $f^l(x)$ . Revealing the secret helps to detect cheating. PBLs has an advantage over Liu et al.'s scheme in the computation overhead concern. Furthermore, PBLs provides cheating detection feasibility, which is not available in Shamir's scheme.

## 2. Preliminaries

This section provides some basic mathematical and cryptographic concepts, which are major tools in secret sharing schemes, and reviews related works. First of all, the definition of finite field is provided together with some properties [32–34]. Polynomials in a finite field and Lagrange interpolation are also discussed to give understanding on the concepts. ElGamal cryptosystem is briefly discussed because it is used in construction of basic scheme. After that, we review related works such as linear secret sharing schemes, access structure, and some previous schemes like Shamir's and Liu et al.'s schemes [6, 14].

*2.1. Basic Mathematical and Cryptographic Concepts.* This section gives an overview of some mathematical concepts, which are useful in secret sharing like finite field, polynomial

and Lagrange interpolation. For more details on finite field, refer to [32–34]. The section also gives an overview of ElGamal cryptosystem, which assists in construction of basic scheme.

### 2.1.1. Finite Field

**Definition 1.** A finite field  $\mathbb{F}$  is a finite set on which addition, subtraction, multiplication, and division are defined and the following axioms are satisfied.

- (1) Associative: for all  $a, b$ , and  $c$  in  $\mathbb{F}$ ,  $a + (b + c) = (a + b) + c$  and  $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ .
- (2) Commutative: for all  $a$  and  $b \in \mathbb{F}$ ,  $a + b = b + a$  and  $a \cdot b = b \cdot a$ .
- (3) Existence of identity: there exists elements  $e$  and  $e' \in \mathbb{F}$ , such that  $a + e = e + a = a$  and  $a \cdot e' = e' \cdot a = a$ .
- (4) Existence of inverse: for every element  $a \in \mathbb{F}$ , there exists an element  $-a$  such that  $a + (-a) = e$ . Similarly for every element  $a \in \mathbb{F}$ , there exists an element  $a^{-1} \in \mathbb{F}$  such that  $a \cdot a^{-1} = e'$ .
- (5) Distributive: for all  $a, b$ , and  $c$  in  $\mathbb{F}$ ,  $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ .

Note that an element  $-a$  is called additive inverse and another element is called multiplicative inverse. An element  $e$  is an additive identity and an element  $e'$  is multiplicative identity. In this paper, we take  $e = 0$  and  $e' = 1$ .

**Definition 2** (finite field of order  $p$ ). Let  $p$  be a prime. The set of integers  $\mathbb{Z}_p = \{0, 1, 2, \dots, p - 1\}$  with addition and multiplication performed modulo  $p$  is a finite field of order  $p$  and is denoted as  $\mathbb{F}_p = \mathbb{Z}_p$ .

**Proposition 3** (multiplicative inverse). Let  $p$  be a prime. Element  $a \in \mathbb{Z}_p$ ,  $a \neq 0$  has a multiplicative inverse  $b = a^{-1}$  such that  $a \cdot b \equiv 1 \pmod{p}$ .

**Definition 4** (group of units). Let  $p$  be a prime. A group  $\mathbb{F}_p^*$  is a set that contains nonzero elements and is called a group of units.

### 2.1.2. Polynomials over Finite Field

**Definition 5** (polynomial over  $\mathbb{F}_p$ ). Let  $\mathbb{F}_p$  be a field. Any expression

$$f(x) = \sum_{i=0}^t a_i x^i \quad a_i \in \mathbb{F}_p, \quad (1)$$

where  $t$  is an arbitrary positive integer which is called a polynomial over  $\mathbb{F}_p$ .

**Definition 6** (degree of polynomial  $f(x)$ ). Given a nonzero polynomial  $f(x) = \sum_{i=0}^t a_i x^i$ , where  $a_t \neq 0$ , the number  $t$  is said to be the degree of  $f(x)$  denoted as  $\deg f(x)$ .

**Definition 7** (equal polynomials). Let  $f(x) = \sum_{i=0}^t a_i x^i$  and  $f'(x) = \sum_{i=0}^m b_i x^i$ , where  $a_t \neq 0$  and  $b_m \neq 0$  are two polynomials of degrees  $t$  and  $m$ , respectively. The two polynomials are equal and write  $f(x) = f'(x)$ , if  $t = m$  and  $a_i = b_i$  for all  $i = \{0, 1, 2, \dots, t\}$ .

**Definition 8** (roots of a polynomial). An element  $\alpha \in \mathbb{F}$  is called a root of  $f(x)$  if  $f(\alpha) = 0$ .

**Proposition 9.** A polynomial  $f(x) = \sum_{i=0}^t a_i x^i$ ,  $a_i \in \mathbb{F}$  of degree  $t$  cannot have more than  $t$  roots in the field  $\mathbb{F}$ .

**2.1.3. Lagrange Interpolation.** This is a method of reconstructing a polynomial from given known points. The polynomial constructed by Lagrange interpolation is called Lagrange interpolation polynomial, which is unique. To reconstruct the polynomial of degree  $t$ ,  $t + 1$  values are required, i.e.,  $(\alpha_i, \beta_i)$  for all  $i = 0, 1, 2, \dots, t$  such that  $f(\alpha_i) = \beta_i$ .

**Proposition 10.** Let  $\alpha_i$  for all  $i = 0, 1, 2, \dots, t$  be distinct elements of  $\mathbb{F}$  and  $\beta_i$  for all  $i = 0, 1, 2, \dots, t$  be arbitrary elements of  $\mathbb{F}$ . There exists no more than one polynomial  $f(x)$  of degree at most  $t$  such that  $f(\alpha_i) = \beta_i$  for all  $i = 0, 1, 2, \dots, t$ .

**Theorem 11** (see [33]). Let  $\alpha_0, \alpha_1, \dots, \alpha_t$  be distinct elements of  $\mathbb{F}$  and  $\beta_0, \beta_1, \dots, \beta_t$  be arbitrary elements of  $\mathbb{F}$ . There exists a unique polynomial

$$f(x) = \sum_{i=1}^t \beta_i \frac{(x - \alpha_0) \dots (x - \alpha_{i-1})(x - \alpha_{i+1}) \dots (x - \alpha_t)}{(\alpha_i - \alpha_0) \dots (\alpha_i - \alpha_{i-1})(\alpha_i - \alpha_{i+1}) \dots (\alpha_i - \alpha_t)} \quad (2)$$

of degree at most  $t$  such that  $f(\alpha_i) = \beta_i$  for all  $i = 0, 1, 2, \dots, t$ .

*Proof.* We adopt the proof by Slinko [33]. The polynomial in Equation (2) was constructed as follows. First construct polynomials  $g_i(x)$  of degree  $t$  such that  $g_i(\alpha_i) = 1$  and  $g_i(\alpha_j) = 0$  for  $i \neq j$ . These polynomials are

$$g_i(x) = \frac{(x - \alpha_0) \dots (x - \alpha_{i-1})(x - \alpha_{i+1}) \dots (x - \alpha_t)}{(\alpha_i - \alpha_0) \dots (\alpha_i - \alpha_{i-1})(\alpha_i - \alpha_{i+1}) \dots (\alpha_i - \alpha_t)}. \quad (3)$$

Thus, the polynomials  $g_0(x), g_1(x), \dots, g_t(x)$  are constructed. Furthermore, we multiply by  $\beta_i$  for  $i = 0, 1, 2, \dots, t$  and obtain the polynomials  $\beta_0 g_0(x), \beta_1 g_1(x), \dots, \beta_t g_t(x)$ . Summing the polynomials  $\beta_i g_i(x)$  the desired polynomial  $f(x)$  is constructed as Equation (4).

$$f(x) = \sum_{i=0}^t \beta_i g_i(x) \quad (4)$$

We set  $f(\alpha_i) = \beta_i$  as required. This polynomial is unique by Proposition 10.  $\square$

**2.1.4. ElGamal Cryptosystem.** ElGamal cryptosystem is in a family of public key cryptography [35]. Public key cryptography uses public key and private key to encrypt and decrypt

messages, respectively, such that knowledge of private key makes decryption easy. Without knowing the private key, it is impossible to decrypt a message in acceptable time. Security of ElGamal is based on discrete logarithm problem (DLP). Therefore, an attacker has to solve DLP to decrypt an intercepted message on ElGamal cryptosystem.

*Definition 12* (finite field DLP (FFDLP)). Given a finite field  $\mathbb{F}_p$ , a primitive element  $g$  of  $\mathbb{F}_p$ , and a nonzero element  $b$  of  $\mathbb{F}_p$ , the FFDLP of  $b$  to base  $g$ , written as  $\log_g(b)$ , is determining the least nonnegative integer  $i$  such that  $d = g^i$ .

Three algorithms are used in ElGamal cryptosystem, which are key generation, encryption, and decryption. We assume Alice and Bob want to communicate over an insecure channel. They have to generate a public and private key pair for encryption and decryption as follows.

*Key Generation.* Bob will do the following steps:

- (i) Generate a large prime  $p$  and a generator  $g$  of a multiplicative group  $\mathbb{Z}_p^*$
- (ii) Select a random integer  $b \in \mathbb{Z}_p^*$  such that  $1 \leq b \leq p - 2$
- (iii) Compute  $Y \equiv g^b \pmod{p}$ .

The public key for Bob is  $(p, g, Y)$  and  $b$  is the private key. Bob publishes the public key so that if anyone wants to send an encrypted message to him, they can use it.  $Y$  is an element in  $\mathbb{Z}_p^*$ , which has a multiplicative inverse in the group.

When Alice wants to send a message  $M$  to Bob, she needs to use Bob's public key to encrypt the message. The following are the steps she takes:

*Encryption*

- (i) Encode the message  $M$  such that  $1 \leq M \leq p - 1$ .
- (ii) Select a random exponent  $k$ .
- (iii) Compute  $C_1 = g^k$  and  $C_2 = M \cdot Y^k$ .

The encrypted message sent to Bob is a pair  $(C_1, C_2)$ .

Once Bob receives the message, he uses his private key to decrypt it in the following way:

*Decryption*

- (i) Compute  $C_1^{-b} = g^{-bk}$
- (ii) Compute  $M = C_1^{-b} \cdot C_2 = g^{-bk} \cdot M \cdot g^{bk}$ .

The element  $C_1^{-b}$  is the multiplicative inverse of  $g^{bk}$ .

*2.2. Related Works.* This section reviews linear secret sharing schemes, access structure of secret sharing schemes, and some previous schemes like Shamir's and Liu et al.'s schemes [6, 14]. Furthermore, the section discusses the strong and weak properties of the reviewed schemes.

*2.2.1. Linear Secret Sharing Scheme.* Linear  $(t, n)$  secret sharing scheme is a special type of secret sharing scheme where all the  $n$  shares of the secret satisfy a linear relationship [6, 14]. The Definition 13 gives what linear secret sharing scheme is.

*Definition 13* (linear secret sharing scheme). A  $(t, n)$  secret sharing scheme is a linear secret sharing scheme when the  $n$  shares,  $v_1, v_2, \dots, v_n$  can be presented as in Equation (5)

$$(v_1, v_2, \dots, v_n) = (k_1, k_2, \dots, k_t) H, \quad (5)$$

where  $H$  is a public  $t \times n$  matrix whose any  $t \times t$  submatrix is not singular. The vector  $(k_1, k_2, \dots, k_t)$  is randomly chosen by the dealer.

According to Definition 13, we can see that Shamir's  $(t, n)$  secret sharing scheme is a linear scheme. Let

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{t-1}x^{t-1}. \quad (6)$$

The shares  $v_i = f(i)$ ,  $i = 1, 2, \dots, n$  can be presented as in Equation (7)

$$(v_1, v_2, \dots, v_n) = (a_0, a_1, a_2, \dots, a_{t-1}) H, \quad (7)$$

where  $h_{i,j} = j^{i-1}$  ( $h_{i,j}$  denotes the entry at  $i$ th row and  $j$ th column of matrix  $H$ ).

*2.2.2. Access Structure.* Assume that  $U$  is the set of users where  $U = \{U_1, U_2, \dots, U_n\}$  and  $D$  is the dealer who facilitates secret sharing. An access structure is defined as follows.

*Definition 14* (access structure). Let  $2^U$  be the power set of the set of all users  $U$ . The set  $\Gamma \subseteq 2^U$  of all authorized coalitions is called the access structure of the secret sharing scheme [33].

If a subset is in the access structure, all sets that contain that subset should also form part of the access structure. Let  $X$  and  $Y$  be subsets of  $\Gamma$  such that  $X \subseteq Y$ . An access structure  $\Gamma$  may be any subset of  $2^U$  such that

$$\begin{aligned} X &\in \Gamma \\ \text{and } X &\subseteq Y, \end{aligned} \quad (8)$$

then  $Y \in \Gamma$ .

The condition in Equation (8) attached to access structure is called monotone property, which shows that if a smaller subset can know the secret, then any other larger set containing the subset will know it too.

*Definition 15.* Let  $\Gamma \subseteq 2^U$  be an access structure. A coalition  $C \subseteq U$  is called minimal authorized coalition if it is authorized and any proper subset of  $C$  is not authorized [33].

For example, if  $T \subset C$  and  $C$  is the minimal authorized coalition, then  $T$  is not authorized because  $|T| < |C|$ . The assumption is that every user is in at least one minimal coalition and otherwise is not useful in reconstructing the secret. In a linear  $(t, n)$  secret sharing scheme, the minimal coalition has subsets with  $t$  users.

**2.2.3. Shamir's  $(t, n)$  Threshold Scheme.** Shamir proposed a  $(t, n)$  threshold scheme that splits a secret  $s \in S$  into  $n$  shares, which are distributed to  $n$  users [6]. Splitting is done by a dealer using an algorithm called share generation algorithm. The algorithm uses a polynomial  $f(x)$  of degree  $t - 1$  to generate and distribute shares. The secret is reconstructed based on interpolating a polynomial using Lagrange interpolation, which is reconstructed by  $t$  users. The users combine their shares to reconstruct a polynomial  $f'(x)$  of degree  $t$  using reconstruction algorithm. The algorithm inputs the user's identity  $i$  and their share  $v_i$ , which forms a point or an ordered pair  $(i, v_i)$  for all  $i = 1, 2, \dots, t$  and outputs the secret  $f'(0) = s$ . Shamir's scheme has the following important properties.

- (i) Share size is exactly equal to secret size.
- (ii) If a new player joins or leaves, it is easy to add or delete shares without affecting the other shares.
- (iii) It is easy to change the shares of the same secret just by changing the polynomial without breaching any security.
- (iv)  $t - 1$  users do not reveal any information about the secret.

However, Tompa and Woll discovered that the scheme cannot withstand cheating if there is an untrusted user during secret reconstruction [7]. As a result, Shamir's scheme faces the following challenges during secret reconstruction.

- (i) Any malicious user can present a forged share without being noticed.
- (ii) It is difficult to detect if the reconstructed secret is invalid.
- (iii) A malicious user, once is successful in cheating other users, will be able to reconstruct the valid secret.

**2.2.4. Cheating Prevention.** Cheating prevention in secret sharing became a great concern after Tompa and Woll introduced cheating concept. As a result, many schemes with cheating prevention are proposed where some detect cheating, others identify cheaters, and so on. Some of the categories of cheating prevention are as follows.

- (i) Cheating detection: schemes provide the method to detect any forged share submitted for secret reconstruction by malicious user [7, 12]. The assumption is that the dealer is trusted.
- (ii) Cheater identification: schemes provide the method to detect and identify any forged share presented for secret reconstruction by a malicious user [15, 17]. The assumption is that the dealer is trusted.
- (iii) Robust secret sharing: schemes assume the dealer is trusted. Schemes can reconstruct a correct secret even if there are a number of forged shares presented by untrusted user [22].
- (iv) Verifiable secret sharing: schemes assume that the dealer is not trusted. Each user verifies the shares if valid using verification algorithm before reconstruction is done [9, 25].

**2.2.5. Liu Et Al's Scheme.** Liu et al. proposed a linear threshold secret sharing scheme, which is capable of cheating detection with share size  $|v_i| \geq |s| / \epsilon$ , where  $\epsilon > 0$  is the probability of cheating [14]. Liu et al.'s scheme is a combination of two Shamir's schemes. Two polynomials are used to share a secret  $s$ . Cheating detection is done by finding a random element  $r \in \mathbb{Z}_p$  during secret reconstruction.

Liu et al.'s scheme adopts Shamir's scheme in sharing the secret. This means that most properties of Liu et al.'s scheme are similar to Shamir's scheme. However, there are some properties, which Shamir's scheme does not have. The properties include the following.

- (i) Share size given to each user is equal to or greater than the secret size, i.e.,  $|v_i| \geq |s| / \epsilon$ .
- (ii) Detect cheating whenever a forged share is presented during reconstruction.

However, the scheme uses two polynomials to achieve the property of cheating detection, which makes number of computations to double as compared to Shamir's scheme.

### 3. New Linear $(t, n)$ Secret Sharing Scheme

This section proposes a new linear  $(t, n)$  threshold secret sharing scheme called polynomial based linear scheme (PBLS), which is based on one polynomial to reduce computational overhead of Liu et al.'s scheme and improve security of Shamir's scheme in terms of cheating detection. PBLS is capable of cheating detection for any forged shares presented for secret reconstruction with the help of the coefficients of  $x$  in the polynomial  $f'(x)$ . The coefficients are determined by a basic scheme which is an initialization of PBLS that adopts its properties from ElGamal cryptosystem. The security of PBLS is based on Shamir's scheme and ElGamal cryptosystem. PBLS provides perfect secret sharing, which is a required feature in all secret sharing schemes. The designed properties of PBLS withdrawn from the previous schemes satisfy SP1, SP2, SP3, and SP4, and computation overhead concern of CO.

**3.1. Fundamental Properties.** Basic scheme and PBLS adopt their properties from already existing scheme of ElGamal and Shamir. This makes the security of basic scheme and PBLS similar to the security of Shamir's scheme and ElGamal cryptosystem.

**3.1.1. Properties of Basic Scheme.** Basic scheme adopts its properties from ElGamal cryptosystem, which uses finite field elements to hide information [35]. The aim of basic scheme is to hide a secret in share generation phase but can be revealed when cheating detection is taking place. Secret hiding is done by multiplying a random element  $r$  by the secret  $s \in S$  to produce  $z$  for all  $r, s$ , and  $z \in \mathbb{F}_p$ . The secret is revealed when a multiplicative inverse of  $r$  is multiplied by  $z$ .

Security of ElGamal cryptosystem depends on the hardness of FFDLP. Therefore, basic scheme adopts the same security as ElGamal cryptosystem. Propositions 18 and 19 give the properties of basic scheme. However, to understand

these properties better, we first provide Definition 16 and Corollary 17 without proof. The proofs for definition and corollary can be obtained in [36].

*Definition 16.*  $a \equiv b \pmod{p}$  if and only if  $a$  and  $b$  leave the same remainder when divided by  $p$ .

**Corollary 17.** *The integer  $c$  is the remainder when  $a$  is divided by  $p$  if and only if  $a \equiv c \pmod{p}$ , where  $0 \leq c < p$ .*

The following are the properties of basic scheme.

**Proposition 18.** *Let  $s$  and  $r \in \mathbb{Z}_p$  be a secret and a random element, respectively, and  $z \equiv s \cdot r \pmod{p}$  such that  $p$  is prime. It has FFDLP difficulty to withdraw  $s$  and  $r$  from the element  $z \in \mathbb{Z}_p$ .*

*Proof.* Since elements  $s$  and  $r$  are field elements, the operation  $n = s \cdot r \equiv z \pmod{p}$  is a modulo multiplication. Thus  $z$  is a remainder when  $p$  divides the integer  $n$ . Assume that there exists only one integer  $n$ , which leaves a remainder  $z$  when  $p \mid n$ , then

$$n = t \cdot p + z \quad \forall t > 0. \quad (9)$$

By Corollary 17,  $n \equiv z \pmod{p}$ . Therefore,  $t$  is unique. Let  $d = a \cdot b$  such that  $a \neq s \neq r$  and  $b \neq s \neq r$ . By Definition 16,  $n \equiv d \pmod{p}$  if and only if  $n$  and  $d$  leave the same remainder when they are divided by  $p$ . Thus

$$\begin{aligned} n &= t \cdot p + z \\ d &= t \cdot p + z \\ \forall t &> 0. \end{aligned} \quad (10)$$

This implies that

$$\begin{aligned} n &= d \\ s \cdot r &= a \cdot b. \end{aligned} \quad (11)$$

But  $a \neq s \neq r$  and  $b \neq s \neq r$ . Therefore,  $t$  is not unique. This contradicts the fact that  $n$  is the only integer that leaves a remainder  $z$  when  $p \mid n$ . Therefore  $n \equiv d \pmod{p}$ , where  $d \neq n$ . Element  $z$  is a remainder whenever  $t \cdot p$  divides  $d$  such that  $d > t \cdot p$  and  $n$  such that  $n > t \cdot p$ . Integers  $d$  and  $n$  contain different factors since they are not equal hence difficult to determine  $s$  and  $r$  from  $z$ , which is FFDLP.  $\square$

**Proposition 19.** *Let  $z \equiv s \cdot r \pmod{p}$  such that  $s \in \mathbb{Z}_p$  is a secret,  $r \in \mathbb{Z}_p$  is a random element, and  $p$  is prime. It is impossible to determine  $s$  from  $z$ , which is based on the difficulty of the fractional decomposition.*

*Proof.* By Proposition 18, it is difficult to determine  $s$  and  $r$  from  $z$  because  $z$  does not reveal  $s$  and  $r$ . If we assume that we know the value of  $r$ , then it is possible to determine  $s$ . Since  $r$  is known, the multiplicative inverse of  $r$  can be computed from finite field  $\mathbb{F}_p$ . Multiplying  $z$  by  $r^{-1}$  gives  $s$  as follows:

$$z \cdot r^{-1} \equiv s \cdot r \cdot r^{-1} \pmod{p} \equiv s \pmod{p}. \quad (12)$$

Knowledge of  $r$  helps to determine  $s$ . Therefore, by contrapositive, we cannot determine  $s$  from  $z$ , which is based on the fractional decomposition difficulty.  $\square$

It is noted that though FFDLP is applied in basic scheme, there is a difference with ElGamal cryptosystem. The difference is that basic scheme makes no use of exponentiation, which helps it to operate in polynomial time.

**3.1.2. Properties of PBLs.** Any secret sharing scheme should be secure from malicious users by denying them the opportunity to obtain the secret when the required number of users is not reached. At the same time, the secret should be able to be reconstructed after sharing. PBLs adopt its properties from Shamir's secret sharing scheme, which shares a secret to  $n$  users to be recovered by  $t$  users where  $t \leq n$  using a polynomial of degree  $t - 1$ , where the coefficients of  $x^0$  and  $x$  are  $s$  and  $z$ , respectively. Therefore, all the properties for Shamir's scheme also hold for PBLs. However, PBLs also consist of some properties of ElGamal cryptosystem because of the use of basic scheme. Two fundamental properties of designing PBLs, which are adopted from Shamir's scheme, are given in Proposition 20 such that every secret sharing scheme has to be achieved.

**Proposition 20** (SP1 and SP3). *Let  $s$  be the secret and  $t$  the threshold. Any less than  $t$  users cannot know the secret.*

**Proposition 21** (SP2). *Let  $t$  be the threshold of a secret sharing scheme. Any  $t$  or more than  $t$  users should be able to reconstruct the secret by combining their shares together.*

Since Shamir's scheme is linear that is  $n$  shares of secret satisfy a linear relationship, PBLs are also linear. However, PBLs have a property of SP4 as in Proposition 22.

**Proposition 22** (SP4). *Let  $t$  be the threshold of a secret sharing scheme and there are any less than  $t$  forged shares used for secret reconstruction. The shares will be detected during secret reconstruction.*

Any secret sharing scheme, which prevents cheating, must give to each participant shares whose sizes are at least the size of the secret plus  $\log 1/\epsilon$ , where  $\epsilon$  is the probability of successful cheating [26]. The Proposition 23 gives a property of the share size of PBLs given to users.

**Proposition 23.** *Let  $v_i = \{f(i), y\}$  be the share given to each user. The share size of PBLs attains the bounds of  $|v_1| \geq |s|/\epsilon$ .*

Any secret sharing scheme has a set of users who are allowed to make reconstruction of the secret called the access structure based on Definitions 14 and 15. Proposition 24 provides an access structure of PBLs.

**Proposition 24.** *Let  $U = \{U_1, U_2, \dots, U_n\}$  be the set of users. The access structure of PBLs is a set  $\Gamma \subseteq 2^{[U]}$  such that  $X \subseteq \Gamma$ , where  $|X| \geq t$ .*

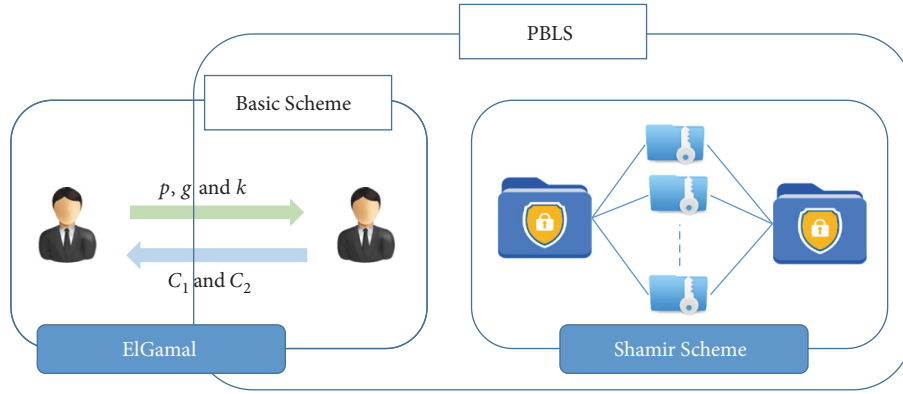


FIGURE 1: Adoption of ElGamal cryptosystem and Shamir's scheme in PBLs.

*Proof.* The scheme uses Lagrange interpolation to come up with a polynomial  $f(x)$  of degree  $t - 1$ . By Theorem 11,  $t$  points are required to interpolate this polynomial. Thus any subset of  $t$  or more than  $t$  shares is authorized to reconstruct the secret.  $\square$

PBLs is composed of basic scheme and Shamir's scheme. However, exponentiation is not used in PBLs to reduce computation cost. Figure 1 shows the properties of ElGamal cryptosystem and Shamir's secret sharing scheme and how they are adopted in basic scheme and PBLs.

**3.1.3. Adversary Model.** In any cryptographic application, an attacker  $A$  has different goals to achieve for the attack mode to it. Secret sharing schemes face cheating attack, which was discovered by Tompa and Woll. Despite different applications of secret sharing schemes, a malicious user presents forged shares during secret reconstruction. Therefore, the following are some goals of cheaters in secret sharing schemes try to achieve:

- (i) To recover the valid secret while the honest users are unable to detect cheating [29]. In this case, honest users believe the secret to be valid.
- (ii) To recover the secret while the honest users are able to detect cheating [37]. The honest users will not have access to this secret; hence, they simply assist  $A$  to reconstruct it without their knowledge.

There are two assumptions in which cheaters behave. These are OKS and CDV as discussed [26, 27]. CDV assumes that cheaters already know the secret to be reconstructed. They only aim at blocking the correct reconstruction of the secret while they already have the secret. Honest users will get the invalid secret. On the other hand, OKS assumes that the cheater does not know the secret to be reconstructed. The aim is to block the correct reconstruction of the secret, but at the end they should be able to get a valid secret. PBLs considers OKS assumption because the aim of secret sharing is that the secret should not be known before reconstruction.

Cheating becomes successful when cheaters managed to reconstruct a valid secret while honest users failed to detect that cheating takes place. PBLs makes sure that  $A$  does not

recover the secret whenever cheating detection is achieved. To prevent  $A$  from learning the secret, a closed reconstruction is done where no user can see the share of the other users. This also prevents malicious users who communicate their shares during reconstruction after learning the shares of honest user as described by [38]. Such users are called rushing cheaters.

**3.2. Proposed Schemes.** This subsection proposes basic scheme and PBLs. Basic scheme has two algorithms, which are secret hiding and secret revealing. The secret is hidden with field element in secret hiding algorithm. It is revealed using the multiplicative inverse of the element in secret revealing algorithm. PBLs has three algorithms, which are share generation, secret reconstruction, and cheating detection.

**3.2.1. Basic Scheme.** This subsection proposes basic scheme, which is the basis for constructing PBLs proposed in Section 3.2.2. Basic scheme provides a conceptual process of how PBLs detects cheating during secret reconstruction. A secret is hidden by a field element  $r$  and can be revealed by a multiplicative inverse  $b$  of the element  $r$ .

*Secret Hiding.* Consider a finite field  $\mathbb{F}_p$  in which  $p$  is a prime such that  $p - 1$  has at least one large prime factor. If  $p - 1$  has only small prime factors, then computing FFDLP is easy as pointed out by [5]. By Proposition 3, all nonzero elements  $a$  in  $\mathbb{F}_p$  have a multiplicative inverse  $b$  such that  $a \cdot b \equiv 1 \pmod p$ . A secret  $s$  is also the field element as  $s \in \mathbb{F}_p$ . Any random number  $r$  except 1 can be used to hide the secret  $s$ . The algorithm for hiding the secret avoids using 1 because it is a multiplicative identity therefore cannot hide  $s$ . After multiplying  $r$  by  $s$ , a different field element  $z$  is obtained. Hence Equation (13) follows

$$z \equiv r \cdot s \pmod p. \tag{13}$$

Algorithm 25 describes how a secret  $s$  is hidden using field element in basic scheme.

*Algorithm 25* (secret hiding).

*Input.* Secret  $s$

*Output.* Element  $z$

*Process*

- (i) Choose a random element  $r \in \mathbb{Z}_p$ .
- (ii) Compute  $z$  by multiplying  $r$  by  $s$ .
- (iii) Output element  $z \in \mathbb{Z}_p$ .

Figure 2 illustrates how Algorithm 25 works.

The element  $z$  does not reveal any information of  $s$  and  $r$  in basic scheme.

*Secret Revealing.* Whenever one wants the secret  $s$  back, he (or she) simply computes the multiplicative inverse of  $r$ , given as  $b$ , and multiplies it by  $z$  to get the secret  $s$ . Therefore, multiplying  $z$  by multiplicative inverse of  $r$  is the same as multiplying  $r$  by  $r^{-1}$  by  $s$ . This means 1 is multiplied by  $s$  to get a result  $s$  as in Equation (14)

$$s \equiv b \cdot z \pmod{p} \equiv r^{-1} \cdot r \cdot s \pmod{p} \equiv s \pmod{p}. \quad (14)$$

Algorithm 26 shows how to reveal the secret  $s$  using the multiplicative inverse of  $r$ .

*Algorithm 26* (secret revealing).

*Input.* Elements  $z$  and  $r$

*Output.* Secret  $s$

*Process*

- (i) Compute multiplicative inverse  $b$  of element  $r$ , i.e.,  $b = r^{-1}$ .
- (ii) Compute  $s$  by multiplying  $b$  by  $z$ .
- (iii) Output  $s$ .

Figure 3 illustrates Algorithm 26: how revealing the secret occurs.

Basic scheme requires  $r$  to be kept secret so that the secret remains private and secure. Otherwise, any adversary will be able to compute the inverse of  $r$  and reveal  $s$  as is done in Algorithm 26. We demonstrate this with a dummy example below.

*Example 27.* Let  $p = 23$  and  $s = 12$ . Secret  $s$  can be hidden as follows. Choose a random element  $r = 7 \in \mathbb{Z}_{23}$ . Compute  $z$  by multiplying  $r$  by  $s$  to obtain  $z$

$$z \equiv r \cdot s \pmod{23} \equiv 7 \cdot 12 \pmod{23} \equiv 15 \pmod{23}. \quad (15)$$

The secret 12 is hidden as 15. It is difficult for an adversary A to know the secret 12 and the random 7 from 15 alone unless he (or she) solves FFDLP. The secret  $s$  is revealed by computing  $b$ , the multiplicative inverse of  $r$

$$b = r^{-1} = 10, \quad (16)$$

and computing  $s$  by multiplying  $y$  by  $z$

$$s \equiv b \cdot z \pmod{23} \equiv 10 \cdot 15 \pmod{23} \equiv 12 \pmod{23}. \quad (17)$$

Note that in practice,  $p$  should be a large prime number for the scheme to be secure enough.

*3.2.2. Polynomial Based Linear Scheme (PBLs).* In this subsection, a linear  $(t, n)$  threshold secret sharing scheme, PBLs, is proposed, which provides cheating detection based on basic scheme and Shamir's secret sharing. We assume using two trusted third parties dealer  $D$  and combiner  $C$ .  $D$  generates and distributes shares to  $n$  users while  $C$  collects any  $t$  shares and reconstructs the secret. A trusted user can also be  $C$  depending on the application.  $C$  does not reveal shares after reconstruction and hence performs a closed reconstruction. In addition,  $C$  performs cheating detection in secret reconstruction phase. PBLs has three algorithms, which are share generation, secret reconstruction, and cheating detection.

*Share Generation.* As Shamir's scheme, share generation algorithm starts with  $D$  setting public parameters, prime  $p$  and threshold  $t$ .  $D$  chooses a random element  $r$  from a finite field  $\mathbb{F}_p$ , then hides the secret using Algorithm 25. The output is element  $z \in \mathbb{F}_p$ .  $D$  computes  $b \in \mathbb{F}_p$ , a multiplicative inverse of  $r$ . The element  $b$  is sent to  $n$  users on public channel while element  $z$  becomes a coefficient of  $x$  in polynomial  $f(x)$ . To share the secret  $s$ ,  $D$  chooses a random polynomial of degree  $t - 1$ , which has constraints of two coefficients,  $a_0 = s$  and  $a_1 = z$ .  $D$  computes  $f(i)$  for all  $i = 1, 2, \dots, n$  and distributes  $v_i = (i, f(i))$  to users where  $i = 1, 2, \dots, n$ . Algorithm 28 shows how shares are generated and distributed to users.

*Algorithm 28* (share generation).

*Input.* Secret  $s$

*Output.* Secret shares  $v_i$  where  $i = 1, 2, \dots, n$

*Process*

- (i)  $D$  uses Algorithm 25 to compute  $z$ .
- (ii)  $D$  uses Algorithm 26 to compute  $b$ .
- (iii)  $D$  chooses a random polynomial  $f(x)$  of degree  $t - 1$  over  $\mathbb{F}_p$ , i.e.,

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{t-1}x^{t-1} \quad (18)$$

such that  $a_0 = s$  and  $a_1 = z$ .

- (iv)  $D$  computes  $f(i)$  and distributes  $v_i = (i, f(i))$  and  $b$  to  $U_i$  for all  $i = 1, 2, \dots, n$  secretly.

Figure 4 illustrates the share generation algorithm in PBLs.

Users cannot obtain information of the secret  $s$  from  $z$  without the knowledge of  $r$  unless they have to solve FFDLP.

*Secret Reconstruction.* If the secret is required for use, any  $t \leq n$  users combine their shares together to reconstruct it.



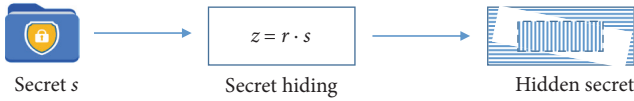


FIGURE 2: Secret hiding in basic scheme.

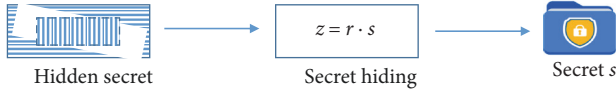


FIGURE 3: Secret revealing in basic scheme.

The combiner can be one of the user or a trusted third party who does the reconstruction without revealing the shares. Users send their shares to  $C$  together with  $b$ .  $C$  uses Lagrange interpolation to reconstruct the polynomial of degree  $t - 1$  from at least  $t$  points, i.e.,  $(1, f(1)), (2, f(2)), \dots, (i, f(i))$  and the secret is  $f(0)$  if all users are honest. Figure 5 illustrates how secret reconstruction is done in PBLs.

*Algorithm 29* (secret reconstruction).

*Input.* Any list of  $t$  shares

*Output.* Secret  $s' = s$  if there is no cheater or  $s' \neq s$  if cheaters exist

*Process*

- (i)  $C$  reconstructs  $f'(x)$  from  $(i, v_i)$  using Lagrange interpolation

$$f'(x) = \sum_{i=1}^t v_i \prod_{j=1, j \neq i}^t \frac{x_i - x}{x_i - x_j} \quad (19)$$

- (ii)  $C$  outputs polynomial  $f'(x) = a_0' + a_1'x + a_2'x^2 + \dots + a_{t-1}'x^{t-1}$ .

By Theorem 11, the reconstructed polynomial  $f'(x)$  is unique and if there is no cheating then polynomial  $f'(x)$  is equal to polynomial  $f(x)$ . Therefore, the secret is  $f'(0) = a_0' = s' = s$ .

*Cheating Detection.* It is important to check if there are forged shares presented during secret reconstruction. In this phase, PBLs uses basic scheme to reveal the secret since it is hidden in the coefficient of  $x$  of the polynomial  $f(x)$ . Consequently, the polynomial  $f'(x)$  should have the same coefficient. The secret is revealed by multiplying the coefficient of  $x$  by a multiplicative inverse of an element  $r$ . If the result gives term  $a_0'$ , then there is no cheating. Therefore, the secret is valid. Otherwise, some forged shares are used during reconstruction of the secret. Algorithm 30 shows how cheating detection is done.

*Algorithm 30* (cheating detection).

*Input.* Elements  $z$  and  $b$

*Output.* No cheating or cheating

*Process*

- (i)  $C$  computes  $b \cdot z = a_0'$ .
- (ii)  $C$  outputs no cheating if  $a_0'$  holds or cheating otherwise.

$C$  uses Algorithm 30 to detect if forged shares were presented to reconstruct the secret.  $C$  uses multiply  $a_1'$  by  $b$  to get  $a_0'$ . Once the output is not  $a_0'$ , then cheating took place, secret reconstruction halted, and the reconstructed secret was not valid.  $C$  sends to users a signal that secret reconstruction has failed. In cases where all users are honest, Algorithm 30 outputs no cheating. Consequently, the output Algorithm 29 valid and  $C$  sends  $a_0'$  to each user, which shows that secret reconstruct is successful. A simple example below demonstrates how the new scheme works.

*Example 31.* Let  $p = 23$ . Given the secret  $s = 12$ , we can share it to  $n = 6$  users such that any  $t = 4$  of them can reconstruct the secret.

We use Algorithm 25 to hide the secret and select a random  $r \in \mathbb{F}_{23}$  as 15

$$z \equiv s \cdot r \pmod{p} \equiv 12 \cdot 15 \pmod{23} \equiv 19. \quad (20)$$

We also compute a multiplicative inverse of  $r$ .

$$b \equiv r^{-1} \pmod{p} \equiv 15^{-1} \pmod{23} \equiv 20 \pmod{23}. \quad (21)$$

Let the random polynomial be

$$f(x) = 12 + 19x + 20x^2 + 9x^3. \quad (22)$$

The shares given to users are

$$\begin{aligned} U_1 : f(1) &= 12 + 19 \times 1 + 20 \times 1^2 + 9 \times 1^3 = 14 \\ U_2 : f(2) &= 12 + 19 \times 2 + 20 \times 2^2 + 9 \times 2^3 = 18 \\ U_3 : f(3) &= 12 + 19 \times 3 + 20 \times 3^2 + 9 \times 3^3 = 9 \\ U_4 : f(4) &= 12 + 19 \times 4 + 20 \times 4^2 + 9 \times 4^3 = 18 \\ U_5 : f(5) &= 12 + 19 \times 5 + 20 \times 5^2 + 9 \times 5^3 = 7 \\ U_6 : f(6) &= 12 + 19 \times 6 + 20 \times 6^2 + 9 \times 6^3 = 7. \end{aligned} \quad (23)$$

Each user also receives 20 a multiplicative inverse of 15  $\in \mathbb{F}_{23}$ .

When the secret is required, any 4 users send their shares to  $C$  to reconstruct the secret  $s$ . Let  $U_1, U_3, U_5,$  and  $U_6$  send their shares to  $C$ . We use Lagrange interpolation to

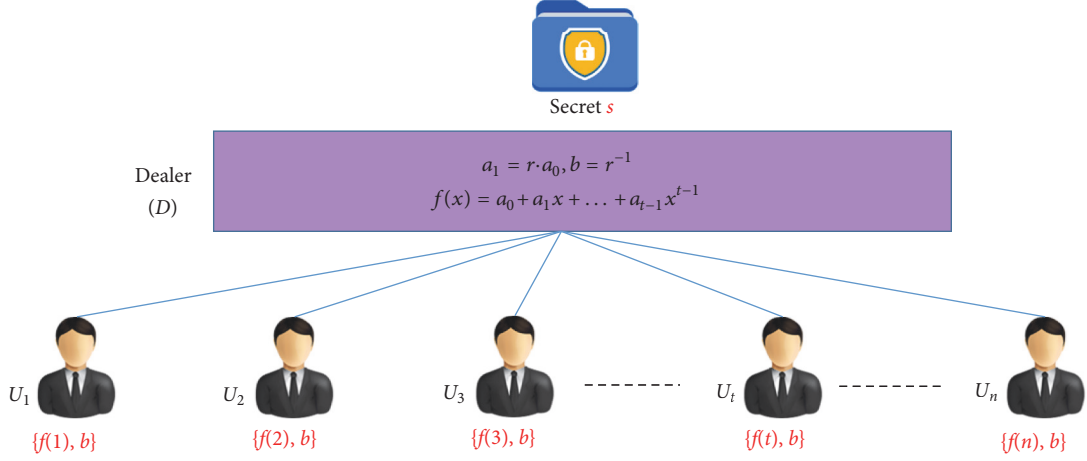


FIGURE 4: Share generation in PBLs.

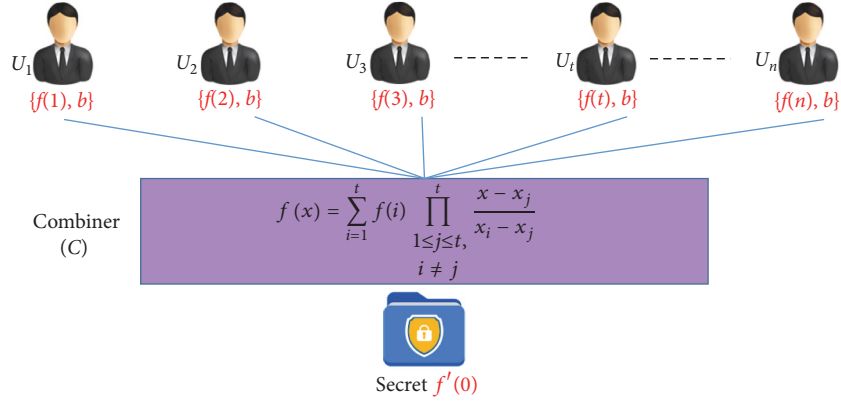


FIGURE 5: Secret reconstruction in PBLs.

reconstruct a polynomial  $f'(x)$ . We find the  $g_i(x) \forall i = \{0, 1, 2, 3\}$  as discussed in Section 2.1.3

$$g_0(x) = \frac{(x-3)(x-5)(x-6)}{(1-3)(1-5)(1-6)}$$

$$= 4(x^3 + 9x^2 + 17x + 2)$$

$$= 4x^3 + 13x^2 + 22x + 8$$

$$g_1(x) = \frac{(x-1)(x-5)(x-6)}{(3-1)(3-5)(3-6)}$$

$$= 2(x^3 + 11x^2 + 18x + 16)$$

$$= 2x^3 + 22x^2 + 13x + 9$$

$$g_2(x) = \frac{(x-1)(x-3)(x-6)}{(5-1)(5-3)(5-6)}$$

$$= 20(x^3 + 13x^2 + 4x + 5)$$

$$= 20x^3 + 7x^2 + 11x + 8$$

$$g_3(x) = \frac{(x-1)(x-3)(x-5)}{(6-1)(6-3)(6-5)} = 20(x^3 + 14x^2 + 8)$$

$$= 20x^3 + 4x^2 + 22.$$

(24)

Therefore, the polynomial  $f'(x)$  is

$$f'(x) = f(1)g_0(x) + f(3)g_1(x) + f(5)g_2(x) + f(6)g_3(x)$$

$$= 14(4x^3 + 13x^2 + 22x + 8)$$

$$+ 9(2x^3 + 22x^2 + 13x + 9)$$

$$+ 7(20x^3 + 7x^2 + 11x + 8)$$

$$+ 7(20x^3 + 4x^2 + 22)$$

$$= 9x^3 + 20x^2 + 19x + 12$$

(25)

The polynomial  $f'(x) = 9x^3 + 20x^2 + 19x + 12$  is the same used to share the secret. Cheating detection is done by

multiplying 19 by 20, which gives 12. Assuming users  $U_1, U_3$ , and  $U_5$  want to cheat  $U_6$ . Let the forged shares be  $U_1: (1, 8), U_3: (3, 10), U_5: (5, 17)$  and  $U_6: (6, 7)$ .  $C$  reconstructs a polynomial

$$f'(x) = 3x^3 + 11x^2 + 10x + 7. \quad (26)$$

Multiplying 11 by 20 gives 22, which is not equal to 7. Cheating is detected by cheating detection algorithm in PBLs.

#### 4. Analysis

This section provides the analysis of basic scheme and PBLs in terms of security and privacy with required features and computational overhead. We also compare the security, privacy, and computations in PBLs to Shamir's scheme and Liu et al.'s scheme [6, 14]. PBLs achieves the required features for the secret sharing schemes like SP1, SP2, SP3, SP4, and CO. Proofs for these requirements are provided. For this, we first provide the proof for the security of basic scheme, which is used by PBLs to detect cheating. The section also provides the proof of how secure PBLs is against cheating based on the assumption of OKS because the aim of secret sharing is to make the secret not known to users until reconstruction.

*4.1. Security and Privacy Analysis.* This subsection provides the analysis on the security and privacy of PBLs and proves that the required features for secret sharing schemes are achieved. We show that PBLs achieves the following properties.

- (i) SP1: the secret is not known to all users and adversary  $A$  before reconstruction.
- (ii) SP2: the secret can be reconstructed once it is shared to  $n$  users.
- (iii) SP3: no less than required number of users can reconstruct the secret.
- (iv) SP4: this is based on OKS assumption, which provides the guarantee that no cheating can be successful in PBLs.

First, we show that basic scheme is secure from  $A$  based on OKS adversary model in Section 3.1.3. At initialization of basic scheme, the secret is multiplied by a random element  $r$  with an aim of hiding it. Two security issues rise up in this case:

- (i) security of  $s$  in  $z$  for the basic scheme and
- (ii) security of  $s$  in the polynomial  $f(x)$ .

Proposition 32 proves the security of basic scheme from any adversary, i.e., dishonest user or anyone who is not taking part in the secret sharing cannot obtain the secret  $s$  without the knowledge of  $r$  and its multiplicative inverse.

**Proposition 32.** *Let  $p$  be prime and  $z \equiv s \cdot r \pmod{p}$  such that  $s \in \mathbb{Z}_p$  and  $r \in \mathbb{Z}_p$  are a secret and a random number, respectively. An adversary  $A$  should solve FFDLP to obtain the secret from  $z$  in basic scheme without the knowledge of  $r$  and its multiplicative inverse.*

*Proof.* We showed in Proposition 19 that an element  $z$  cannot reveal any information about  $s$  without the knowledge of  $r$ . It was indicated that it is necessary to solve FFDLP to reveal  $s$  from  $z$ . Therefore, to know the secret from  $z$  in basic scheme, one has to solve FFDLP. Basic scheme is secure from  $A$ .  $\square$

However, we also need to show that the secret cannot be revealed by  $A$  in  $z$ , which is the polynomial used to distribute shares to users. Proposition 33 proves that basic scheme is secure in share generation algorithm.

**Proposition 33.** *Let  $z, s$ , and  $r \in \mathbb{Z}_p$  as defined in Proposition 32 and  $p$  be prime. An adversary  $A$  should solve FFDLP to obtain the secret from  $z$  in the polynomial  $f(x)$  even if multiplicative inverse of  $r$  is known.*

*Proof.* By Proposition 19, it is difficult to obtain  $s$  from  $z$  without the knowledge of  $r$ . However, if  $r$  is known, the secret  $s$  can be obtained. During share generation all users have access to the multiplicative inverse of  $r$  and their share; hence, it is possible to obtain the secret if  $z$  is known. However, the secret and the element  $z$  are coefficients of the polynomial, which are only known to the dealer. But the multiplicative inverse of  $r$  cannot give information of  $s$  and  $z$ . This is the same as solving the FFDLP.  $\square$

Security of secret sharing schemes depends on the private distribution of shares to user so that no user should know the shares of the other users. Therefore, each user has to receive the share from the dealer using a private channel. It is assumed that users do not communicate about their shares to each other unless they collaborate to cheat. Once the secret is divided, the shares do not show any information about the secret. As a result users do not have any information about the secret as assumed by OKS. In addition to this, shares are delivered privately to users and hence cannot know the share of the other users. Lemma 34 proves the fact of SP1 that users do not have access to the secret.

**Lemma 34.** *Any secret share given to a participant in PBLs does not reveal the secret  $s$ .*

*Proof.* In share generation, PBLs uses Shamir's method to share the secret, which uses the polynomial  $f(x)$  of degree  $t - 1$ . Each share is evaluated from  $i$  to give  $f(i)$  for all  $i > 0$  such that  $i$  is the identity of the user. The secret in the polynomial is  $f(0)$ . Shares in Shamir's scheme do not reveal any information of the secret. Since PBLs adopts Shamir's method, the shares generated have the same security as those generated by Shamir's scheme.  $\square$

Lemma 34 proves that no single user can have access to the secret using only his (or her) share. However, the secret can be reconstructed if  $t$  users pool in their shares together. Since the reconstruction is done by a trusted third party called combiner, no user is able to know the secret. However, the secret is obtained by the combiner. We

now prove Proposition 35, which gives a proof on SP2 in PBLs.

**Proposition 35.** *Let  $t$  be the threshold. Any  $t$  users can reconstruct the secret by combining their shares together in PBLs by using secret reconstruction algorithm.*

*Proof.* Secret reconstruction in PBLs is done by interpolating the polynomial  $f'(x)$  by Lagrange interpolation. Given  $t$  points  $(i, v_i)$  for all  $i=1, 2, \dots, t$ , interpolated polynomial as in Equation (19). We rewrite the Equation below for clarity

$$f'(x) = \sum_{i=1}^t v_i \prod_{j=1, j \neq i}^t \frac{x - x_j}{x_i - x_j} = \sum_{i=0}^{t-1} a'_i x^i. \quad (27)$$

By Theorem 11 the polynomial  $f'(x)$  is unique which has degree  $t - 1$ . The dealer used Equation (18) to generate shares and we write it

$$f(x) = \sum_{i=0}^{t-1} a_i x^i. \quad (28)$$

Therefore, the two polynomials in Equations (27) and (28) are equal. By Definition 7,  $a_i = a'_i$ . So,  $a_0 = a'_0$ . Since  $a_0$  is the secret,  $a'_0$  is also a secret, which has been reconstructed from  $t$  shares. Therefore, the secret in PBLs can be reconstructed.  $\square$

If  $t$  shares can reveal the secret, then less than  $t$  shares should not be able to show any information about the secret. This is the recommendation of a secret sharing scheme to be achieved, which is called privacy. Proposition 36 proves SP3 of PBLs such that less than the required threshold can neither recover the secret nor gain information about the secret.

**Proposition 36.** *Let  $t$  be the threshold. Less than  $t$  users cannot reconstruct or know any information of the secret in PBLs.*

*Proof.* We need to show that  $t - 1$  shares do not reveal any information about the secret. Assume  $t - 1$  participants collude to recover the secret, which means they will have  $t - 1$  points to interpolate a polynomial  $f'(x)$  of degree  $t - 1$ . However, these points will interpolate a polynomial of degree  $t - 2$  as far as Theorem 11 is concerned, hence

$$f'(x) = \sum_{i=1}^{t-1} v_i \prod_{j=1, j \neq i}^{t-1} \frac{x - x_j}{x_i - x_j} = \sum_{i=0}^{t-2} a'_i x^i. \quad (29)$$

Equations (27) and (28) are not equal since they have different degrees. The fact that polynomial in Equation (29) is unique makes their coefficient differ as well. The other way is to try to solve a system of  $t - 1$  equations with  $t$  unknowns as shown in the matrix Equation (30).

$$\begin{pmatrix} 1 & ID_1 & \dots & ID_1^{t-1} \\ 1 & ID_2 & \dots & ID_2^{t-1} \\ \vdots & \vdots & \dots & \vdots \\ 1 & ID_{t-1} & \dots & ID_{t-1}^{t-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} f(ID_1) \\ f(ID_2) \\ \vdots \\ f(ID_{t-1}) \end{pmatrix} \quad (30)$$

But it is impossible to solve such equations unless the  $t$ -th term is guessed. Thus, we need at least  $t$  points to interpolate the polynomial, which might be not correct since the  $t$ -th share is secretly delivered to user. By Lemma 34, each share does not reveal the information about the secret. Therefore,  $t - 1$  shares cannot reveal any information about the secret in PBLs.  $\square$

During secret reconstruction,  $t - 1$  users may collude to cheat the  $t$ -th user. Since the aim of the cheaters is to prevent the correct recovery of the secret at the same time they should be able to reconstruct the valid secret. This is avoided in PBLs because all shares from participating users are not revealed during secret reconstruction. Therefore, if  $t - 1$  users communicated their shares, it will be difficult to reconstruct the secret without the  $t$ -th share as proved in Proposition 35. Furthermore, PBLs will be able to detect cheating if  $t - 1$  or less shares are forged. Proposition 37 proves that no cheating can be successful in PBLs even if  $t - 1$  forged shares were presented during secret reconstruction.

**Proposition 37.** *Let  $t$  be the threshold and  $t - 1$  be forged shares presented to the combiner. The forged shares will be detected during secret reconstruction in PBLs.*

*Proof.* Once fake shares are pooled to the combiner, the combiner uses Lagrange interpolation to compute a polynomial  $f'(x)$  of degree  $t - 1$ , which is unique; hence  $f'(x) \neq f(x)$ . After coming up with the polynomial, the combiner will use the multiplicative inverse of the random element  $r^{-1} \in \mathbb{F}_p$  to verify the secret as follows:

$$z \cdot r \equiv s \cdot r \cdot r^{-1} \pmod{p} \equiv s \pmod{p}, \quad (31)$$

where  $s$  is the coefficient of  $x^0$ . This works by Proposition 3. Since forged shares are used in secret reconstruction, the interpolated polynomial  $f'(x)$  will not be equal to the polynomial used by the dealer in share generation algorithm; therefore,  $a'_i \neq a_i$ . Equation (31) will not work; thus  $z \cdot r^{-1} \neq s$ . Therefore, cheating is detected in PBLs.  $\square$

PBLs achieves the share size of  $|v_i| = |s|/\epsilon$  such that  $\epsilon > 0$  is the probability for successful cheating. Theorem 38 provides the security of PBLs and a proof on the share size of PBLs [30–39].

**Theorem 38.** PBLS described in Section 3.2.2 realizing the access structure  $\Gamma$  is  $\varepsilon$ -secure against up to  $t - 1$  cheaters who may somehow know the secret beforehand. Moreover, the share size is  $|v_i| = p^2 = |s|/\varepsilon$ , where  $\varepsilon = 1/p$  and  $v_i$  denotes the share space of the  $i$ -th participant  $U_i$ .

*Proof.* Since  $p$  is prime,  $\mathbb{Z}_p$  is a finite field. We let the secret to be  $p$  bits long so

$$|s| = p. \quad (32)$$

For successful cheating, an adversary A has the sample space of  $p$ . Hence the probability is

$$\varepsilon = \frac{1}{p}. \quad (33)$$

Since every user receives  $f(i)$  and  $y$ , the share is

$$v_i = \{f(i), y\} \quad \text{where } y = r^{-1}. \quad (34)$$

So  $|f(i)| = p$  and  $|y| = p$ . Therefore,

$$|v_i| = p^2. \quad (35)$$

$$\varepsilon(\text{PBLS}, A) = \Pr [g'(x) \text{ passes through a point } (x_h, g(x_h)) \text{ unknown to A}]. \quad (39)$$

Since  $g(x)$  is a polynomial of degree  $t - 1$  over  $\mathbb{F}_p$  from different constant term,  $\varepsilon = 1/p$ . This proves the theorem.  $\square$

Propositions 35 and 36 indicate that PBLS is perfect since the secret can be recovered by the required threshold but not less. This is the same for Shamir's scheme and Liu et al.'s scheme. Theorem 38 indicates that any cheating behavior can be detected, which is similar to Liu et al.'s scheme. Furthermore, PBLS and Liu et al.'s scheme have the same share size given to users. Therefore, PBLS and Liu et al.'s scheme have the same property of cheating detection.

We now compare the security and properties of PBLS with Shamir's scheme and Liu et al.'s scheme. Table 1 shows that comparison of SPs and CO of these schemes.

Table 1 shows that the schemes have similar properties in terms of SP1, SP2, and SP3. However, Shamir's scheme does not provide SP4. It also shows that PBLS achieves CO simultaneously. Even if PBLS provides good aspects in security and privacy, we need to mention that there is possibility that shareholders have some advantages over learning the secret since they have  $b$  as mentioned in Algorithm 28. So, our future research should focus on devising a new scheme to solve this probability.

**4.2. Computation Analysis.** In this subsection, we analyze the computation overhead of PBLS and give a comparison to Shamir's scheme and Liu et al.'s scheme. Three operations are used in this analysis, which are modulo addition (add), modulo multiplication (mul), and modulo inverse (inv).

From Equations (32) and (33),

$$\frac{|s|}{p} = \varepsilon p. \quad (36)$$

This implies that

$$\frac{|s|}{\varepsilon} = p^2. \quad (37)$$

From Equation (35)

$$|v_i| = p^2 = \frac{|s|}{\varepsilon}. \quad (38)$$

Suppose an honest participant  $U_h$  having the share  $v_h = (h, f(h))$  belongs to the  $k$ -th compartment. For a valid but incorrect secret  $s' \in S$  to be accepted by  $U_h$ , after parsing another check polynomial  $g'(x)$  with  $g'(0) = s'$ , the point  $(x_h, g(x_h))$  should lie on the polynomial  $g'(x)$ . So, the successful cheating probability (PBLS, A) of cheaters A against PBLS is defined as

The analysis will consider all the two algorithms. However Algorithm 25 of basic scheme is not considered in this analysis since it is an initialization phase of PBLS.

**4.2.1. Computations in PBLS.** In PBLS, a share is computed from a polynomial  $f(x)$  of degree  $t - 1$ . Therefore, the polynomial has  $t$  terms and  $t - 1$  of them are multiplied by a variable  $x$ . Each share requires modulo addition and modulo multiplication to be computed. Therefore, Table 2 shows computation in share generation of PBLS.

The aim of secret reconstruction is to come up with a secret, which is done using Lagrange interpolation. A polynomial is interpolated using  $t$  points, which are the shares and the identity of the user, i.e.,  $(i, v_i)$ . The polynomial  $f'(x)$  is obtained as in Equation (27), which is rewritten as

$$f'(x) = \sum_{i=1}^t v_i \cdot p_i(x), \quad (40)$$

$$\text{where } p_i(x) = \prod_{j=1, j \neq i}^t \frac{x - x_j}{x_i - x_j}.$$

To compute each  $p_i(x)$ , Table 3 shows the computations in secret reconstruction of PBLS.

**4.2.2. Comparison.** Now, we compare the computation overhead of PBLS with Shamir's scheme and Liu et al.'s scheme. We follow the method done in Section 4.2.1 to provide computation overhead comparison. Table 4 shows the computation

TABLE 1: Comparison of required properties for the schemes.

Scheme	Property				CO
	SP1	SP2	SP3	SP4	
Shamir's scheme	√	√	√	X	1
Liu et al.'s scheme	√	√	√	√	2
PBLS	√	√	√	√	1

√: the property exists; X: the property does not exist.

TABLE 2: Computations in share generation of PBLS.

Share generation				
Operation	mul	add	inv	Total
Number of operations	$n(t-1)$	$nt$	-	$n(t-1)$ mul + $nt$ add

TABLE 3: Computations in secret reconstruction of PBLS.

Secret reconstruction				
Operation	mul	add	inv	Total
Number of operations	$t^3 + t + 1$	$t$	$t$	$(t^3 + t + 1)$ mul + $t$ add + $t$ inv

TABLE 4: Computation overhead of related schemes.

Share generation				
Scheme	Operation			Total
	mul	add	inv	
Liu et al.'s scheme	$2n(t-1)$	$2nt$	-	$2n(t-1)$ mul + $2nt$ add
Shamir's scheme	$n(t-1)$	$nt$	-	$n(t-1)$ mul + $nt$ add
PBLS	$n(t-1)$	$nt$	-	$n(t-1)$ mul + $nt$ add
Secret reconstruction				
Scheme	Operation			Total
	mul	add	inv	
Liu et al.'s scheme	$2(t^3 + t + 1)$	$2t$	$2t$	$2(t^3 + t + 1)$ mul + $2t$ add + $2t$ inv
Shamir's scheme	$t^3 + t + 1$	$t$	$t$	$(t^3 + t)$ mul + $t$ add + $t$ inv
PBLS	$t^3 + t + 1$	$t$	$t$	$(t^3 + t + 1)$ mul + $t$ add + $t$ inv

overhead comparisons in share generation phase and secret reconstruction phase.

Results in Table 4 show that PBLS and Shamir's scheme have the same computation overhead at share generation phase. The result is due to the use of a single polynomial when sharing and distributing the secret to users. However, comparing this result to Liu et al.'s scheme, the computation overhead is reduced by half in share generation phase. This indicates that PBLS has an efficient way to share the secret as compared to Liu et al.'s scheme but comparable with Shamir's scheme.

Considering secret reconstruction process, computation overhead on PBLS is higher by 1 mul as compared to Shamir's scheme. This is so because of cheating detection in PBLS in which the operation is 1 mul but Shamir's scheme does not have. However, results show that computation overhead of Liu et al.'s scheme at secret reconstruction still doubles

as compared to PBLS. Therefore, PBLS is more efficient as compared to Liu et al.'s scheme in the concern of computation overhead.

## 5. Conclusion

In this paper, we proposed a new linear  $(t, n)$  threshold secret sharing scheme called PBLS, which is based not only on Shamir's scheme but also on ElGamal cryptosystem. PBLS satisfies the required properties like security, recoverability, privacy, cheating detection, and share size. PBLS is  $(t, n)$  threshold scheme, which requires at least  $t$  shares to reconstruct the secret while any less than  $t$  should not be able to do it.

Firstly, we draw the required features that secret sharing schemes satisfied by reviewing and analyzing some previous schemes like Shamir's and Liu et al.'s. The required features

drawn are security, recoverability of the secret, privacy of the secret, cheating detection of the forged shares presented for reconstruction of a secret and share size given to each user. We also reviewed some basic mathematical and cryptographic concepts, which assisted in designing methods for cheating detection such as finite fields and ElGamal cryptosystem.

Based on the withdrawn required features of secret sharing schemes, basic scheme and PBLs were designed. Basic scheme aims at hiding the secret, which is the initialization of PBLs. The secret is revealed during cheating detection. This is an idea of ElGamal who developed a cryptosystem that can hide a message using field elements. PBLs applies Shamir's secret sharing scheme to share the secret. Polynomial  $f(x)$  is used in share generation phase such that the coefficient of  $x$  is the element hiding the secret. Secret reconstruction was done by interpolating a polynomial using Lagrange interpolation. Cheating detection was achieved by multiplying the coefficient of  $x$  of the polynomial  $f(x)$  by multiplicative inverse of  $r$  to reveal the secret  $f'(0)$ .

After the design of PBLs, an analysis was made, which was presented in two ways. These were security analysis and privacy analysis with required features and computational overhead analysis. It was determined that the security with privacy of PBLs was similar to Liu et al.'s scheme. However, in terms of cheating, Shamir's scheme proved to be weak. Cheating detection was attained in both PBLs and Liu et al.'s schemes even though PBLs used only one polynomial. Furthermore, the required features like recoverability were analyzed to be similar to Liu et al.'s scheme. Computational analysis showed that number of operations in PBLs is almost equal to the computations in Shamir's scheme, which is half of Liu et al.'s scheme. This analysis made PBLs to be a better scheme in terms of efficiency than Liu et al.'s scheme and in terms of security than Shamir's scheme.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The results in this paper are part of Kenan Kingsley Phiri's Master degree thesis. Corresponding author is Hyunsung Kim. This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B04032598).

## References

- [1] R. Oppliger, *Contemporary Cryptography*, Artech House, Boston, MA, USA, 2005.

- [2] S. Vaudenay, *A Classical Introduction to Cryptography: Applications for Communications Security*, Springer, New York, NY, USA, 2006.
- [3] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Florida, Fla, USA, 1996.
- [4] A. Beigel, "Secret-sharing schemes: a survey," *Lecture Notes in Computer Science*, vol. 6639, pp. 11–46, 2011.
- [5] K. M. Martin, "Challenging the adversary model in secret sharing schemes: Coding and cryptography II," in *Proceedings of the Royal Flemish Academy of Belgium for Science and Art*, pp. 45–63, 2008.
- [6] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [7] M. Tompa and H. Woll, "How to share a secret with cheaters," *Journal of Cryptology*, vol. 1, no. 3, pp. 133–138, 1989.
- [8] B. Srikanth, G. Padmaja, S. Khasim, P. V. S. Likhshmi, and A. Haritha, "Secure bank authentication using image processing and visual cryptography," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 2432–2437, 2014.
- [9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 89–98, New York, NY, USA, November 2006.
- [10] T. Tassa, "Generalized oblivious transfer by secret sharing," *Designs, Codes and Cryptography*, vol. 58, no. 1, pp. 11–21, 2011.
- [11] M. O. Rabin, "Randomized byzantine generals," in *Proceedings of the 24th Annual Symposium on Foundations of Computer Science (SFCS '83)*, pp. 403–409, Tucson, AZ, USA, November 1983.
- [12] P. Lin, Y. Chen, M. Hsu, and F. Juang, "Secret sharing mechanism with cheater detection," in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA '13)*, pp. 1–4, Kaohsiung, Taiwan, October 2013.
- [13] Y. Liu, Y. Zhang, and Y. Hu, "Efficient  $(t, n)$  secret sharing scheme against cheating," *Journal of Computational Information Systems*, vol. 8, no. 9, pp. 3815–3821, 2012.
- [14] Y. Liu, Z. Wang, and W. Yan, "Linear  $(k, n)$  secret sharing scheme with cheating detection," in *Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM '15)*, pp. 1942–1947, Liverpool, UK, October 2015.
- [15] I.-C. Lin and C.-C. Chang, "A  $(t, n)$  threshold secret sharing system with efficient identification of cheaters," *Computing and Informatics*, vol. 24, no. 5, pp. 529–541, 2005.
- [16] R. Kalombe and M. Kamble, "Cheater detection and identification based on shamir scheme," *International Research Journal of Computer Science Engineering and Application*, vol. 2, no. 2, pp. 255–259, 2013.
- [17] S. Obana, "Almost optimum  $t$ -cheater identifiable secret sharing scheme," *Lecture Notes in Computer Science*, vol. 6632, pp. 284–302, 2011.
- [18] D. Pasaila, V. Alexa, and S. Iftene, "Cheating detection and cheater identification in CRT-based secret sharing schemes," *International Journal of Computing*, vol. 9, no. 2, pp. 107–117, 2010.

- [19] C. Guo, R. Zhuang, L. Yuan, and B. Feng, "A group authentication scheme supporting cheating detection and identification," in *Proceedings of the 9th International Conference on Frontier of Computer Science and Technology (FCST'15)*, vol. 52, pp. 110–114, Dalian, China, August 2015.
- [20] L. Harn, "Generalised cheater detection and identification," *IET Information Security*, vol. 8, no. 3, pp. 171–178, 2014.
- [21] M. P. Jhanwar and R. Safavi-Naini, "Unconditionally-secure robust secret sharing with minimum share size," *Lecture Notes in Computer Science*, vol. 7859, pp. 96–110, 2013.
- [22] A. Bishop and V. Pastro, "Robust secret sharing schemes against local adversaries," *Lecture Notes in Computer Science*, vol. 9615, pp. 327–356, 2016.
- [23] M. P. Jhanwar and R. Safavi-Naini, "On the share efficiency of robust secret sharing and secret sharing with cheating detection," *Lecture Notes in Computer Science*, vol. 8250, pp. 179–196, 2013.
- [24] J. S.-T. Juan, Y.-L. Chuang, and M.-J. Li, "An online verifiable and detectable  $(t, n)$  multi-secret sharing scheme based on a hyperelliptic function," *Journal of Information and Computational Science*, vol. 8, no. 4, pp. 688–696, 2011.
- [25] M. Backes, A. Kate, and A. Patra, "Computational verifiable secret sharing revisited," *Lecture Notes in Computer Science*, vol. 7073, pp. 590–609, 2011.
- [26] M. Carpentieri, A. De Santis, and U. Vaccaro, "Size of shares and probability of cheating in threshold schemes," *Lecture Notes in Computer Science*, vol. 765, pp. 118–125, 1994.
- [27] W. Ogata and K. Kurosawa, "Optimum secret sharing scheme secure against cheating," *Lecture Notes in Computer Science*, vol. 1070, pp. 200–211, 1996.
- [28] M. Liu, L. Xiao, and Z. Zhang, "Linear multi-secret sharing schemes based on multi-party computation," *Finite Fields and Their Applications*, vol. 12, no. 1, pp. 704–713, 2006.
- [29] J. Pieprzyk and X. Zhang, "Cheating prevention in linear secret sharing," *Information Security and Privacy*, vol. 2384, no. 1, pp. 121–135, 2002.
- [30] R. Cramer, I. B. Damgård, N. Döttling, S. Fehr, and G. Spini, "Linear Secret sharing schemes from error correcting codes and universal hash functions," *Lecture Notes in Computer Science*, vol. 9057, pp. 313–336, 2015.
- [31] H. Ghodosi, "Comments on Harn–Lin's cheating detection scheme," *Designs, Codes and Cryptography*, vol. 60, no. 1, pp. 63–66, 2011.
- [32] J. Hoffstein, J. Pipher, and J. H. Silverman, *An Introduction to Mathematical Cryptography*, Springer, New York, NY, USA, 2008.
- [33] A. Slinko, *Algebra for Applications: Cryptography, Secret Sharing, Error Correcting, Fingerprinting, Compression*, Springer, Heidelberg, Germany, 2015.
- [34] M. W. Baldoni, C. Ciliberto, and G. M. P. Cattaneo, *Elementary Number Theory, Cryptography and Codes*, Springer-Verlag, Berlin Heidelberg, Germany, 2009.
- [35] T. El Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *Lecture Notes in Computer Science*, vol. 196, pp. 10–18, 1985.
- [36] T. Koshiy, *Elementary Number Theory with Applications*, Academic press, New York, NY, USA, 2nd edition, 2007.
- [37] T. C. Wu and T. S. Wu, "Cheating detection and cheater identification in secret sharing schemes," *IEEE Transactions on Computers and Digital Techniques*, vol. 142, no. 1, pp. 367–369, 1995.
- [38] A. Adhikari, K. Morozov, S. Obana, P. S. Roy, K. Sakurai, and R. Xu, "Efficient threshold secret sharing schemes secure against rushing cheaters," *Lecture Notes in Computer Science*, vol. 10015, pp. 3–23, 2016.
- [39] J. Pramanik, P. S. Roy, S. Dutta, A. Adhikari, and K. Sakurai, "Secret sharing schemes on compartmental access structure in presence of cheaters," *Lecture Notes in Computer Science*, vol. 11281, pp. 171–188, 2018.



## Research Article

# Research on Defensive Strategy of Real-Time Price Attack Based on Multiperson Zero-Determinant

Zhuoqun Xia,<sup>1</sup> Zhenwei Fang,<sup>1</sup> Fengfei Zou,<sup>1</sup> Jin Wang ,<sup>1</sup> and Arun Kumar Sangaiah <sup>2</sup>

<sup>1</sup>School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China

<sup>2</sup>School of Computing Science and Engineering, Vellore Institute of Technology, Vellore 632014, India

Correspondence should be addressed to Jin Wang; [jinwang@csust.edu.cn](mailto:jinwang@csust.edu.cn)

Received 24 March 2019; Accepted 13 June 2019; Published 16 July 2019

Guest Editor: Ki-Hyun Jung

Copyright © 2019 Zhuoqun Xia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The smart grid solves the growing load demand of electrical customers through two-way real-time communication of electricity supply and demand sides and home energy management system (HEMS). However, these technical features also bring network security risks to the real-time price signal of the smart grid. The real-time price attack (RTPA) can maliciously raise the real-time price in smart meter, resulting in an increase in electrical customers load demand, causing the extensive damage to the power transmission lines due to overload. In this paper, we based on the behavioral relationship between load demand of electrical customers and real-time price of electricity suppliers (ES), defined the game relationship between RTPA, ES, and electrical customers, established a price elasticity of electricity demand (PEED) model, and proposed a defensive strategy of real-time price attack based on multiperson zero-determinant strategy (MPZDS). The experimental results show that the combination of MPZDS to some extent cut the expected load demand of electrical customers and protect the safety of power transmission lines.

## 1. Introduction

Smart grid (SG) is a cyberphysical system (CPS) that can be described as the next generation power grid, in which not only generation, transmission, and distribution of power but also utilization and management aspects of the grid are upgraded to improve the grid's reliability, efficiency, flexibility, scalability, safety, security, and environmental friendliness. Different from the traditional power grid, the SG relies on a two-way communication infrastructure and its overall performance is optimized using a number of existing and emerging technologies, including wireless sensor networks (WSNs) and other communication technologies, smart sensor devices, automation systems, monitoring, computers, and renewable energy solutions [1, 2]. However, these features have also brought many security risks to the smart grid; for example, on December 23, 2015, the Ukrainian power grid was attacked by malicious software called "BlackEnergy", resulting in a massive blackout, this blackout was caused by the switching of seven substations, nearly two million people were influenced, and the time of power cut is 3-6h.

With the rapid development of new technologies such as intelligent manufacturing [3, 4], in order to improve the reliability and effectiveness of smart grid, some scholars have proposed some concepts such as smart meters [5] and smart home management system [6], but these infrastructures also bring new security vulnerabilities to the smart grid. Power generation, distribution, and electricity consumption units are vulnerable to security attacks due to their strong openness, but attacker attacks on power generation and distribution units require a higher cost than electricity consumption units, therefore, to ensure that electricity consumption units from various types of cyberattacks are crucial.

Information injection attacks on electricity consumption units are aimed at tampering with electricity price signals or information [7], and attackers can launch attacks more easily through automated and distributed software injection agents. Furthermore, because smart grids possess some features like load control and automatic energy consumption dispatch, this makes attacks more efficient. The automatic energy consumption dispatching unit schedules the energy consumption of indoor energy consumption equipment through

the electricity price given by the electricity supplier and the power consumption information of the electrical customers, so as to minimize the energy consumption. The electricity price information is transmitted through the information network, which makes the false price injection attack can potentially lead to changes in the load demand of the electricity consumption unit, resulting in unbalanced load demand control. In recent years, the academia conducted a deep study on the cyberphysical security of smart grid [8–10]; however, there are few existing researches on the analysis and prevention of real-time price attack (RTPA). Real-time price information release requires low delay, and delay attack will bring too high real-time price information delay, which will affect the electrical customers to obtain real-time price information, thereby affecting the demand response of electrical customers and real-time scheduling of electricity supplier [11].

Aiming at the scaling attack and delay attack of real-time price signal integrity, Tan et al. established a closed-loop based on the interdependence between real-time price signal and incentive demand of electricity price and deduced the basic conditions of real-time price system stability in the case of attack by using control theory; the experimental results help the system operators to effectively analyze the impact of attack on the stability parameters of the real-time price system [12, 13]. Based on scaling attack and time-delay attack, Giraldo et al. proposed an analysis method based on sensitivity function on the basis of considering power market model and real-time price integrity attack model, by adding low-pass filters to the price signal, selecting the price update cycle and controller parameters, designing robust control algorithm, and measures to detect abnormal behavior of the system to reduce the impact of the attack [14, 15]. However, it does not consider the potential impact of the attack on power demand side. Jia et al. have studied the false data attack against the node marginal price and proposed a geometrical analysis framework based on the upstream and downstream boundary conditions of the optimal data attack and validated the validity of the framework by PJM 5-node power system, IEEE 14-node power system, and IEEE 118-node power system [16]; however, the framework only guarantees the security of the node's marginal price data and does not analyze the impact of the attack from the supply or demand side. The price tampering attack will cause the change of load configuration information of individual electrical customers, resulting in the load transfer or load redistribution, which leads to the change of load configuration information of the power network, which eventually leads to transmission lines damage in large areas due to overload, thus forming a cascading failure [17]. However, the above researches on the real-time price market focus on the protection of single-stage security and do not analyze in depth the impact of RTPA on the electrical customers load demand and how to prevent real-time price attack from the viewpoint of ensuring the normal load demand of electrical customers.

In information security analysis, the classical prisoner dilemma model clarifies a rational game between the implementation and effectiveness of an attack between an attacker (such as FDIA) and a defender (such as IDS). In order to

maximize the impact of an attack, the attacker will choose to strengthen the attack. At the same time, the defender will also choose to strengthen the defense to optimize the defense effect, which means that there is a noncooperative behavior between the attacker and the defender is the dominant strategy between the two sides of the game. However, in the Repeated Prisoner's Dilemma Game, the choice of behavior strategies of both players is not the same as before. In order to avoid being punished for previous noncooperative behavior, each player may choose cooperation in the subsequent game process, which provides a theoretical basis for the generation of zero-determinant strategy. Zero-determinant strategy (ZDS) shows that the profits of both players satisfy a certain linear relationship, or one player controls the profits of the other player between the incompatible profits and the cooperative profits.

In order to solve the above research shortcomings of the RTPA, this paper proposes a RTPA defense strategy based on multiplayer zero-determinants under repeated game to minimize the impact of attack and ensure the user's expected load demand and transmission line safety.

Our contributions are summarized as follows:

- (i) We define the load supply and consumption behavior of ES, attackers, and electrical customers according to the relationship between load demand of electrical customers and real-time price of ES.
- (ii) We set up a  $3 \times 2$  repeated game according to the behavioral relationship among the ES, the attackers, and the electrical customers, and each of the three players in the game has two kinds of behavioral states, namely, the active state and the idle state. In each stage of the game, all three players in the game may be active or idle and have the same structure of the game payoff matrix.
- (iii) We propose Multiplayer Zero-Determinant Strategy (MPZDS) to ensure the safety of electrical customers' side and transmission lines in smart grid to prevent RTPA.

The structure of this paper is as follows. Section 2 defines the behavioral characteristics of ES, attackers, and electrical customers and establishes the power transmission line model. We analyze the game situation of ES, attackers, and electrical customers according to the price elasticity of electricity demand (PEED) model in Section 3. We next combine the MPZDS for safety analysis in Section 4. The effectiveness of the proposed method is validated through experiments in Section 5. The conclusion is shown in Section 6.

## 2. Related Definitions

This section models the power transmission lines and defines the behavior of the ES, attackers, and electrical customers to describe the dynamic characteristics of all the three.

*2.1. Power Transmission Lines.* From the point of view of power supply and demand system of the smart grid, each node in  $Z$  can represent a power generation unit, a power

transmission unit, and a power consumption unit (such as a common customers, an industrial consumers, or a data center, and so on),  $L$  represents the transmission line between nodes, and the smart grid can be abstracted as  $G(Z, L)$ . In the node set  $Z$ , it includes the set of generating nodes  $PU \subset Z$ , the set of transmitting nodes  $OU \subset Z$ , and the set of electricity utilization nodes  $DU \subset Z$ . In the transmission line,  $LD_i$  represents the power load distribution of the  $i_{th}$  transmission line under stable state,  $LD_{total}$  represents the total load demand of the electrical customers in the stable state,  $L_{max}$  represents the maximum loadability of each transmission line, and  $R_{LP}$  represents the protection rate of power transmission line. When the power load transmission exceeds the maximum loadability of the transmission line, the line  $i$  will be damaged due to overload, so the transmission line should meet the following conditions:

$$\begin{aligned} & \max R_{LP} \\ & \sum_{i \in L} LD_i \leq LD_{total} \\ & 0 < LD_i \leq L_{max} \end{aligned} \quad (1)$$

**2.2. Real-Time Price Attack.** In smart grid, there are many ways to attack real-time electricity price signals. For example, attackers can destroy the intermediate nodes of smart grid communication network (such as routers) and obtain the encryption/decryption key in ISO issued/smart meter to intercept and forge data packets containing price data. In addition, attackers launch small-scale tampering attacks against real-time price signals, which may also iteratively amplify the attack effect through feedback, resulting in inefficiency of power system operation and large-scale fluctuation of load demand, and may cause larger and even whole network power failures, such as power outages.

RTP is a dynamic price mechanism, the update cycle is usually 1h or 0.5h, or even shorter. At present, the minimum RTP update cycle is 5 minutes on the international. In order to better reflect the game characteristics among ES, attackers and electrical customers, the update cycle of RTP in this paper is 5 minutes. Suppose  $t_0$  and  $t$  are the starting and ending moments of a single-stage game,  $\Delta T$  is the period length of a single-stage game, and  $t_0 = t - \Delta T$ , where  $D_{RTPA}(i)$  is the relative increase of the customers load demand under the RTPA in the  $i_{th}$  period,  $\Delta P$  is the relative increase of the electricity price under the RTPA in the  $i_{th}$  period, and then the behavior of RTPA can be defined as

$$\begin{aligned} D_{RTPA}(i) &= +\Gamma_i, \\ \Delta P &\geq 0 \end{aligned} \quad (2)$$

There to, + indicates the increases of load demand, and  $\Gamma_i \geq 0$ ; if the RTPA is in an idle state, the relative increase of load demand is zero at this period; that is,  $D_{RTPA}(i) = 0$ .

**2.3. Stability Control of Home Energy Management System.** Some studies have shown that the robustness and convergence of the system model can be improved by optimizing

the activation function of machine learning methods such as neural networks, but it cannot meet the requirements of both suppliers and demanders at the same time [18]. But the home energy management system (HEMS) is able to schedule the power consumption scheme of the intelligent electrical appliances optimally according to the information such as the real-time price and reduce the use of the intelligent electrical appliances during the high electricity price; in this way, the supply and demand balance between the electricity supply of ES and the electricity consumption of electrical customers can be balanced. In this paper, to reduce the impact of increased load demand caused by RTPA, HEMS moderately reduces the electrical customers load demand over an optional period of time. Suppose  $t_0$  and  $t$  are the starting and ending moments of a single-stage game,  $\Delta T$  is the period length of a single-stage game, and  $t_0 = t - \Delta T$ , where  $D_{HEMS}(i)$  is the relative reduction of the customers load demand under the stability control of HEMS in the  $i_{th}$  period,  $\Delta P$  is the relative increase of the electricity price under the RTPA in the  $i_{th}$  period, and then the stability control behavior of HEMS can be defined as

$$\begin{aligned} D_{HEMS}(i) &= -\Lambda_i, \\ \Delta P &\geq 0 \end{aligned} \quad (3)$$

There to, - indicates the decreases of load demand, and  $\Lambda_i \geq 0$ ; if the HEMS is in an idle state, the relative decrease of load demand is zero at this period; that is,  $D_{HEMS}(i) = 0$ .

**2.4. Electricity Suppliers Scheduling.** By intercepting and forging the electricity price signal from the ES, the attacker maliciously increases the real-time price in the smart meter, causing a large amount of power load in the current period to be transferred to other periods to increase the total load demand on the demand side. From the point of view of ES, in order to meet the increased load demand of electrical customers, the ES can increase the planned power generation. Suppose  $t_0$  and  $t$  are the starting and ending moments of a single-stage game,  $\Delta T$  is the period length of a single-stage game, and  $t_0 = t - \Delta T$ , where  $D_{ES}(i)$  is the relative increase of the planned power generation of ES in the  $i_{th}$  period,  $\Delta P$  is the relative increase of the electricity price under the RTPA in the  $i_{th}$  period, and then the behavior of ES can be defined as

$$\begin{aligned} D_{ES}(i) &= +\Phi_i, \\ \Delta P &\geq 0 \end{aligned} \quad (4)$$

There to, + indicates the increases of planned power generation, and  $\Phi_i \geq 0$ ; if the ES does not increase the planned power generation, that is, the ES is idle, and then relative increase of planned power generation is zero at this period, that is,  $D_{ES}(i) = 0$ .

### 3. Price Elasticity of Electricity Demand

In the process of sending price information from the electricity price database to the Energy Consumption Controller

(ECC), RTPA can intercept and forge real-time price signals from ES to improve the electricity price in the ECC of smart meters. After receiving the wrong price information, the power users transfer the load demand of the current period to other periods (such as low price period), which will increase the total load demand of the power users and also increase the power supply pressure of the low price period, which will bring economic losses to the power users and destroy the balance of power supply and demand [19].

From (2), (3), and (4), we know that RTPA, HEMS, and ES may be active or idle in each stage of the game. In this paper, we consider eight possible game scenarios in a single-stage period and divide the load demand of electrical customers into single-period (self-elasticity) load demand and multiperiod (cross-elasticity) load demand by the behavior analysis of price elasticity of electricity demand [20], and the impact of RTPA on the total load demand of electrical customers is analyzed by PEED model in eight game scenarios. Aalami et al. proposed single-period and multiperiod models of price elasticity of electricity demand [21]; we obtain a PEED model that contains both self-elasticity and cross-elasticity by reducing the time granularity; the PEED model is as follows:

$$D(i) = D_{bd}(i) + \sum_{j=1}^{288} E_{i,j} \times \frac{D_{bd}(i) \times [P_{gd}(j) - P_{bd}(j)]}{P_{bd}(j)} \quad (5)$$

$$E_{i,j} = \frac{(D_{gd}(i) - D_{bd}(i)) / D_{bd}(i)}{(P_{gd}(j) - P_{bd}(j)) / P_{bd}(j)} \quad (6)$$

There,  $i = 1, 2, \dots, 288$  is the number of periods; when  $i = j$ ,  $E_{i,j}$  is the self-elasticity coefficient of the price of electricity demand; when  $i \neq j$ ,  $E_{i,j}$  is the cross-elasticity coefficient of price of electricity demand.  $D_{bd}(i)$  is the load demand in the  $i_{th}$  period of the base day,  $P_{bd}(j)$  is the real-time price in the  $j_{th}$  period of the base day,  $D_{gd}(i)$  is the load demand in the  $i_{th}$  period of the goal day,  $P_{gd}(j)$  is the real-time price in the  $j_{th}$  period of the goal day, and  $D(i)$  is the total load demand of electrical customers in the  $i_{th}$  period.

*Case 1* (idle ES, RTPA and HEMS). The total load demand of electrical customers is in stable state. In (2), (3), and (4),  $\Delta P = 0$ ,  $D_{ES}(i) = D_{RTPA}(i) = D_{HEMS}(i) = 0$ , and the total load demand of electricity users in the  $i_{th}$  period is

$$D(i) = D_{bd}(i) + \sum_{j=1}^{288} E_{i,j} \times \frac{D_{bd}(i) \times [P_{gd}(j) - P_{bd}(j)]}{P_{bd}(j)} \quad (7)$$

$$:= D_{i,N}$$

*Case 2* (idle ES, active RTPA, and idle HEMS). RTPA is maliciously raising electricity price, resulting in an increase in total load demand. In (3) and (4),  $D_{ES}(i) = D_{HEMS}(i) = 0$ , and the total load demand of electricity users in the  $i_{th}$  period is

$$D(i) = D_{bd}(i) + \sum_{j=1}^{288} E_{i,j} \times \frac{D_{bd}(i) \times [P_{gd}(j) - P_{bd}(j)]}{P_{bd}(j)}$$

$$+ D_{RTPA}(i) = D_{i,N} + D_{RTPA}(i) \quad (8)$$

*Case 3* (idle ES, idle RTPA, and active HEMS). HEMS aims to reduce the load demand of electrical customers by closing part of the interruptible or noninterruptible load and transferring part of the load demand to other periods in the optional period (such as peak period). In (2) and (4),  $\Delta P = 0$ ,  $D_{ES}(i) = D_{RTPA}(i) = 0$ , and the total load demand of electricity users in the  $i_{th}$  period is

$$D(i) = D_{bd}(i) + \sum_{j=1}^{288} E_{i,j} \times \frac{D_{bd}(i) \times [P_{gd}(j) - P_{bd}(j)]}{P_{bd}(j)} \quad (9)$$

$$+ D_{HEMS}(i) = D_{i,N} + D_{HEMS}(i)$$

*Case 4* (idle ES, active RTPA, and active HEMS). RTPA maliciously raise the electricity price and increase the load demand of electrical customers. Under the stability control of HEMS, electrical customers can manually or automatically transfer part of the load to other periods. At this time, the coefficient of price elasticity of electricity demand mainly shows cross-elasticity. The total load demand of electricity users in the  $i_{th}$  period is

$$D(i) = D_{bd}(i) + \sum_{j=1}^{288} E_{i,j} \times \frac{D_{bd}(i) \times [P_{gd}(j) - P_{bd}(j)]}{P_{bd}(j)} + D_{RTPA}(i) \quad (10)$$

$$+ D_{HEMS}(i) = D_{i,N} + D_{RTPA}(i) + D_{HEMS}(i)$$

under the combined action of RTPA and HEMS, RTPA can increase the load demand of electrical customers by increasing the real-time price, and meanwhile, HEMS combined with MPZDS minimizes the impact of attacks, so that the total load demand of electrical customers is close to stable value. If  $|D_{HEMS}(i)| < |D_{RTPA}(i)|$ , then the total load demand of electrical customers will be greater than the stable value, that is  $D(i) > D_{i,N}(i)$ ; if  $|D_{HEMS}(i)| \geq |D_{RTPA}(i)|$ , then  $D(i) \leq D_{i,N}(i)$ , but the stable control gain of HEMS in this case is slightly lower than that of Case 3. In this paper, we assume that both the game formed by HEMS and RTPA are rational, and then the MPZDS analysis in the following based on the condition  $|D_{HEMS}(i)| \geq |D_{RTPA}(i)|$ .

*Case 5* (active ES, idle RTPA, and idle HEMS). ES increase the planned power generation; then the supply of electricity is greater than the demand of electrical customers for electricity. In (2) and (3),  $\Delta P = 0$ ,  $D_{RTPA}(i) = D_{HEMS}(i) = 0$ , and the total load demand of electricity users in the  $i_{th}$  period is

$$D(i) = D_{bd}(i) + \sum_{j=1}^{288} E_{i,j} \times \frac{D_{bd}(i) \times [P_{gd}(j) - P_{bd}(j)]}{P_{bd}(j)} + D_{ES}(i)$$

$$= D_{i,N} + D_{ES}(i) \quad (11)$$

*Case 6* (active ES, active RTPA, and idle HEMS). RTPA is maliciously raising electricity price, resulting in increased load demand of electrical customers. In (3),  $D_{HEMS}(i) = 0$ , and the total load demand of electricity users in the  $i_{th}$  period is

$$\begin{aligned} D(i) &= D_{bd}(i) + \sum_{j=1}^{288} E_{i,j} \\ &\times \frac{D_{bd}(i) \times [P_{gd}(j) - P_{bd}(j)]}{P_{bd}(j)} + D_{ES}(i) \\ &+ D_{RTPA}(i) = D_{i,N} + D_{ES}(i) + D_{RTPA}(i) \end{aligned} \quad (12)$$

*Case 7* (active ES, idle RTPA, and active HEMS). On the one hand, HEMS aims to reduce the load demand of electrical customers by closing part of the interruptible or noninterruptible load and transferring part of the load demand to other periods in the optional period (such as peak period). On the other hand, ES increase the planned power generation to meet the demand of electrical customers for electricity. In (2),  $\Delta P = D_{RTPA}(i) = 0$ , and the total load demand of electricity users in the  $i_{th}$  period is

$$\begin{aligned} D(i) &= D_{bd}(i) + \sum_{j=1}^{288} E_{i,j} \\ &\times \frac{D_{bd}(i) \times [P_{gd}(j) - P_{bd}(j)]}{P_{bd}(j)} + D_{ES}(i) \\ &+ D_{HEMS}(i) = D_{i,N} + D_{ES}(i) + D_{HEMS}(i) \end{aligned} \quad (13)$$

if  $|D_{HEMS}(i)| < |D_{ES}(i)|$ , then the total load demand of electrical customers will be greater than the stable value, that is  $D(i) > D_{i,N}(i)$ ; if  $|D_{HEMS}(i)| \geq |D_{ES}(i)|$ , then  $D(i) \leq D_{i,N}(i)$ . In this paper, we assume that both the game formed by ES and HEMS are rational; then the MPZDS analysis in the following is based on the condition  $|D_{HEMS}(i)| \geq |D_{ES}(i)|$ .

*Case 8* (active ES, RTPA, and HEMS). RTPA is maliciously raising electricity price, resulting in increased load demand of electrical customers. The electrical customers can manually or automatically transfer part of the load to other periods under stability control of HEMS. At the same time, ES

increase the planned power generation to meet the demand of electrical customers for electricity; the total load demand of electricity users in the  $i_{th}$  period is

$$\begin{aligned} D(i) &= D_{bd}(i) + \sum_{j=1}^{288} E_{i,j} \\ &\times \frac{D_{bd}(i) \times [P_{gd}(j) - P_{bd}(j)]}{P_{bd}(j)} + D_{ES}(i) \\ &+ D_{RTPA}(i) + D_{HEMS}(i) \\ &= D_{i,N} + D_{ES}(i) + D_{RTPA}(i) + D_{HEMS}(i) \end{aligned} \quad (14)$$

under the combined action of RTPA, ES, and HEMS, RTPA can increase the load demand of electrical customers by increasing the real-time price, and meanwhile, HEMS combined with MPZDS minimizes the impact of attacks, so that the total load demand of electrical customers is close to stable value. In addition, ES increase the planned power generation to meet the demand of electrical customers for electricity in order to achieve the balance between supply and demand of electricity. If  $|D_{HEMS}(i)| < |D_{RTPA}(i) + D_{ES}(i)|$ , then the total load demand of electrical customers will be greater than the stable value, that is  $D(i) > D_{i,N}(i)$ ; if  $|D_{HEMS}(i)| \geq |D_{RTPA}(i) + D_{ES}(i)|$ , then  $D(i) \leq D_{i,N}(i)$ . In this paper, we assume that all the three players formed by HEMS, RTPA, and ES are rational, and then the MPZDS analysis in the following is based on  $|D_{HEMS}(i)| \geq |D_{RTPA}(i) + D_{ES}(i)|$ .

## 4. Price Elasticity of Electricity Demand

*4.1. Multiperson Zero-Determinant Strategy.* Consider a  $3 \times 2$  repeated game, note that  $n_i(t)$  is the behavioral states taken by game player  $i$  in the  $t_{th}$  stage of the game, and “1” means the game player is active in the current game stage and “2” means the game player is idle in the current game stage, that is,  $n_1(t), n_2(t), n_3(t)$  are the behavioral states taken by the HEMS, RTPA, and ES, respectively, in the  $t_{th}$  stage of the game. Let  $\mathbf{n}(t)$  be the behavioral states of all game players in the  $t_{th}$  stage of the game, and  $\mathbf{n}(t) \in \{1, 2\}^3 := \mathbf{S}$ . Based on the PEED model, if HEMS takes action state  $n_1(t) = u$ , RTPA takes action state  $n_2(t) = v$ , and ES takes action state  $n_3(t) = w$  in the current game stage, where  $u, v, w \in \{1, 2\}$ , then  $D_{u,v,w}$  represents the total load demand of electrical customers in the current game stage, and the single-stage game payoff matrix is a  $2 \times 4$  matrix, as shown in Table 1.

$M$

$$= \begin{bmatrix} p_H^{1,1} p_A^{1,1,1} p_E^{1,1,1} \dots p_H^{1,1,1} (1 - p_A^{1,1,1}) (1 - p_E^{1,1,1}) \dots (1 - p_H^{1,1,1}) p_A^{1,1,1} (1 - p_E^{1,1,1}) (1 - p_H^{1,1,1}) (1 - p_A^{1,1,1}) p_E^{1,1,1} (1 - p_H^{1,1,1}) (1 - p_A^{1,1,1}) (1 - p_E^{1,1,1}) \\ p_H^{1,1,2} p_A^{1,1,2} p_E^{1,1,2} \dots p_H^{1,1,2} (1 - p_A^{1,1,2}) (1 - p_E^{1,1,2}) \dots (1 - p_H^{1,1,2}) p_A^{1,1,2} (1 - p_E^{1,1,2}) (1 - p_H^{1,1,2}) (1 - p_A^{1,1,2}) p_E^{1,1,2} (1 - p_H^{1,1,2}) (1 - p_A^{1,1,2}) (1 - p_E^{1,1,2}) \\ \dots \\ p_H^{2,2,1} p_A^{2,2,1} p_E^{2,2,1} \dots p_H^{2,2,1} (1 - p_A^{2,2,1}) (1 - p_E^{2,2,1}) \dots (1 - p_H^{2,2,1}) p_A^{2,2,1} (1 - p_E^{2,2,1}) (1 - p_H^{2,2,1}) (1 - p_A^{2,2,1}) p_E^{2,2,1} (1 - p_H^{2,2,1}) (1 - p_A^{2,2,1}) (1 - p_E^{2,2,1}) \\ p_H^{2,2,2} p_A^{2,2,2} p_E^{2,2,2} \dots p_H^{2,2,2} (1 - p_A^{2,2,2}) (1 - p_E^{2,2,2}) \dots (1 - p_H^{2,2,2}) p_A^{2,2,2} (1 - p_E^{2,2,2}) (1 - p_H^{2,2,2}) (1 - p_A^{2,2,2}) p_E^{2,2,2} (1 - p_H^{2,2,2}) (1 - p_A^{2,2,2}) (1 - p_E^{2,2,2}) \end{bmatrix} \quad (15)$$

**Theorem 1** (see [22]). *In  $K \times N$  repeated games, both players have the same structure of payoff matrix in each game stage. If a game player has adopted a short-memory strategy, then for this game player, the short-memory strategy can achieve the long-term expected benefits under the long-memory strategy.*

Based on Theorem 1, this paper presents a single-stage game process as a Markov chain with a single-memory cycle. In this paper,  $\mathbf{n}(t) = (n_1(t), n_2(t), n_3(t))$  are the behavioral states of the  $t_{\text{th}}$  stage of the game,  $\mathbf{Prob}(\bullet)$  is the probability that each player in the single-stage game takes each behavior state, and  $\mathbf{k} = (k_1, k_2, k_3)$ , then

$$p_{\text{H}}^{\mathbf{k}} = \mathbf{Prob}(n_1(t+1) = 1 \mid \mathbf{n}(t) = \mathbf{k}), \quad \forall \mathbf{k} \in \mathbf{S} \quad (16)$$

this formula represents the probability of HEMS taking action 1 in stage  $t+1$  game in the case that HEMS, RTPA, and ES, respectively, adopt behavior states  $k_1, k_2, k_3$  in the  $t_{\text{th}}$  stage of the game. Similarly, the following equations represent the probability that RTPA and ES will take action 1 in stage  $t+1$ .

$$p_{\text{A}}^{\mathbf{k}} = \mathbf{Prob}(n_2(t+1) = 1 \mid \mathbf{n}(t) = \mathbf{k}), \quad \forall \mathbf{k} \in \mathbf{S} \quad (17)$$

$$p_{\text{E}}^{\mathbf{k}} = \mathbf{Prob}(n_3(t+1) = 1 \mid \mathbf{n}(t) = \mathbf{k}), \quad \forall \mathbf{k} \in \mathbf{S} \quad (18)$$

According to (16), (17), and (18), the state transition matrix of the Markov chain can be obtained as an  $8 \times 8$  matrix as shown formula (15).

$$\boldsymbol{\pi}^{\text{T}} \mathbf{f} = \begin{bmatrix} -1 + P_{\text{H}}^{1,1,1} P_{\text{A}}^{1,1,1} P_{\text{E}}^{1,1,1} & \dots & -1 + P_{\text{H}}^{1,1,1} & \dots & -1 + P_{\text{A}}^{1,1,1} & -1 + P_{\text{E}}^{1,1,1} & f_1 \\ P_{\text{H}}^{1,1,2} P_{\text{A}}^{1,1,2} P_{\text{E}}^{1,1,2} & \dots & -1 + P_{\text{H}}^{1,1,2} & \dots & -1 + P_{\text{A}}^{1,1,2} & -1 + P_{\text{E}}^{1,1,2} & f_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ P_{\text{H}}^{2,2,1} P_{\text{A}}^{2,2,1} P_{\text{E}}^{2,2,1} & \dots & P_{\text{H}}^{2,2,1} & \dots & P_{\text{A}}^{2,2,1} & P_{\text{E}}^{2,2,1} & f_7 \\ P_{\text{H}}^{2,2,2} P_{\text{A}}^{2,2,2} P_{\text{E}}^{2,2,2} & \dots & P_{\text{H}}^{2,2,2} & \dots & P_{\text{A}}^{2,2,2} & P_{\text{E}}^{2,2,2} & f_8 \end{bmatrix} \quad (22)$$

Note  $\mathbf{m}_{\text{H}}^{\text{T}} = (-1 + P_{\text{H}}^{1,1,1}, -1 + P_{\text{H}}^{1,1,2}, -1 + P_{\text{H}}^{1,2,1}, -1 + P_{\text{H}}^{1,2,2}, P_{\text{H}}^{1,1,1}, P_{\text{H}}^{1,1,2}, P_{\text{H}}^{1,2,1}, P_{\text{H}}^{1,2,2})$ ; obviously,  $\mathbf{m}_{\text{H}}^{\text{T}}$  is independent of the behavioral state taken by RTPA and ES. Assuming  $\mathbf{m}_{\text{H}} = \mathbf{f}$ , then (22) equals zero. Assuming that  $\mathbf{f} = a\mathbf{D} + b\mathbf{I}$ , where  $\mathbf{I}$  is a column vector with all elements of 1, then (22) can be simplified to

$$a\mu_{\text{D}} + b = 0 \quad (23)$$

thereto,  $a, b$  is a nonzero real number.

Combining the equations  $\mathbf{m}_{\text{H}} = \mathbf{f}$  and (23), we can get the probability that the HEMS will take the active state in different situations as follows:

$$p_{\text{H}}^{1,1,1} = 1 + \left(1 - \frac{D_{1,1,1}}{\mu_{\text{D}}}\right) b,$$

Let  $\pi_{u,v,w}$  be the probability that the HEMS, RTPA, and ES take the behavioral states  $u, v, w$ , and  $\mathbf{D}^{\text{T}} = (D_{1,1,1}, D_{1,1,2}, D_{1,2,1}, D_{1,2,2}, D_{2,1,1}, D_{2,1,2}, D_{2,2,1}, D_{2,2,2})$ , and then the Markov chain has a uniquely stationary distribution of  $\boldsymbol{\pi}^{\text{T}} = (\pi_{1,1,1}, \pi_{1,1,2}, \pi_{1,2,1}, \pi_{1,2,2}, \pi_{2,1,1}, \pi_{2,1,2}, \pi_{2,2,1}, \pi_{2,2,2})$  and  $\boldsymbol{\pi}^{\text{T}} \mathbf{M} = \boldsymbol{\pi}^{\text{T}}$ . After the multistage repeated game, we can get the long-term expected benefits of electrical customers

$$\mu_{\text{D}} = \boldsymbol{\pi}^{\text{T}} \mathbf{D} \quad (19)$$

Note  $\hat{\mathbf{M}} = \mathbf{M} - \mathbf{E}$ , then

$$\boldsymbol{\pi}^{\text{T}} \hat{\mathbf{M}} = 0 \quad (20)$$

According to Cramer's Rule, we can draw the following:

$$\text{adj}(\hat{\mathbf{M}}) \hat{\mathbf{M}} = \det(\hat{\mathbf{M}}) \mathbf{E} = 0 \quad (21)$$

From (20) and (21), We can conclude that  $\det(\hat{\mathbf{M}}) = 0$ , then for any vector  $\mathbf{f}^{\text{T}} = (f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8)$  there exists the following formula:

$$p_{\text{H}}^{1,1,2} = 1 + \left(1 - \frac{D_{1,1,2}}{\mu_{\text{D}}}\right) b$$

$$p_{\text{H}}^{1,2,1} = 1 + \left(1 - \frac{D_{1,2,1}}{\mu_{\text{D}}}\right) b,$$

$$p_{\text{H}}^{1,2,2} = 1 + \left(1 - \frac{D_{1,2,2}}{\mu_{\text{D}}}\right) b$$

$$p_{\text{H}}^{2,1,1} = \left(1 - \frac{D_{2,1,1}}{\mu_{\text{D}}}\right) b,$$

$$p_{\text{H}}^{2,1,2} = \left(1 - \frac{D_{2,1,2}}{\mu_{\text{D}}}\right) b$$

$$p_{\text{H}}^{2,2,1} = \left(1 - \frac{D_{2,2,1}}{\mu_{\text{D}}}\right) b,$$

TABLE 1: Single-stage game payoff matrix.

HEMS	ES			
	$n_3 = 1$		$n_3 = 2$	
	$n_2 = 1$	$n_2 = 2$	$n_2 = 1$	$n_2 = 2$
$n_1 = 1$	$D_{1,1,1}$	$D_{1,2,1}$	$D_{1,1,2}$	$D_{1,2,2}$
$n_1 = 2$	$D_{2,1,1}$	$D_{2,2,1}$	$D_{2,1,2}$	$D_{2,2,2}$

TABLE 2: Single-stage game load demand matrix.

HEMS	ES			
	$n_3 = 1$		$n_3 = 2$	
	$n_2 = 1$	$n_2 = 2$	$n_2 = 1$	$n_2 = 2$
$n_1 = 1$	$D_{i,N}(i) + D_{ES}(i) + D_{RTPA}(i) + D_{HEMS}(i)$	$D_{i,N}(i) + D_{ES}(i) + D_{HEMS}(i)$	$D_{i,N}(i) + D_{RTPA}(i) + D_{HEMS}(i)$	$D_{i,N}(i) + D_{HEMS}(i)$
$n_1 = 2$	$D_{i,N}(i) + D_{ES}(i) + D_{RTPA}(i)$	$D_{i,N}(i) + D_{ES}(i)$	$D_{i,N}(i) + D_{RTPA}(i)$	$D_{i,N}(i)$

$$P_H^{2,2,2} = \left(1 - \frac{D_{2,2,2}}{\mu_D}\right) b \quad (24)$$

**4.2. Analysis of Multiperson Zero-Determinant Strategy.** In a single-stage game, RTPA injects false information of electricity price into the smart meter during the period  $\Delta T$ , so that the indoor smart appliances generate additional load demands. RTPA aims to make the total load demand  $D(i)$  of electrical customers larger than the total load demand  $D_{i,N}(i)$  under steady state. HEMS optimizes the electrical customer's electricity plan by optimized scheduling module and turns off some dispatchable loads; if necessary, HEMS can sacrifice partial comfort of electrical customers and turn off some nondispatchable loads. HEMS fully incorporates a MPZDS to minimize the impact of attacks so that the total load demand  $D(i)$  of electrical customers is close to the total load demand  $D_{i,N}(i)$  under steady state. The value of  $\mathbf{D}$  is shown in Table 2.

**Theorem 2** (see [23, 24]). *In the repeated game, assuming that the game players  $N \geq 2$ ,  $U_{r,\min}$  and  $U_{r,\max}$  are, respectively, the maximum and minimum values of row  $r$  in the  $N \times 2$  repeated game payoff matrix, where  $r = 1, 2$ , then*

$$\begin{aligned} U_{r,\min} &= \min(U_{r,1}, \dots, U_{r,2}^{N-1}) \\ U_{r,\max} &= \max(U_{r,1}, \dots, U_{r,2}^{N-1}) \end{aligned} \quad (25)$$

If there exists  $k \min, k \max \in \{1, 2\}$  which makes  $U_{k \max, \max} \leq U_{k \min, \min}$  satisfied, the game player  $i$  can ignore the behavior strategy of other game players and control the long-term expected benefit of game player  $i$  in the interval  $[U_{k \max, \max}, U_{k \min, \min}]$ .

Let  $D_{r,\max}$  and  $D_{r,\min}$  be the maximum and minimum values of row  $u$  in the single-stage game load demand matrix.

Combined with the conclusions of (7)–(14), the maximum and minimum values of row  $u$  of  $\mathbf{D}$  are

$$\begin{aligned} D_{1,\max} &= D_{i,N} + D_{ES}(i) + D_{RTPA}(i) + D_{HEMS}(i) \\ D_{1,\min} &= D_{i,N} + D_{HEMS}(i) \\ D_{2,\max} &= D_{i,N} + D_{ES}(i) + D_{RTPA}(i) \\ D_{2,\min} &= D_{i,N} \end{aligned} \quad (26)$$

In formula (26),  $D_{1,\max} \leq D_{2,\min}$  and  $D_{2,\max} \geq D_{1,\min}$ ; since  $D_{1,\max} \leq D_{2,\min}$ , we can get the long-term expected benefit  $\mu_D$  of the game based on Theorem 2, so its value range is

$$\begin{aligned} [D_{1,\max}, D_{2,\min}] \\ = D_{i,N} + [D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i), 0] \end{aligned} \quad (27)$$

Considering the upstream and downstream boundary values of long-term expected benefit  $\mu_D$ ,  $\mu_D$  can be expressed as

$$\mu_D = D_{i,N} + \alpha [D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)] \quad (28)$$

where  $\alpha$  is the excitation factor and  $0 \leq \alpha \leq 1$ . The value of  $\mu_D$  increases as  $\alpha$  increases and approaches the value of  $D_{i,N} + D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)$ , which indicates that HEMS is more active in the process of stable control, which can make the total load demand of electrical customers approach the stable state.

The value of  $b$  in formula (24) is

$$\max\left(\frac{-\mu_D}{\mu_D - D_{1,\min}}, \frac{\mu_D}{\mu_D - D_{2,\max}}\right) \leq b < 0 \quad (29)$$

Combining (26), (28), and (29), we can get  $\mu_D - D_{1,\min} \geq D_{2,\max} - \mu_D$  when  $\alpha \geq 1/2$ , then  $b_{\min} \leq b < 0$ , where  $b_{\min} = (\mu_D / \mu_D - D_{2,\max})$ , then

$$\begin{aligned} b_{\min} \\ = \frac{D_{i,N} + D_{ES}(i) + D_{RTPA}(i) + D_{HEMS}(i)}{\alpha D_{HEMS}(i) + (\alpha - 1) D_{ES}(i) + (\alpha - 1) D_{RTPA}(i)} \end{aligned} \quad (30)$$

Taking into account the upstream and downstream boundary values of  $b$ , note  $b = \beta b_{\min}$ , then

$$b = \beta \times \frac{D_{i,N} + D_{ES}(i) + D_{RTPA}(i) + D_{HEMS}(i)}{\alpha D_{HEMS}(i) + (\alpha - 1) D_{ES}(i) + (\alpha - 1) D_{RTPA}(i)} \quad (31)$$

where  $\beta$  is the control factor and  $0 \leq \beta \leq 1$ . The probability that HEMS will take a stable control action increases as  $\beta$  increases.

Assuming that the behavioral states of HEMS, RTPA, and ES are  $u, v, w$ , then  $p_H^{u,v,w}$  is the probability that HEMS will take action 1 in stage  $t + 1$  game when HEMS takes action state  $u$ , RTPA takes actions state  $v$ , and ES takes actions state  $w$  under the current game occurring in the period  $i - 1$ , the  $t_{th}$  stage of the game. Combining (24), (28), and (31), the probability of HEMS adopting stable control behavior in the  $i_{th}$  period is shown in formula (32) and (33).

$$p_H^{1,1,2} = \beta \times \frac{[(\alpha - 1) D_{HEMS}(i) + (\alpha - 1) D_{RTPA}(i) + \alpha D_{ES}(i)]}{[\alpha D_{HEMS}(i) + (\alpha - 1) D_{RTPA}(i) + (\alpha - 1) D_{ES}(i)]} \times \frac{[D_{i,N} + D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]}{\{D_{i,N} + \alpha [D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]\}} + 1$$

$$p_H^{1,1,1} = p_H^{1,1,2} - \beta \times \frac{D_{ES}(i)}{[\alpha D_{HEMS}(i) + (\alpha - 1) D_{RTPA}(i) + (\alpha - 1) D_{ES}(i)]} \times \frac{[D_{i,N} + D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]}{\{D_{i,N} + \alpha [D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]\}} \quad (32)$$

$$p_H^{1,2,2} = \beta \times \frac{[(\alpha - 1) D_{HEMS}(i) + \alpha D_{RTPA}(i) + \alpha D_{ES}(i)]}{[\alpha D_{HEMS}(i) + (\alpha - 1) D_{RTPA}(i) + (\alpha - 1) D_{ES}(i)]} \times \frac{[D_{i,N} + D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]}{\{D_{i,N} + \alpha [D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]\}} + 1$$

$$p_H^{1,2,1} = p_H^{1,2,2} - \beta \times \frac{D_{ES}(i)}{[\alpha D_{HEMS}(i) + (\alpha - 1) D_{RTPA}(i) + (\alpha - 1) D_{ES}(i)]} \times \frac{[D_{i,N} + D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]}{\{D_{i,N} + \alpha [D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]\}}$$

$$p_H^{2,1,2} = \beta \times \frac{[\alpha D_{HEMS}(i) + (\alpha - 1) D_{RTPA}(i) + \alpha D_{ES}(i)]}{[\alpha D_{HEMS}(i) + (\alpha - 1) D_{RTPA}(i) + (\alpha - 1) D_{ES}(i)]} \times \frac{[D_{i,N} + D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]}{\{D_{i,N} + \alpha [D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]\}} + 1$$

$$\times \frac{[D_{i,N} + D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]}{\{D_{i,N} + \alpha [D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]\}}$$

$$p_H^{2,1,1} = p_H^{2,1,2} - \beta \times \frac{D_{ES}(i)}{[\alpha D_{HEMS}(i) + (\alpha - 1) D_{RTPA}(i) + (\alpha - 1) D_{ES}(i)]} \times \frac{[D_{i,N} + D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]}{\{D_{i,N} + \alpha [D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]\}}$$

$$p_H^{2,2,2} = \beta \times \frac{\alpha [D_{HEMS}(i) + D_{RTPA}(i) + D_{ES}(i)]}{[\alpha D_{HEMS}(i) + (\alpha - 1) D_{RTPA}(i) + (\alpha - 1) D_{ES}(i)]} \times \frac{[D_{i,N} + D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]}{\{D_{i,N} + \alpha [D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]\}}$$

$$p_H^{2,2,1} = p_H^{2,2,2} - \beta \times \frac{D_{ES}(i)}{[\alpha D_{HEMS}(i) + (\alpha - 1) D_{RTPA}(i) + (\alpha - 1) D_{ES}(i)]} \times \frac{[D_{i,N} + D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]}{\{D_{i,N} + \alpha [D_{ES}(i) + D_{HEMS}(i) + D_{RTPA}(i)]\}} \quad (33)$$

If  $\alpha = 1$ , indicating that HEMS is the most active in the process of stable control, then the probability of HEMS behavior state is

$$p_H^{1,1,1} = 1,$$

$$p_H^{1,1,2} = p_H^{1,1,1} + \beta \times \frac{D_{ES}(i)}{D_{HEMS}(i)}$$

$$p_H^{1,2,1} = 1 + \beta \times \frac{D_{RTPA}(i)}{D_{HEMS}(i)},$$

$$p_H^{1,2,2} = p_H^{1,2,1} + \beta \times \frac{D_{ES}(i)}{D_{HEMS}(i)}$$

$$p_H^{2,1,1} = \beta,$$

$$p_H^{2,1,2} = p_H^{2,1,1} + \beta \times \frac{D_{ES}(i)}{D_{HEMS}(i)}$$

$$p_H^{2,2,1} = \beta \times \frac{D_{HEMS}(i) + D_{RTPA}(i)}{D_{HEMS}(i)},$$

$$p_H^{2,2,2} = p_H^{2,2,1} + \beta \times \frac{D_{ES}(i)}{D_{HEMS}(i)} \quad (34)$$

Equation (34) shows that if RTPA is active in the current game stage, the probability of HEMS taking stable control action will increase in the next round of game stage; if ES is active, the probability that HEMS taking stable control action will also increase in the next round of game stage. This behavioral characteristic shows that if RTPA is active, HEMS can take corresponding stable control behavior, and ES can



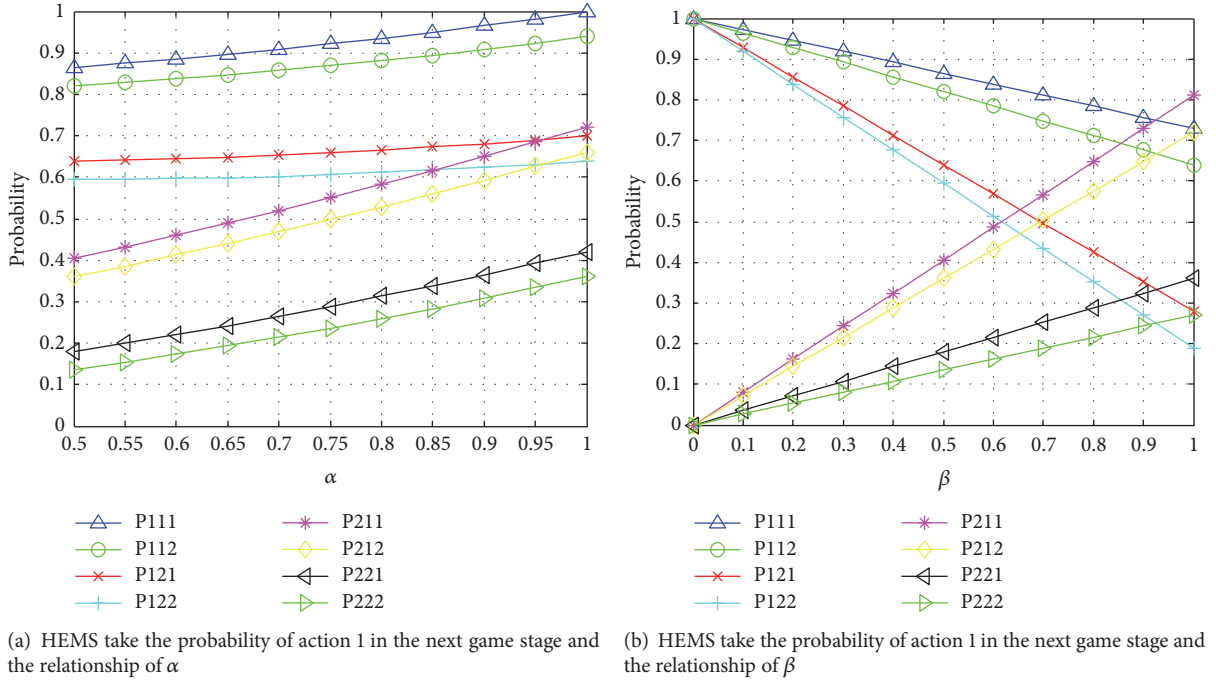


FIGURE 1: HEMS take the probability of action 1 in the next game stage.

optimize the behavior selection of HEMS to a certain extent in order to reduce the impact of the increase of total load demand caused by RTPA.

## 5. Experiment Analysis

**5.1. The Probability of HEMS Being Active.** In this paper, the experimental hardware platform is AMD A8-5550M quad-core processor, 2.1GHz frequency, 6GB memory, software platform: Windows 10 operating system, Eclipse with PyDev integrated simulation tools, and MATLAB R2014a experimental environment; among them, Eclipse is 4.6.2; Python is 3.5.2. The experimental data uses the real-time price and electricity load demand data of the New South Wales of Australian Energy Market on April 6, 2017, and April 7, 2017, thereto, the data of April 6 for the base day data; data of April 7 for the goal day data. The significance of the data selection is that the weather and other environmental conditions and the electricity consumption habits of electrical customers are almost at the same in two adjacent days; the load demand of electrical customers is protected from weather conditions and electricity consumption habits of electrical customers. The coefficient of price elasticity of electricity demand is calculated from (6).

In this paper, we assume that the update cycle of real-time price is 5min, and one day is divided into 288 periods. As a result, the daily load demand curve can be divided into three periods: low period, flat period, and peak period, as shown in Table 3.

It is assumed that the real-time price market is in a normal state during the flat period, that is, when  $198 \leq i \leq 209$ ; assume that an attacker intercepts the real-time price signal

TABLE 3: Segment of real-time price time.

Periods	The start and end time of period	Duration/h
Low	00:00-06:00, 22:00-24:00	8
Flat	06:00-07:00, 11:00-18:00, 20:00-22:00	10
Peak	07:00-11:00, 18:00-20:00	6

from the electricity suppliers and tamper with the real-time price information when  $i = 210$ ; assume that HEMS, RTPA, and ES are rational. Then under the ES scheduling, and eight kinds of game situations, HEMS take the probability (formula (32) and (33)) of action 1 (active) in the next game stage and the relationship of  $\alpha$ ,  $\beta$  shown in Figures 1 and 2, where (P111, P112, P121, P122, P211, P212, P221, P222) :=  $(p_H^{1,1,1}, p_H^{1,1,2}, p_H^{1,2,1}, p_H^{1,2,2}, p_H^{2,1,1}, p_H^{2,1,2}, p_H^{2,2,1}, p_H^{2,2,2})$ .

As can be seen from Figure 1(a), the probability that HEMS will take action 1 in the next game stage will increase as  $\alpha$  increases, no matter what kind of behavior of the attackers and electricity suppliers are in the previous game stage. If the HEMS is active during the previous game stage, the probability that the HEMS will take action 1 in the next game stage is significantly greater than the probability that the HEMS was idle during the previous game stage. If the RTPA is active during the previous game stage, the probability that HEMS will take action 1 in the next game stage is also greater than the probability that RTPA was idle during the previous game stage, the reason is that in the game process, HEMS has a memory depth of 1 under combining with a MPZDS, which can memorize the behavioral state of the three game players in the previous game stage to optimize the behavior selection of HEMS in the next game stage. In

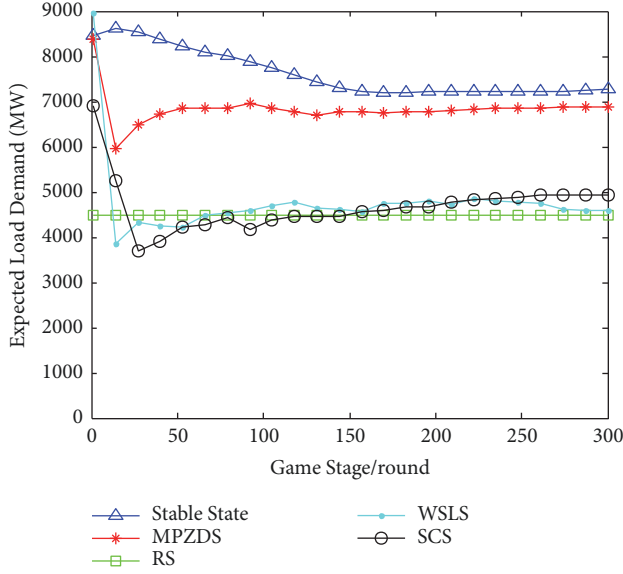


FIGURE 2: The expected load demands of the four strategies vary with the number of different game stages.

the previous game stage, if the behavior states of HEMS and RTPA remain unchanged, when the ES is active, the probability that HEMS takes action 1 in the next game stage is greater than the probability that ES was idle state in the previous game stage, the reason is that the dispatching effect of ES on load demand of electrical customers aims to make the power supply and demand balance, and this shows that ES can optimize the behavior selection of HEMS in the next game stage by participating in game. Similarly, as can also be seen from Figure 1(b), if the RTPA is active in the previous game stage, the probability that the HEMS will take action 1 in the next game stage is significantly greater than the probability that RTPA was idle in the previous game stage. Although the probability that HEMS takes action 1 in the next game stage decreases slightly when the HEMS was active in the previous game stage, but the mean value of the probability which is still greater or equal to the average of the HEMS was idle state in the previous game stage, and  $p_H^{1,1,1} + p_H^{2,2,2} = 1$ ,  $p_H^{1,1,2} + p_H^{2,2,1} = 1$ ,  $p_H^{1,2,1} + p_H^{2,1,2} = 1$ , and  $p_H^{1,2,2} + p_H^{2,1,1} = 1$ , and the relations of  $p_H^{1,1,1}$  and  $p_H^{2,2,2}$ ,  $p_H^{1,1,2}$  and  $p_H^{2,2,1}$ ,  $p_H^{1,2,1}$  and  $p_H^{2,1,2}$ , and  $p_H^{1,2,2}$  and  $p_H^{2,1,1}$  are shown to be complementary.

**5.2. Expected Load Demand Reduction Rate of Electrical Customers.** Assuming that the number of repeated game phases is 300 round and the game period is  $\Delta T = 5\text{min}$ , we, respectively, simulate the four kinds of single-memory strategies under different game stage numbers to get the game sequence of the corresponding strategies and calculate the expected load demand. The four kinds of single-memory strategies are Multiperson Zero-Determinant Strategies (MPZDS), Random Strategies (RS), Win-Stay-Lose-Shift Strategies [25] (WLS), and Stochastic Cooperator Strategies [26] (SCS). Figure 2 shows the comparison between the four strategies and the expected load demand in stable state under different game stage numbers.

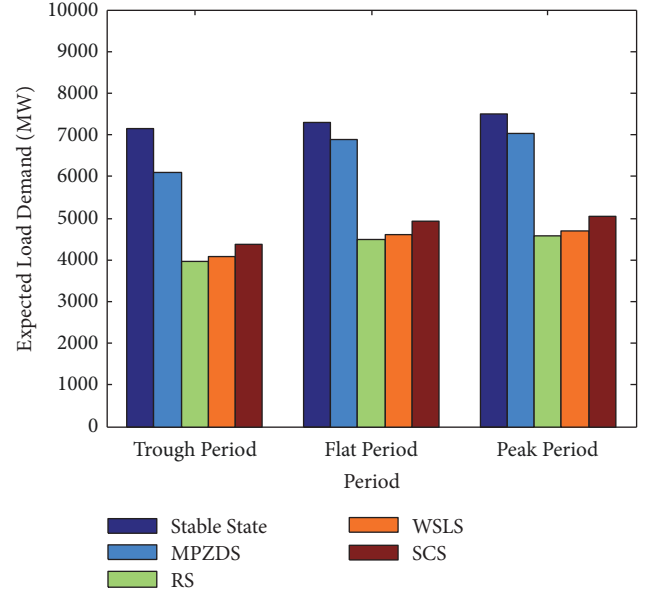


FIGURE 3: Comparing expected load demands of different periods.

As can be seen from Figure 2, expected load demand reduction rate of MPZDS was 5.4%, expected load demand reduction rate of RS was 38.5%, expected load demand reduction rate of WLS was 36.9%, and expected load demand reduction rate of SCS was 32.2% compared with the expected load demand of stable state. The expected load demand of MPZDS is closer to the expected load demand under stable state, because HEMS takes the behavioral state “1” with a high probability in combination with MPZDS under the condition of RTPA, to achieve stable control effect of the total load demand of electrical customers, and ES also allocates a certain amount of power to the electrical customers in order to balance the power supply and demand. Electrical customers do not need to turn off schedulable or nonschedulable loads too much while ensuring electricity consumption satisfaction under the MPZDS, but RS, WLS, and SCS greatly reduces the total load demand of electrical customers by shutting off schedulable or nonschedulable loads, although it achieves the goal of preventing RTPA to a certain extent; it also greatly reduces the satisfaction with electricity consumption of electrical customers.

Figure 3 compares the expected load demand of four kinds of single-memory strategy and stable state under RTPA which occurs at different periods of time. Among them, the trough hours take  $i = 272$  (22: 00-24: 00), flat hours take  $i = 210$  (11: 00-18: 00), and peak hours take  $i = 123$  (07: 00-11: 00). As can be seen from Figure 3, if the RTPA occurs in the trough period, the expected load demand reduction rates of MPZDS, RS, WLS, and SCS are 14.8%, 44.6%, 43.1%, and 39%, respectively; if RTPA occurs in the flat period, the expected load demand reduction rates of MPZDS, RS, WLS, and SCS are 5.4%, 38.5%, 36.9%, and 32.2%, respectively; if RTPA occurs in the peak period, the expected load demand reduction rates of MPZDS, RS, WLS, and SCS are 6.2%, 39%, 37.4%, and 32.8%, respectively. From

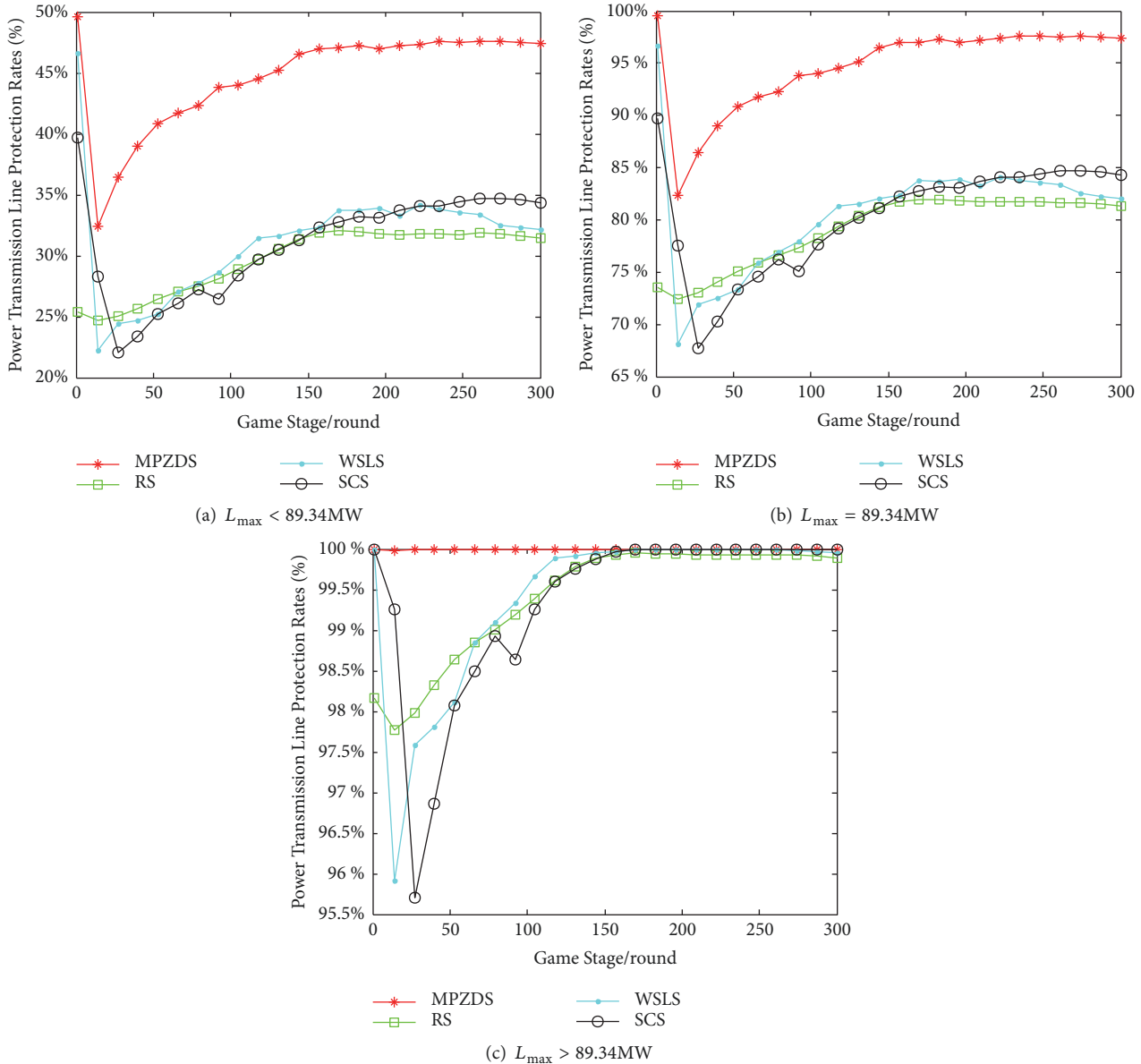


FIGURE 4: Power transmission line protection rates.

the above analysis, we can see that MPZDS has better attack prevention effect than RS, WLS, and SCS in view of RTPA maliciously increasing the total load demand of electrical customers; the reason is that, due to the influence of incentive factor  $\alpha$  and control factor  $\beta$  and the electricity supplier's scheduling, the probability that HEMS is active in the next game stage is greater than that of RS, WLS, and SCS by taking MPZDS. In addition, Figure 3 shows that the expected load demand reduction rate during flat period is lower than the expected load demand reduction rates during peak period and trough period under the premise of HEMS combined with MPZDS; the reason is that the total load demand of electrical customers during flat period is stable and the actual load demand is lower than that during trough period and peak period; HEMS does not need to turn off a large number

of schedulable or nonschedulable loads, so as to achieve a better load demand scheduling solution.

**5.3. Power Transmission Line Protection Rate.** It is assumed that the RTPA occurs during the peak period and the number of power transmission lines is 84. The expected load demand under a stable state is averaged out, and the maximum loadability of each transmission line is 89.34MW, Figures 4(a), 4(b), and 4(c), respectively, when  $L_{max} < 89.34MW$ ,  $L_{max} = 89.34MW$ , and  $L_{max} > 89.34MW$ , the protection rate of the power transmission line in four strategies.

From Figures 4(a), 4(b), and 4(c), if the maximum loadability of each transmission line is less than 89.34MW, the average protection ratios of power transmission lines under

MPZDS, RS, WSLs, and SCS are, respectively, 44.8%, 29.6%, 31.3%, and 31%, and the maximum power transmission line protection rates of MPZDS, RS, WSLs, and SCS are, respectively, 49.6%, 32%, 46.6%, and 39.7%, and the minimum power transmission line protection rates of MPZDS, RS, WSLs, and SCS are, respectively, 32.4%, 24.7%, 22.3%, and 22.1%; if the maximum loadability of each transmission line is equal to 89.34MW, the average protection ratios of power transmission lines under MPZDS, RS, WSLs, and SCS are, respectively, 94.8%, 79.1%, 80.7%, and 80.3%, and the maximum power transmission line protection rates of MPZDS, RS, WSLs, and SCS are, respectively, 99.6%, 82%, 96.6%, and 89.7%, and the minimum power transmission line protection rates of MPZDS, RS, WSLs, and SCS are, respectively, 82.4%, 72.4%, 68.1%, and 67.8%; if the maximum loadability of each transmission line is greater than 89.34MW, the average protection ratios of power transmission lines under MPZDS, RS, WSLs, and SCS are, respectively, 99.9%, 99.4%, 99.4%, and 99.4%, and the maximum power transmission line protection rates of MPZDS, RS, WSLs, and SCS are, respectively, 100%, 99.9%, 100%, and 100%, and the minimum power transmission line protection rates of MPZDS, RS, WSLs, and SCS are, respectively, 99.9%, 97.8%, 95.9%, and 95.7%. Therefore, the protection effect of MPZDS on power transmission lines is obviously better than that of RS, WSLs, and SCS, and the protection rate of power transmission lines increases with the increase of the maximum loadability of single transmission line. The reason is that when HEMS is combined with MPZDS, the active state is taken with a high probability, so that the expected load demand of electrical customers is closer to the expected load demand in stable state; however, RS, WSLs, and SCS significantly reduce the electricity load demand of electrical customers at the expense of sacrificing the electricity satisfaction of electrical customers, so that the expected load demand of electrical customers deviates excessively from the expected load demands under stable state.

## 6. Conclusions

Aiming at the problem that RTPA increases the total load demand of electrical customers, we propose a defensive strategy of real-time price attack based on MPZDS. In order to achieve the goal of stabilizing the total load demand of electrical customers, firstly, according to the game relationship between RTPA, HEMS, and ES, the behavior characteristics of the three players are, respectively, defined. Secondly, we analyze eight kinds of game situations among the three players and get the total load demand of electrical customers. Finally, we combine the MPZDS for safety analysis. Experimental results show that the proposed method of the paper has a lower expected load demand reduction rate and can better protect the safety of power transmission lines. In the future research work, in addition to considering the natural factors such as new energy, it will also consider the impact of collaborative real-time price attack on load demand of electrical customers and electricity market. At the same time, new defense methods such as machine learning or intelligent

judgment will also be considered in defensive strategies of real-time price attack [27, 28].

## Data Availability

The data is available upon request with prior concern to the first author of this paper.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This paper is supported by Hunan Provincial Natural Science Foundation of China, 2019JJ40314.

## References

- [1] J. Wang, Y. Gao, W. Liu, Wu. W, and S. Lim, "An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks," *Computers, Materials & Continua*, vol. 58, no. 3, pp. 711–725, 2019.
- [2] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, and H. Kim, "An intelligent data gathering schema with data fusion supported for mobile sink in WSNs," *International Journal of Distributed Sensor Networks*, vol. 15, no. 3, 2019.
- [3] K. Gu, L. Wang, and B. Yin, "Social community detection and message propagation scheme based on personal willingness in social network," *Soft Computing*, vol. 23, no. 15, pp. 6267–6285, 2019.
- [4] B. Yin and X. Wei, "Communication-efficient data aggregation tree construction for complex queries in IoT applications," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3352–3363, 2019.
- [5] N. Ramos, P. Pereira, and J. Martins, "Smart-meter in power quality," in *Young Engineers Forum*, pp. 42–46, IEEE, 2017.
- [6] R. R. Nejad and S. M. Tafreshi, "A new method for demand response by real-time pricing signals for lighting loads," in *Proceedings of the Power Engineering and Automation Conference (PEAM '12)*, pp. 1–5, IEEE, Wuhan, Hubei, China, 2012.
- [7] S. Mishra, X. Li, A. Kuhnle, M. T. Thai, and J. Seo, "Rate alteration attacks in smart grid," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks, IEEE INFOCOM '15*, pp. 2353–2361, Kowloon, Hong Kong, 2015.
- [8] Y. Liu, M. Reiter, and P. Ning, "False data injection attacks against state estimation in electric power grids," *Acm Transactions on Information System Security*, vol. 14, no. 1, pp. 21–32, 2010.
- [9] G. Liang, J. Zhao, F. Luo, S. R. Weller, and Z. Y. Dong, "A review of false data injection attacks against modern power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1630–1638, 2017.
- [10] S. Lee, J. Kim, and T. Shon, "User privacy-enhanced security architecture for home area network of Smartgrid," *Multimedia Tools & Applications*, vol. 75, no. 20, pp. 1–16, 2016.
- [11] X. Wang, Z. Shi, J. Ren, A. Yang, and L. Sun, "A defensive strategy against delay attacks on real-time pricing in smart grids," *Journal of Beijing University of Posts and Telecommunications*, vol. 51, pp. 116–120, 2015.

- [12] R. Tan, V. Krishna, and D. Yau, "Impact of integrity attacks on real-time pricing in smart grids," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS '13*, pp. 439–450, 2013.
- [13] R. Tan, V. Krishna, and D. Yau, "Integrity attacks on real-time pricing in electric power grids," *Acm Transactions on Information & System Security*, vol. 18, no. 2, pp. 1–33, 2015.
- [14] J. Giraldo, A. Cárdenas, and N. Quijano, "Attenuating the impact of integrity attacks on real-time pricing in smart grids," *Computer Science*, 2014.
- [15] J. Giraldo, A. Cardenas, and N. Quijano, "Integrity attacks on real-time pricing in smart grids: impact and countermeasures," *IEEE Transactions on Smart Grid*, vol. 81, no. 4, pp. 1–9, 2016.
- [16] L. Jia, R. J. Thomas, and L. Tong, "Impacts of malicious data on real-time price of electricity market operations," in *Proceedings of the 45th Hawaii International Conference on System Sciences, HICSS '12*, pp. 1907–1914, IEEE computer society, 2012.
- [17] S. Mishra, X. Li, T. Pan, A. Kuhnle, M. T. Thai, and J. Seo, "Price modification attack and protection scheme in smart grid," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1864–1875, 2017.
- [18] F. Yu, L. Liu, L. Xiao, K. Li, and S. Cai, "A robust and fixed-time zeroing neural dynamics for computing time-variant nonlinear equation using a novel nonlinear activation function," *Neurocomputing*, vol. 350, pp. 108–116, 2019.
- [19] X.-S. Wang, Z.-Q. Shi, J.-J. Ren, A. Yang, and L.-M. Sun, "A defensive strategy against delay attacks on real-time pricing in smart grids," *Journal of Beijing University of Posts and Telecommunications*, vol. 38, supplement 1, pp. 116–120, 2015.
- [20] D. S. Kirschen, G. Strbac, P. Cumperayot, and D. de Mendes, "Factoring the elasticity of demand in electricity prices," *IEEE Transactions on Power Systems*, vol. 15, no. 2, pp. 612–617, 2000.
- [21] H. A. Aalami, M. P. Moghaddam, and G. R. Yousefi, "Demand response modeling considering interruptible/curtailable loads and capacity market programs," *Applied Energy*, vol. 87, no. 1, pp. 243–250, 2010.
- [22] W. H. Press and F. J. Dyson, "Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 109, no. 26, pp. 10409–10413, 2012.
- [23] A. A. Daoud, G. Kesidis, and J. Liebeherr, "Zero-determinant strategies: a game-theoretic approach for sharing licensed spectrum bands," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 11, pp. 2297–2308, 2014.
- [24] X. He, H. Dai, P. Ning, and R. Dutta, "Zero-determinant strategies for multi-player multi-action iterated games," *IEEE Signal Processing Letters*, vol. 23, no. 3, pp. 311–315, 2016.
- [25] S. K. Baek, H. Jeong, C. Hilbe, and M. A. Nowak, "Comparing reactive and memory-one strategies of direct reciprocity," *Scientific Reports*, vol. 6, no. 1, article 25676, 2016.
- [26] C. Adami and A. Hintze, "Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything," *Nature Communications*, vol. 4, no. 4, p. 2193, 2013.
- [27] K. Guo, Z. Liang, Y. Tang, and T. Chi, "SOR: an optimized semantic ontology retrieval algorithm for heterogeneous multimedia big data," *Journal of Computational Science*, vol. 28, pp. 455–465, 2018.
- [28] K. Guo, Z. Liang, R. Shi, C. Hu, and Z. Li, "Transparent learning: an incremental machine learning framework based on transparent computing," *IEEE Network*, vol. 32, no. 1, pp. 146–151, 2018.