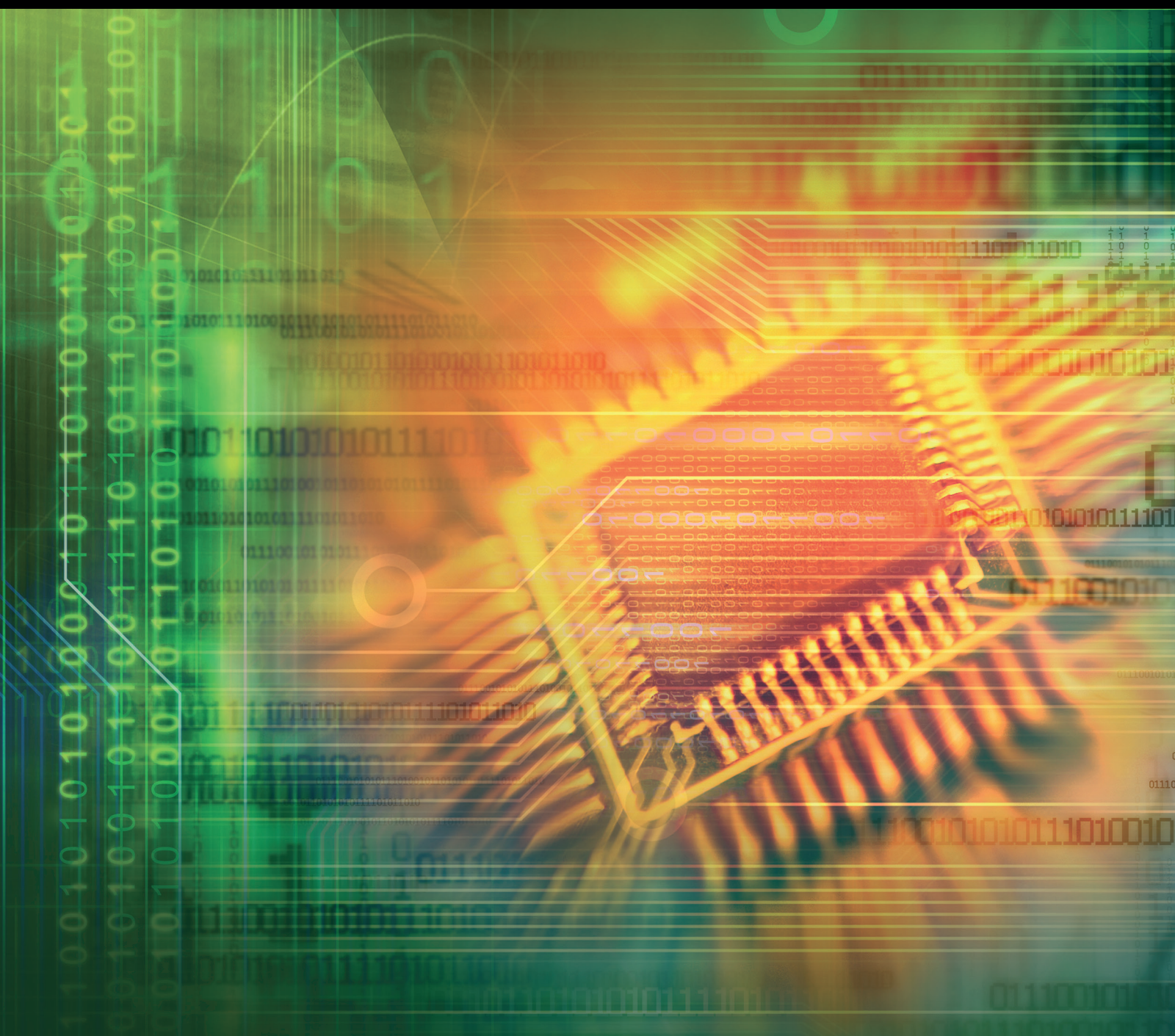
Journal of Electrical and Computer Engineering

Innovations in Communications Security

Guest Editors: Mazdak Zamani, Isao Echizen, and Arash Habibi Lashkari



Innovations in Communications Security

Journal of Electrical and Computer Engineering

Innovations in Communications Security

Guest Editors: Mazdak Zamani, Isao Echizen,
and Arash Habibi Lashkari

Circuits and Systems

Muhammad Abuelma'atti, KSA
Ishfaq Ahmad, USA
Dhamin Al-Khalili, Canada
Ivo Barbi, Brazil
Martin A. Brooke, USA
Tian-Sheuan Chang, Taiwan
M. Jamal Deen, Canada
Andre Ivanov, Canada
Wen B. Jone, USA
H. Kuntman, Turkey
Shen-Iuan Liu, Taiwan
João Antonio Martino, Brazil
Pianki Mazumder, USA
Sing Kiong Nguang, New Zealand
Shun Ohmi, Japan
Mohamed A. Osman, USA
Ping Feng Pai, Taiwan
Marco Platzner, Germany
Dhiraj K. Pradhan, UK
Gabriel Robins, USA
Raj Senani, India
Gianluca Setti, Italy
Nicolas Sklavos, Greece
Ahmed M. Soliman, Egypt
Charles E. Stroud, USA
Ephraim Suhir, USA
Hannu A. Tenhunen, Finland
George S. Tombras, Greece
Spyros Tragoudas, USA
Chi Kong Tse, Hong Kong
Chin-Long Wey, USA
Fei Yuan, Canada

Communications

Sofiène Affes, Canada
Enzo Baccarelli, Italy
Stefano Basagni, USA
Jun Bi, China
René Cumplido, Mexico
Luca De Nardis, Italy
Maria-Gabriella Di Benedetto, Italy
Jocelyn Fiorina, France
Zabih F. Ghassemlooy, UK
K. Giridhar, India
Amoakoh Gyasi-Agyei, Ghana
Yaohui Jin, China
Peter Jung, Germany
Rajesh Khanna, India
Kiseon Kim, Republic of Korea
Tho Le-Ngoc, Canada
Cyril Leung, Canada
Petri Mähönen, Germany
Jit S. Mandeep, Malaysia
Montse Najar, Spain
Adam Panagos, USA
Samuel Pierre, Canada
John N. Sahalos, Greece
Christian B. Schlegel, Canada
Vinod Sharma, India
Iickho Song, Republic of Korea
Ioannis Tomkos, Greece
Chien Cheng Tseng, Taiwan
George Tsoulos, Greece
Jian-Kang Zhang, Canada

Signal Processing

Sos Agaian, USA
Panajotis Agathoklis, Canada
Petar M. Djuric, USA
Karen Egiazarian, Finland
Woon-Seng Gan, Singapore
Zabih F. Ghassemlooy, UK
Martin Haardt, Germany
Jiri Jan, Czech Republic
Chi Chung Ko, Singapore
James Lam, Hong Kong
Riccardo Leonardi, Italy
Cédric Richard, France
William Sandham, UK
Ravi Sankar, USA
Andreas Spanias, USA
Yannis Stylianou, Greece
Ioan Tabus, Finland
Ari J. Visa, Finland
Jar Ferr Yang, Taiwan

Contents

Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2016, Article ID 4743084, 1 page
http://dx.doi.org/10.1155/2016/4743084

Editorial
Innovations in Communications Security

Mazdak Zamani,[1] Isao Echizen,[2] and Arash Habibi Lashkari[3]

[1]*Kean University, Union, NJ, USA*
[2]*National Institute of Informatics, Tokyo, Japan*
[3]*University of New Brunswick, Fredericton, NB, Canada*

Correspondence should be addressed to Mazdak Zamani; zamani.mazdak@gmail.com

Received 12 July 2016; Accepted 12 July 2016

Communications security is the discipline of preventing unauthorized interceptors from accessing telecommunications in an intelligible form, while still delivering content to the intended recipients.

This special issue collects 6 papers from 15 authors belonging to different countries and institutions. It summarizes the most recent developments and ideas on emerging communications security, with particular focus on privacy preserving, key distribution scheme, and digital image watermarking.

In the paper by X. Cao et al. entitled "A Privacy-Preserving Outsourcing Data Storage Scheme with Fragile Digital Watermarking-Based Data Auditing," a privacy-preserving and auditing-supporting outsourcing data storage scheme by using encryption and digital watermarking is proposed which combines digital watermark technology with encryption method for outsourcing data storage.

In the paper by H. Zhong et al. entitled "An Efficient Electronic English Auction System with a Secure On-Shelf Mechanism and Privacy Preserving," a novel electronic English auction system is proposed which uses symmetrical encryptions and fewer ECC operations and improves the security and reduces the system cost.

In the paper by J. N. Luo and M. H. Yang entitled "Analysis and Improvement of Key Distribution Scheme for Secure Group Communication," a scheme to enhance the security of EGK is proposed which guarantees forward and backward secrecy, prevents message modification and forgery during rekeying, requires sender verification, and therefore prevents MITM attacks in group communication.

In the paper by Y. Zolotavkin and M. Juhola entitled "A New Scalar Quantization Method for Digital Image Watermarking," a new scalar QIM-based watermarking method is proposed which provides higher robustness under AWGN and GA compared to other quantization methods. The advantage of the method is due to the introduced procedure of recovery after GA as well as new distribution of quantized samples with IDL.

In the paper by M. C. Tran and Y. Nakamura entitled "Communication Behaviour-Based Big Data Application to Classify and Detect HTTP Automated Software," a new method is proposed to detect and classify auto-ware communication behaviour based on HTTP traffic that uses minor features in HTTP traffic and does not use any signature or content-based feature.

In the paper by J. Wang and J. Liu entitled "The Comparison of Distributed P2P Trust Models Based on Quantitative Parameters in the File Downloading Scenarios," a new method is proposed to compare and evaluate the trust models with quantitative parameters in P2P file downloading scene that evaluated parameters extracted from the trust related concepts and modelled into a hierarchical structure.

Mazdak Zamani
Isao Echizen
Arash Habibi Lashkari

Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2016, Article ID 2017373, 11 pages
http://dx.doi.org/10.1155/2016/2017373

Research Article

Communication Behaviour-Based Big Data Application to Classify and Detect HTTP Automated Software

Manh Cong Tran and Yasuhiro Nakamura

Department of Computer Science, National Defense Academy, 1-10-20 Hashirimizu, Yokosuka, Kanagawa 239-0811, Japan

Correspondence should be addressed to Manh Cong Tran; manhtc@gmail.com

Received 25 December 2015; Revised 25 March 2016; Accepted 26 June 2016

Academic Editor: Jun Bi

HTTP is recognized as the most widely used protocol on the Internet when applications are being transferred more and more by developers onto the web. Due to increasingly complex computer systems, diversity HTTP automated software (autoware) thrives. Unfortunately, besides normal autoware, HTTP malware and greyware are also spreading rapidly in web environment. Consequently, network communication is not just rigorously controlled by users intention. This raises the demand for analyzing HTTP autoware communication behaviour to detect and classify malicious and normal activities via HTTP traffic. Hence, in this paper, based on many studies and analysis of the autoware communication behaviour through access graph, a new method to detect and classify HTTP autoware communication at network level is presented. The proposal system includes combination of MapReduce of Hadoop and MarkLogic NoSQL database along with xQuery to deal with huge HTTP traffic generated each day in a large network. The method is examined with real outbound HTTP traffic data collected through a proxy server of a private network. Experimental results obtained for proposed method showed that promised outcomes are achieved since 95.1% of suspicious autoware are classified and detected. This finding may assist network and system administrator in inspecting early the internal threats caused by HTTP autoware.

1. Introduction

Application layer attacks pose an ever serious threat to network security for years since it always comes after a technically legitimate connection has been established. Because of the flexibility and interoperability of HTTP since everything users need can be found through web services, its based communication is always allowed in most of network. Consequently, HTTP-based automated software (autoware) is blooming in utilizing in reaching Internet users. Unfortunately, besides normal autoware such as for operating system or software updating purpose, in recent years, cyber criminals turn to fully exploit web as a medium of communication environment to lurk a variety of forbidden or illicit activities through spreading HTTP malicious autoware such as fraudulent adware, spyware, or bot. HTTP traffic and autoware can be classified in some categories as in Figure 1:

(i) Human traffic is kind of traffic which is generated by users with their intention when they use normal software such as web browser to access their websites to get information they needed. In this kind of traffic, users clearly understand their accessed sites, who they contact to, and which information they obtain.

(ii) On the other side, the graph presents nonhuman traffic to which users unintentionally have access; they come from autoware. This traffic can be requested from normal software such as antivirus updater, mail client, browser's toolbar, greyware encompasses adware, spyware, joke programs, and malicious software acting as HTTP-based botnet and trojan horses.

Normal autoware can be controlled and beneficial for user; however, since greyware and malicious software penetrate into users' network, they turn out to be internal threats, from which attackers can conduct various types of application layer attacks through these agents, which are really difficult to prevent such as DoS/DDoS, malware distribution, or identity theft. The distinction between malicious and normal activities from HTTP traffic is becoming tougher because the malicious requests merges adequately with legitimate HTTP

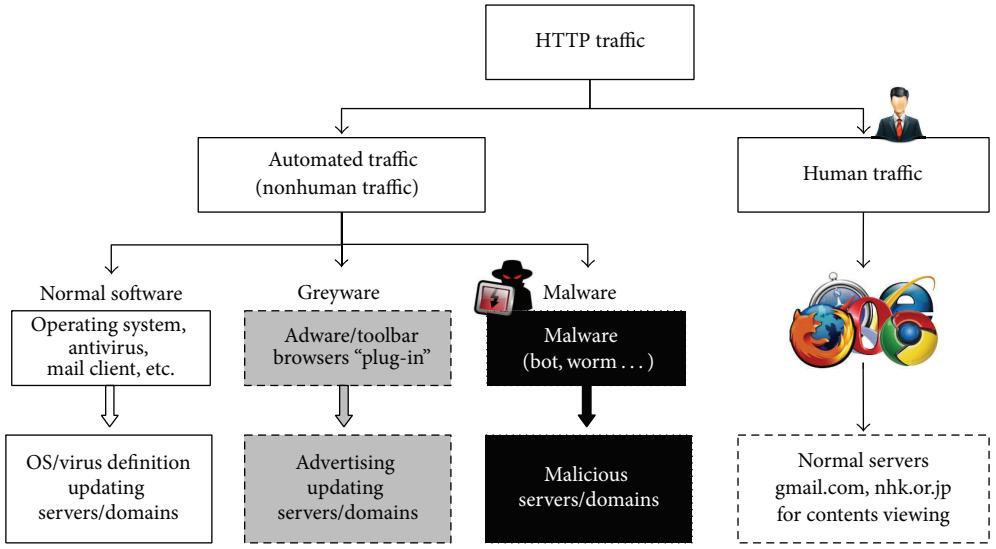


Figure 1: HTTP traffic and automated software categories.

traffic. Furthermore, in a large private network, detection and also classification between types of HTTP autoware traffic are really great challenge when huge requests are generated each day.

To maintain communication, perform updates, or receive commands, all kinds of HTTP-based autoware have common characteristics that they generate repetitively legal traffic and requests to their servers/domains. However, in detail, there are some sophisticated differences in the way of communication behaviour of autoware to their sites. In this paper, based on the analysis and study of autoware communication behaviour, a method in classification and detection of HTTP autoware at network level is proposed. To overcome the issue of handling huge of traffic each day, a big data based system proposal is implemented. In that, a combination between MapReduce of Hadoop [1] and MarkLogic NoSQL database [2] with xQuery supported [3] is suggested for experiment. The method is experimented with real traffic data generated from a university network, and a promised result is archived in classification and detection of malicious HTTP autoware communication.

The remainder of the paper is organized as follows. Related work is discussed in Section 2. In Section 3, features extraction and terminology which included autoware communication behaviour analysis and core terminologies are presented. Section 4 is about detailed description of proposed method which includes algorithms and all components responsible. Section 5 presents applied big data application, the evaluation for proposed method, and experiment results. Finally, conclusion and future work are summarized in Section 6.

2. Related Work

There were a considerable number of techniques which aim to protect users against malware; however, it continues to be a challenging problem. Traditional defense mechanisms such as antivirus (AV) products are the most common content-based malware detection techniques. These types of AV software run on end-user systems and employ signature-based detection to identify variants of known malware. As a consequence, the signature generation and update cycle cause an inherent delay in protecting users against new variants of malware [4]. Additionally, with the aim of limiting AV engines effectiveness, malware authors have developed increasingly sophisticated evasion techniques such as packing and polymorphism, aimed at circumventing detection by AV engines [5, 6]. Oberheide et al. [7] figure many undetected malware binaries by using signature-based techniques, and major AV engines just detect only 30% to 70% of recent malware. As the same content, Rajab et al. [4] show that less than 40% of malicious binaries can be detected by four AV engines in their experiment.

Many botnet detection methods are presented in [8–11]. Ashley [8] has suggested a method for detecting potential HTTP C&C activity based on repeated HTTP connections to a website. According to this, an algorithm is proposed for detecting HTTP polling activity. Lu et al. in [9], using signature-based techniques, propose a hierarchical framework to automatically discover malicious bot on a large-scale Wi-Fi ISP network, in which the network traffic is classified into different application communities by using payload-signature. These signatures were used to separate known traffic from unknown traffic in order to decrease the false alarm rates. Eslahi et al. [10] proposed an approach to reduce the false alarm HTTP botnet detection; in this research, high access rate traffic, which might be other security threats, is filtered out. Basil AsSadhan and Moura [11] proposed a detection method in which it concentrates in C&C communication analysis and find that it exhibits a periodic behaviour. In [11], a method which applied discrete time series is analyzed to examine the aggregate traffic behaviour in order to detect botnet C&C communication channels traffic. These researches [8–11] focus on botnet communication to C&C server, but

actually HTTP threats do not just come from malicious bots but also can be from other types of automated software such as HTTP spyware, adware, or unauthorized applications.

Shin et al. in [12] proposed a framework to detect bot malware at host and network level. At host level, they monitor human-process interactions by using hook technique to capture user mouse and keyboard activities. These hook actions might affect users PC systems. At network level, a simple way to prevent a malware infected PC sending out the information is to prevent all the direct TCP/IP connection from clients. However allowing HTTP protocol is really leaking hole which might be exploited by HTTP malware. In [12], to overcome this issue, they monitored DNS queries to determine C&C server, but actually, many botnets use hacked URL as C&C server. Therefore, the detection method might be insufficient.

Some of approaches use lexical features or keywords extracted from URL and web contents as in [13–16]. However, many other types of malicious web pages are disguised by domain names or URLs like normal website and can harm users PC systems. In this case, lexical or keywords features might be compromised. Bartlett et al. [17] proposed an approach to identify low-rate periodic network traffic and changes in regular communication of autoware. Their research also focuses on many types of autoware and monitor TCP flows to detect, but, in this paper, the target does not just focus only on detecting general types of autoware but also on particular URLs where autoware request to. In addition, our method just collects and processes with basic features of HTTP Traffic at application layer. This will help reduce process cost compared with method used TCP packets features since the number of packets to be processed increases.

3. Features Extraction and Terminology

In this paper, classification and detection method is based on autoware communication behaviour. For that target, by observation of HTTP traffic, autoware communication is analyzed, from which beneficial features are extracted in order to classify and detect various types of autoware. In this section, background related contents and also core terminologies are presented.

3.1. Features Extraction. HTTP traffic from a client consisted of many requests from that client to outside. At application layer, a request includes basic information: IP address of client, full URL, and request method. Full URL's parts contain webpage/server URL and parameter path, as shown in Figure 2. At network level, numerous features are extracted which are made from basic client requests information as follows:

(i) Client IP: source IP address of machine in network which generated requests.

(ii) Request method: main methods of HTTP requests, POST/GET.

(iii) Request date time: date and time when a client sends request.

(iv) Webpage/server URL (shorten as URL): URL requested by a client IP but without parameters' part, as shown in Figure 2. Some normal web servers are hacked and some of their resource paths are exploited as C&C servers. Additionally, parameter parts are easily changed based on the specification of requests content, but actually the functionality of that webpage/server URL, such as C&C server or advertise content update, is the same in each request. Therefore, nonparameter URL is used instead of domain or full URL (will be parameter part), and this matter will help the classification of autoware access behaviour become more detailed and accurate.

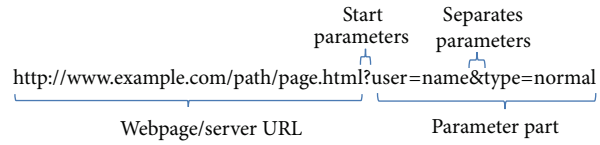


Figure 2: Main parts of URL.

(v) Unique URL: set of unique URLs requested by a client.

(vi) Request interval: break time between two consecutive requests to the same URLs.

(vii) Request count: number of requests to URL from a client in a period of observation data.

(viii) Access time: a period of time in seconds during which a client accessed to URL from the fist request to the end request.

3.2. Access Graph. Access graph presents communication behaviour of a client to a specific URL in a duration of time. It is formed on request interval which are extracted from HTTP traffic. Assuming that $R = \{r_1, r_2, \ldots, r_N\}$ is set of requests from a client to a webpage/server and all r_i have the same webpage/server URL, as described in Figure 2, then access graph G is a sequence which included $N - 1$ items, $G = \{g_1, g_2, \ldots, g_{N-1}\}$, where g_i is a pair of (t_i, d_i), where t_i is timing of request r_{i+1} and d_i is request interval between r_i and r_{i+1}. An access graph is shown as in Figure 3, in which, X-axis is timing of request (except the first request) and Y-axis shows the request interval value in second. An installed or infected autoware client will establish a different access graph for each URL which it sends requests to. For that, this graph can present the behaviour in communication between an autoware to its webpage or server URLs.

3.3. Autoware Communication Behaviour. For keeping communication, update or receive command, all kinds of HTTP-based autoware have common characteristics that they generate repetitively legal traffic and requests to their servers/domains. However, in detail, there are some sophisticated differences in the way of communication behaviour of autoware to their sites.

(i) Malicious HTTP-based bots always follow the PULL style where they connect to their command and control server periodically in order to get the commands and updates. The number of requests from malicious bots are not high as normal autoware (e.g., updater

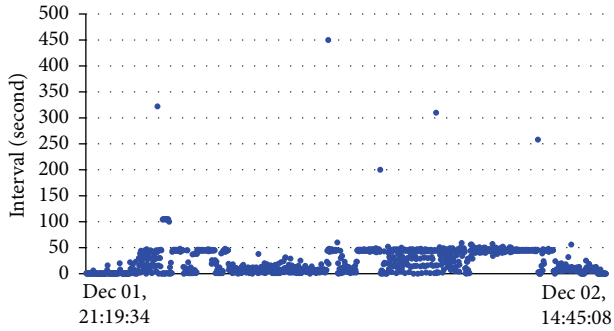


Figure 3: An access graph of a client request to URL.

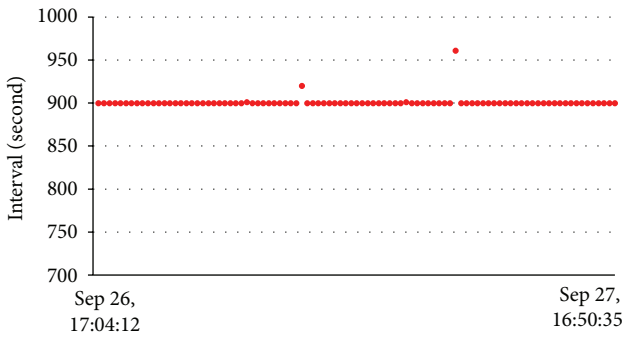


Figure 4: An access graph of HTTP malicious bot.

and downloader) which just generate requests with a long interval than unusual malicious bots [10, 11, 18]. Because interval in communication between a malicious bot to their C&C server is stable, there is almost no variation in their access graph as can be seen in Figure 4 showing the access graph of a bot communication.

(ii) Malicious bots often connect to one control domain and to a specific server resource. Difference with that, unwanted HTTP applications, or greyware, such as annoying adware or spyware, often report back to or request new information from many external resources [17]. Therefore, they keep communicating to their numerous advertising sites or URLs to update pop-up or advertisement and commercial content areas. Autoware will behave the same communication pattern to its URLs if they are requested at the same or approximately equivalent timing so access graph of URLs from a specified autoware is looked similar. In addition, many URLs are requested with the same timing by a specified autoware, so the access duration to these URLs is approximately equal. It means that the first and the last requests timing to these URLs are the same with others. In Figure 5, a sample of two similar access graphs presents the communication from one autoware to two different URLs, and the first and the last requests moment of them are equal.

(iii) On the contrary with autoware, there are no interval or periodic patterns in users' web access; however, in recent years, many sites (e.g., shopping online site or social media webpage) append advertisement path to their sites and use JavaScript or Flash as autoaware part to automatically collect the advertising content as adware or spyware. Therefore parts of users access sites can generate HTTP traffic which act as autoware communication.

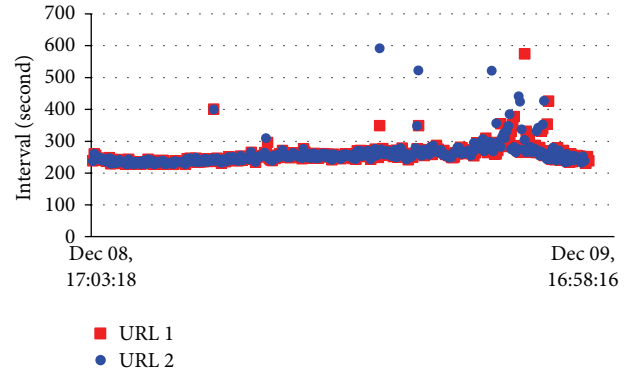


Figure 5: Access graphs from an autoware of a client IP to two different URLs are similar, and the access times of URLs are the equal since both of them are requested from Dec 08, 17:03:18, to Dec 09, 16:58:16.

3.4. Access Graph Distance. As analysis in Section 3.3, even URLs are different; if they are requested by the same autoware then the access graphs look similar, as can be seen in Figure 5. This part proposes a distance to measure the similarity of autoware access behaviour in communication to URLs from a client. The calculation method is based on Modified Hausdorff (MH) distance which is presented in [19].

Assume that there are two access graphs $A = (a_1, \ldots, a_N)$ and $B = (b_1, \ldots, b_M)$. Define that the distance between two points a_i and b_j is calculated as Euclidean distance $d(a_i, b_j) = \|a_i - b_j\|$. From that, distance between point a_i and graph B is defined as $d(a_i, B) = \min_{b_j \in B} \|a_i - b_j\|$. Generalized Hausdorff distance of A and B in [19, 20] is defined as follows:

$$d(A, B) = \frac{1}{N} \sum_{a_i \in A} d(a_i, B). \quad (1)$$

Based on (1), distance between access graphs A and B, which follow by MH distance (MHD), is formed as follows:

$$\mathrm{MHD}(A, B) = \max(d(A, B), d(B, A)). \quad (2)$$

The smaller the MH distance between A and B is, the more A and B are similar to each other.

3.5. Suspicious Score. As described in Section 3.3, malicious bots connect to their command and control server (C&C server) periodically in order to get the commands and updates; therefore, almost there is no large variation in the access graph from malicious bot to its C&C, as can be seen in Figure 4. Based on this analysis, a score is proposed to measure the variation of a access graph, from which it shows suspicious of communication between client to its URL.

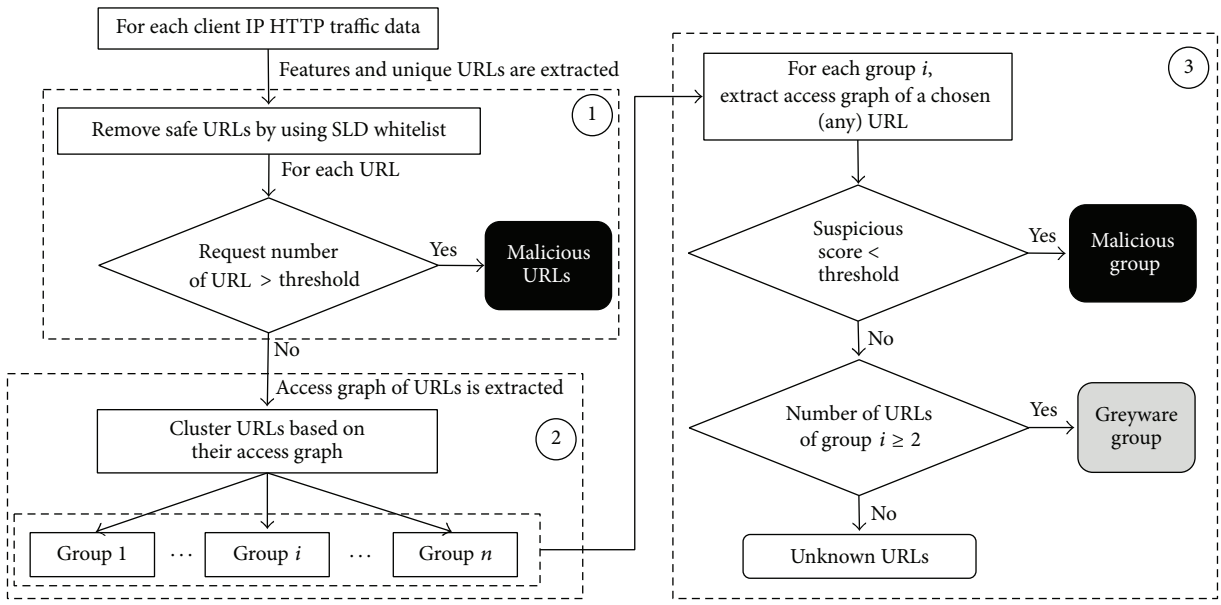


Figure 6: Proposed method diagram in classification and detection of HTTP automated software. Labels of 1, 2, and 3 are preprocesing, clustering, and detection/classification phase, respectively.

Assuming that the access graph of URL S is specified and denoted as $X = (x_1, \ldots, x_N)$, a suspicious score will be defined as *coefficient of variation* of X as follows:

$$\text{Suspicious Score}(X) = \frac{\sigma}{\mu} \tag{3}$$

in which σ and μ are standard deviation and mean of X, respectively. The smaller suspicious score shows that URL is more suspicious.

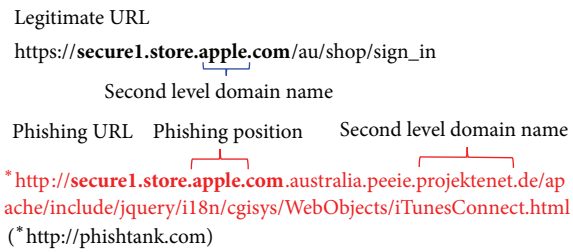


Figure 7: Phishing websites are less consistent with their content when compared with those of legitimate websites.

4. Proposed Method

Based on the autoware communication behaviour which is described in Section 3 and the observation of access graphs in Section 3.1, a classification and detection method, including three phrases, is proposed as in Figure 6; details are as follows.

4.1. Preprocessing Phase. This preprocessing phase is objective to eliminate unnecessary processed data. For each client IP, the one-day HTTP traffic features are extracted and preprocessed; in order to process this phase two methods are applied:

(i) The first one is to filter URLs requests from client IP through a whitelist of second level domain names (SLDN). This filter method is described in [13]; according to that, the tokens in the URLs of phishing websites are less consistent with their content when compared with those of legal websites. An example is illustrated in Figure 7. In this example, the legitimate website contains the brand names *apple* in the SLDN. Even though the phishing website also contains the brand name *apple* in the URL, it is not in the SLDN. Therefore, a domain name which contains a second level domain name which is defined in SLDN whitelist is marked as benign.

(ii) The second method is based on the number of requests to URL from a client IP. Based on the observations number of requests from autoware to URL, it can be seen that suspicious autoware has access many times to URL in a duration of time. Therefore, if the number of requests to URL is too small, it seems not to be requested by an autoware.

Also in this phase, URLs which are requested with extremely fast speed in a duration time will pose a malicious autoware communication; access speed is defined as follows:

$$\text{Access Speed}(\text{URL}_i) = \frac{\text{Request Count}(\text{URL}_i)}{\text{Access Time}(\text{URL}_i)}. \tag{4}$$

In that access time and request count features are described in Section 3.1.

4.2. Clustering Phase. After preprocessing phase, in this phase, remaining URLs will be clustered into number of

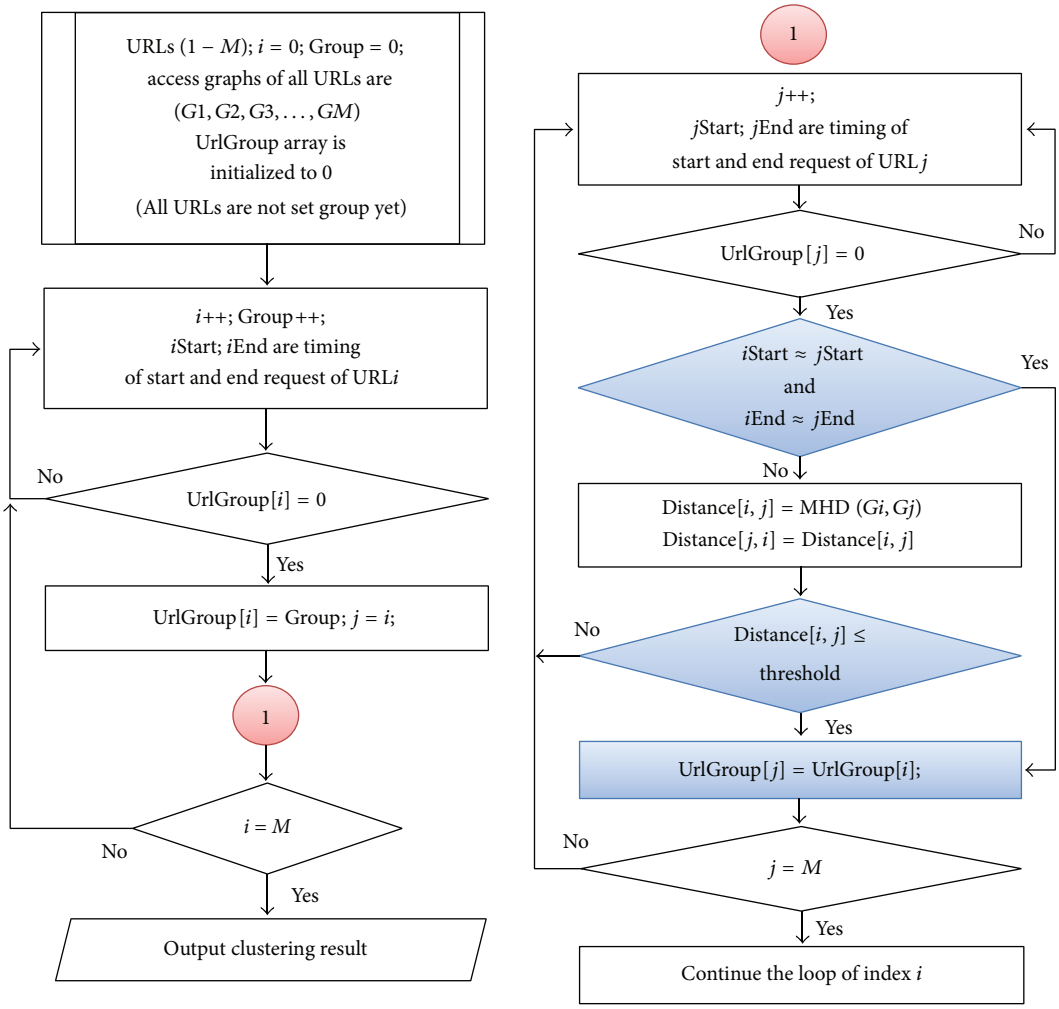


Figure 8: Autoware communication clustering algorithm.

groups based on their characteristics which are presented in Section 3.3. Accordingly, two URLs are of the same group (requested by the same autoware from a client) if they match one of following conditions:

(i) The first and the last request timing to two URLs are approximately the same.

(ii) Based on the similarity of its access graph, MH distance between two access graphs of URLs is calculated; if this distance is small enough, they will be recognized as in the same group.

An algorithm is suggested to decide a group for any two URLs. In order to optimize the consumption processing time of method, the steps of algorithm are proposed in Figure 8. By using a group label array, from this algorithm, distance between all pair of access graphs need not to be calculated. If URL is labeled to a group, it will not need to check group again with other URLs.

4.3. Detection and Classification Phase. The third phase is detection and classification. For each group, a URL (any in the group) is chosen and its access graph is extracted. Then the suspicious score of this URL is calculated; in order to detect whether it is malicious or not a threshold is proposed as 0.04. If the suspicious score is less than or equal to the threshold it is detected as malicious. Finally, remaining groups will be detected by examining the number of unique URLs in group. As analyzed in Section 3, difference with malicious bots, greyware commonly access to various URLs instead of only one server or URL. Therefore, a group having number of unique URLs which are not less than 2 will be marked as greyware groups.

5. Big Data Proposed Framework and Experiment Results

5.1. Big Data Proposed Framework. In this paper, based on above proposed method, big data application is suggested to classify and detect autoware communication. Data for experiment are collected from web proxy of a certain network which served about 2000 clients. Collected data are divided by day saved into logs' file as raw data. Big data application is

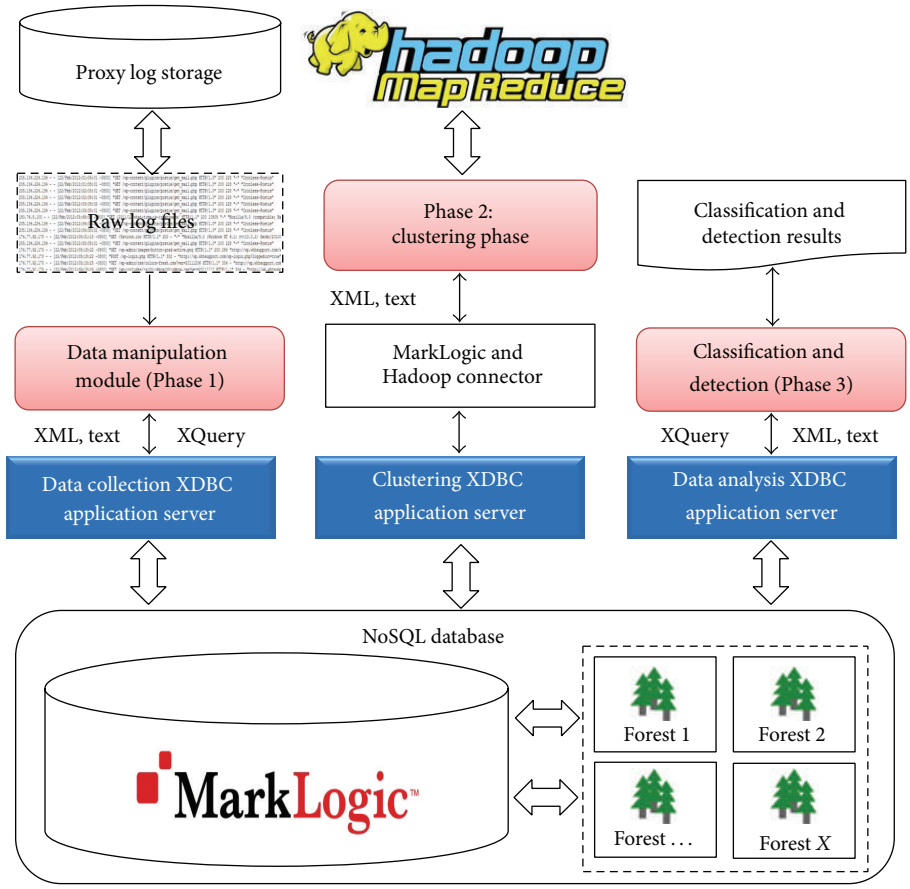


Figure 9: Big data based framework proposal.

composed by combination of MarkLogic database and MapReduce of Hadoop.

As described in [2, 21], MarkLogic is an enterprise NoSQL (Not only Structured Query Language) database which supports a very flexible and convenient XQuery when working with structured and also unstructured data. In addition, it also has had ACID transactions (ACID stands for Atomicity, Consistency, Isolation, and Durability). In a transactional application ACID's properties are necessary so that reads and writes are durably logged to disk and strongly isolated from other transactions. Without this feature, users run the risk of encountering data corruption, stale reads, and inconsistent data. In this framework, XML and text data format are suggested to use because of easily transforming from raw data log file into database.

Hadoop is a great tool to help database application developers and organizations to store and analyze massive amounts of structured and unstructured data from disparate data sources, of which data are too massive to manage effectively with traditional relational databases. Hadoop has become popular because it is designed to cheaply store data in the Hadoop Distributed File System (HDFS) and run large-scale MapReduce jobs for batch analysis. MapReduce is a processing framework that uses a divide-and-conquer paradigm that takes a huge task and breaks it into small parts (Map) and then aggregates the resulting outputs from each part (Reduce). Any large task that can be broken into smaller pieces is a candidate for use with Hadoop [2].

The combination between MarkLogic database and MapReduce of Hadoop in this framework is described in Figure 9, whereby a cluster of MarkLogic is set, and due to optimizing performance in query to database, three XDBC application servers, Data Collection, Clustering, and Data Analysis, are configured along with a number of forests. There are three modules working independently for each phase in Figure 6; details are expressed as follows:

(i) Phase 1 is processed as a part in Data Manipulation Module which will read raw log files, convert to XML and text format, and do the preprocessing before being stored into MarkLogic database via Data Collection Application Server.

(ii) Core functions of heavy Phase 2, Clustering Phase, are implemented according to algorithm in Figure 8 and deployed in the middle part between MarkLogic database and MapReduce of Hadoop. This module will archive results from Phase 1, and URLs are clustered in MapReduce by the distributed processing paradigm. Finally, results of Phase 2 will be returned to MarkLogic database through CLUSTERING XDBC application server. The data exchange between MarkLogic and MapReduce of Hadoop will be

Table 1: Experimental data statistic.

Item	Statistic	Unit	Note
Number of logs	95	PC	Log equals HTTP traffic in a day of IP
Total of requests	13,905,165	Request	All requests of 95 logs
Max requests	479,751	Request	Requests from log
Min requests	22,305	Request	
Average requests	146,370	Request	
Max access time	24	Hour	From the first request to the last request
Min access time	6	Hour	
Average requests	20	Hour	

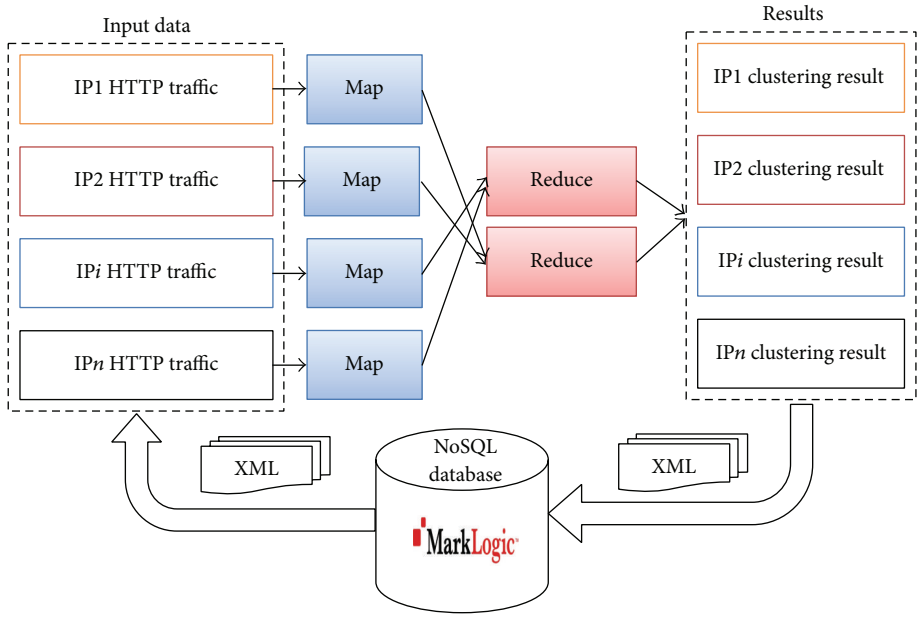


Figure 10: Process flow of clustering phase.

undertaken by a connector. Detailed process flow of this phase is described in Figure 10.

(iii) Classification and Detection Module is implemented for Phase 3, Detection and Classification Phase. It will process the result which is archived from Phase 2 and work with database through Data Analysis Application Server and after that give out processed results.

5.2. Experimental Analysis and Results. Experiment environment is shown in Figure 11; in that free developer licenses of MarkLogic verion 8.0.1 and Hadoop 2.6.0 are used [22]. From this experiment model, HTTP traffic from a university network is captured through a proxy server in separated files which are divided by date and stored in a proxy storage. These logs' raw data files will import to system through Data Manipulation Module as in Figure 9. Denoted log is HTTP traffic of IP in one day, which will be stored in its own directory in MarkLogic; 95 logs' data of clients are extracted, analyzed, and classified through the proposed method. Experiment data is detailed and summarized in Table 1. In that there are two Zeus bots [23] which are installed into a client with difference interval in communication to C&C. All output results are manually checked with the support of VirusTotal online system [24] and McAfee Web Gateway which is installed in experiment network [25].

After preprocessing phase of proposed method described in Figure 6, a set of unique URLs (for logs of each IP) is established with 5621 URLs. In that, there are 14 URLs requested by numerous IPs which are generated with extreme speed over a threshold which is set as 0.8 in this experiment. In Table 2, details of 14 malicious URLs detected by preprocessing phase are summarized. The request per second (access speed) is determined by request count and access time via (4). Based on the characters of malicious autoware which is infected into client IP, the access speed and also communication behaviour to these URLs are determined. For example, as can be seen in Table 2, just in only 0.6 hours, URL2 is requested 80,903 times so it owns highest access speed at 32.98 requests per second. Vise versa, with URL12, it is requested with lowest speed at 0.82 requests per second, 71,004 times in 24 hours; however it is still higher than access speed to other URLs in experimental data. By manually checking the support of [24, 25], all these 14 URLs from domains/web servers contain unwanted software

Table 2: Malicious URLs detected in Phase 1 (preprocessing phase).

Number	Malicious URL	Requests		Access time (h)	Requests per second
		Count	Percent		
1	URL1	237,291	1.71%	2.04	32.38
2	URL2	80,903	0.58%	0.68	32.98
3	URL3	80,032	0.58%	24.00	0.93
4	URL4	303,633	2.18%	10.56	7.98
5	URL5	81,256	0.58%	24.00	0.94
6	URL6	149,966	1.08%	12.53	3.32
7	URL7	496,781	3.57%	4.40	31.39
8	URL8	364,809	2.62%	11.69	8.67
9	URL9	80,761	0.58%	24.00	0.93
10	URL10	297,938	2.14%	16.65	4.97
11	URL11	80,423	0.58%	24.00	0.93
12	URL12	71,004	0.51%	24.00	0.82
13	URL13	80,549	0.58%	24.00	0.93
14	URL14	81,040	0.58%	24.00	0.94
	Total	*2,486,386*	*17.88%*		

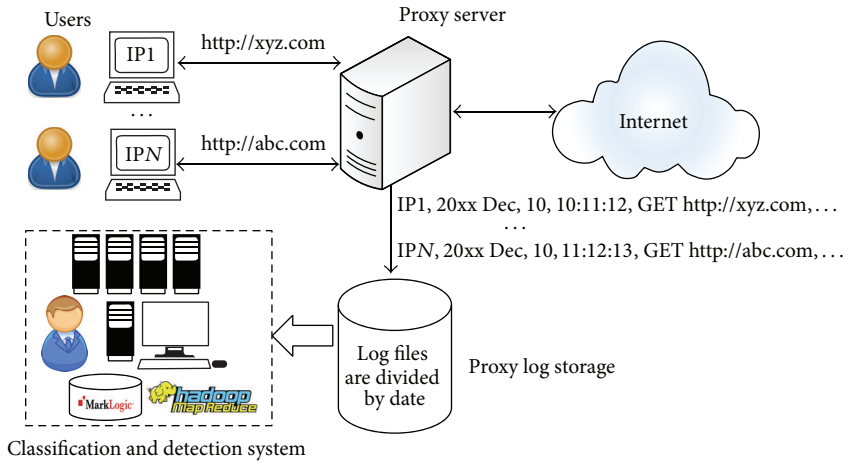


Figure 11: Experiment environment.

and are marked as malicious by many network security companies and software. These 14 URLs are requested 2,486,386 times, and they derive 17.88% of 13,905,165 total requests in experimental data.

Remaining 5607 URLs are classified in 673 groups in which 393 groups which contain 2 URLs above are detected as greyware. MapReduce just needed about 30 seconds to process all these URLs of 95 logs. As results summarized in Table 3, beside 14 malicious URLs which are detected in Phase 1 (preprocessing phase), 5 URLs requested are detected as malicious in Phase 3 (classification and detection phase), 2 of them are matched with C&C servers communicated by installed Zeus bots and other 3 URLs are detected from experimental captured data. All the detected greyware communication groups are confirmed when they come from shopping sites, social media, and adverting companies. Remaining 275 URLs are unclustered; system can not detect these URLs.

Table 3: Experimental results.

Phase	Malicious URLs	Greyware		Unknown URLs
		Group	URLs	
Phase 1	14			
Phase 3	5	393	5327	275

These constitute a false negative of 4.9% and the accuracy rate reaches 95.1%.

6. Conclusion and Future Work

In this paper, a new method is proposed to detect and classify autoware communication based on its behaviour via analysis of HTTP traffic. The major advantage of the proposed method is that it just used minor features in HTTP

traffic and does not use any signature or content-based technique. In addition, big data application framework also is proposed by combination of two leading technologies, which are the power of distributed processing of MapReduce of Hadoop and the convenient in working with unstructured data through XDBC servers of NoSQL database MarkLogic. Experiment results are promised and methods are working well in private network environment.

There are some reasons contributing undetected rate. First, even autoware commonly communicates with sites by the same behaviour, some rare cases of autowares' requests are different. Second, some types of autoware have less activities in network since they just send out little requests. In other situations, users' Internet accessed traffic also might be automated communication since their access sites automatically refresh its contents via HTML script such as JavaScript or Flash. In these cases, clustering and detection of these URLs access graphs are become tougher. Based on this result, with the objective of reducing the undetected rate, some new features need to be considered in the future work. For that matter, data size sent in each request is regarded since this feature from malicious bot communication to its C&C server is almost steady whist variation of adware's data size in each request depends on the content which they get. In addition, unclustered URLs are also considered to be classified by checking the matching between domain name part of them and clustered group which is in clustering phase.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

References

[1] MapReduce Tutorial, *Apache Hadoop*, 2008, https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html.

[2] MarkLogic database, "What is Marklogic," 2015, http://www.marklogic.com/what-is-marklogic/.

[3] MarkLogic 8 Product Documentation, https://docs.marklogic.com/.

[4] M. A. Rajab, L. Ballard, N. Lutz, P. Mavrommatis, and N. Provos, "CAMP: content-agnostic malware protection," in *Proceedings of the Network and Distributed Systems Security Symposium (NDSS '13)*, Internet Society, 2013.

[5] A. Averbuch, M. Kiperberg, and N. J. Zaidenberg, "An efficient VM-based software protection," in *Proceedings of the 5th International Conference on Network and System Security (NSS '11)*, pp. 121–128, IEEE, Milan, Italy, September 2011.

[6] P. Royal, M. Halpin, D. Dagon, R. Edmonds, and W. Lee, "PolyUnpack: automating the hidden-code extraction of unpack-executing malware," in *Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC '06)*, pp. 289–298, IEEE, Miami Beach, Fla, USA, December 2006.

[7] J. Oberheide, E. Cooke, and F. Jahanian, "Cloudav: N-version antivirus in the network cloud," in *Proceedings of the 17th Conference on Security Symposium*, pp. 91–106, USENIX Association, 2008.

[8] D. Ashley, *An Algorithm for HTTP Bot Detection*, University of Texas at Austin—Information Security Office, Austin, Tex, USA, 2011.

[9] W. Lu, M. Tavallaee, and A. A. Ghorbani, "Automatic discovery of botnet communities on large-scale communication networks," in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (ASIACCS '09)*, pp. 1–10, ACM, Sydney, Australia, March 2009.

[10] M. Eslahi, H. Hashim, and N. M. Tahir, "An efficient false alarm reduction approach in HTTP-based botnet detection," in *Proceedings of the IEEE Symposium on Computers & Informatics (ISCI '13)*, pp. 201–205, Langkawi, Malaysia, April 2013.

[11] B. AsSadhan and J. M. F. Moura, "An efficient method to detect periodic behavior in botnet traffic by analyzing control plane traffic," *Journal of Advanced Research*, vol. 5, no. 4, pp. 435–448, 2014.

[12] S. Shin, Z. Xu, and G. Gu, "EFFORT: a new host-network cooperated framework for efficient and effective bot malware detection," *Computer Networks*, vol. 57, no. 13, pp. 2628–2642, 2013.

[13] Y.-S. Chen, H.-S. Liu, Y.-H. Yu, and P.-C. Wang, "Detect phishing by checking content consistency," in *Proceedings of the 15th IEEE International Conference on Information Reuse and Integration (IRI '14)*, pp. 109–119, Redwood City, Calif, USA, August 2014.

[14] A. Blum, B. Wardman, T. Solorio, and G. Warner, "Lexical feature based phishing URL detection using online learning," in *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security (AISec '10)*, pp. 54–60, 2010.

[15] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious URLs," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pp. 1245–1254, ACM, Paris, France, July 2009.

[16] T.-C. Chen, S. Dick, and J. Miller, "Detecting visually similar web pages: application to phishing detection," *ACM Transactions on Internet Technology*, vol. 10, no. 2, article 5, pp. 5:1–5:38, 2010.

[17] G. Bartlett, J. Heidemann, and C. Papadopoulos, "Low-rate, flow-level periodicity detection," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS '11)*, pp. 804–809, April 2011.

[18] M. C. Tran and Y. Nakamura, "In-host communication pattern observed for suspicious HTTP-based auto-ware detection," *International Journal of Computer and Communication Engineering*, vol. 4, no. 6, pp. 379–389, 2015.

[19] M.-P. Dubuisson and A. K. Jain, "A modified Hausdorff distance for object matching," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Conference A: Computer Vision & Image Processing*, vol. 1, pp. 566–568, IEEE, Jerusalem, Israel, 1994.

[20] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, 1993.

[21] C. Brooks, *Enterprise NoSQL for Dummies*, John Wiley & Sons, Hoboken, NJ, USA, 2014.

[22] MarkLogic Developer License, *Enterprise NoSQL Power for Developers*, 2008, https://developer.marklogic.com/free-developer.

[23] N. Falliere and E. Chien, "Zeus: King of the bots," Symantec Security Response, 2009, https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/zeus_king_of_bots.pdf.

[24] VirusTotal, 2015, http://virustotal.com/.

[25] McAfee Web Gateway, http://www.mcafee.com/us/products/web-gateway.aspx.

Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2016, Article ID 4361719, 10 pages
http://dx.doi.org/10.1155/2016/4361719

Research Article

The Comparison of Distributed P2P Trust Models Based on Quantitative Parameters in the File Downloading Scenarios

Jingpei Wang and Jie Liu

Information Security Research Center, China CEPREI Laboratory, Guangzhou 510610, China

Correspondence should be addressed to Jingpei Wang; wjpbupt@163.com

Received 26 December 2015; Revised 29 April 2016; Accepted 9 June 2016

Academic Editor: Arash Habibi Lashkari

Varied P2P trust models have been proposed recently; it is necessary to develop an effective method to evaluate these trust models to resolve the commonalities (guiding the newly generated trust models in theory) and individuality (assisting a decision maker in choosing an optimal trust model to implement in specific context) issues. A new method for analyzing and comparing P2P trust models based on hierarchical parameters quantization in the file downloading scenarios is proposed in this paper. Several parameters are extracted from the functional attributes and quality feature of trust relationship, as well as requirements from the specific network context and the evaluators. Several distributed P2P trust models are analyzed quantitatively with extracted parameters modeled into a hierarchical model. The fuzzy inferring method is applied to the hierarchical modeling of parameters to fuse the evaluated values of the candidate trust models, and then the relative optimal one is selected based on the sorted overall quantitative values. Finally, analyses and simulation are performed. The results show that the proposed method is reasonable and effective compared with the previous algorithms.

1. Introduction

Due to the openness of distributed networks, security issue becomes one of the most important challenges when deploying these networks into application. Traditional strategies, such as traditional encryption and access control, because of their poor scalability, are no longer suited for resolving security issues of distributed P2P system. Trust management resolves the security issues in semantic and behavioral levels and filters malicious nodes based on their real-time behaviors between transactions. Trust mechanism can transfer between heterogeneous mixed networks seamlessly, and the researches of trust management are of considerable interest in recent years [1, 2].

The researches of trust in computer network have emerged several years ago. However, there are no uniform definitions of trust related issues. First, we give our definition of trust and trust management in this paper.

Trust. Trust has been interpreted as opinion, reputation, probability, and so forth. In the trust management based on trust degree, trust is defined as subjective expectations that denote the uncertainty in collaboration between the subjects and agents.

Trust Model. Trust model defines the method and procedure of trust modeling and trust evaluation.

Trust Management. Trust management is a service mechanism that self-organizes a set of items based on their trust status to take an informed decision.

The core issue of trust management is constructing reliable trust models, and a wide range of P2P trust models for distributed P2P network is proposed during the last decades [3–14]. One problem is that there are no evaluating criteria for comparing these varied models, making it difficult for an interested party to decide upon an optimal trust model to implement. We give a P2P scenario as follows.

One user (service requester) performs the file download in P2P network, supposing that varied trust models are

available for computing the trust values of the target service provider. The user should ask some questions firstly:

(1) Which trust model is the most consistent to compute and guide the trust flows?

(2) Which trust model can assist in providing more qualified service in this application context?

(3) The diverse requirements from decision maker would influence the results of choosing the optimal trust model; what is the degree of this influence?

In order to resolve the above quandaries, it is necessary to develop a quantitative method to analyze some trust models and induce choosing the most proper one according to the application context and multiple requirements of user.

In this paper, a method for comparing trust models based on hierarchical model of parameters is proposed. The evaluated parameters are extracted from the trust related concepts, network context, and the requirements of users. The evaluated values of trust models are obtained by quantitative calculation of the parameters model with the Delphi and fuzzy inference methods. The optimal trust model is selected based on the sorted quantized values. Analysis and simulation results show that the proposed algorithm is reasonable and effective.

The rest of the paper is organized as follows. Section 2 presents the related works. Section 3 outlines the parameters and gives qualitative analysis. In Section 4, hierarchical parameter modeling and formal quantization are proposed. Further, parameter fusion and trust model evaluation are given in Section 5. Section 6 presents analysis and simulation, followed by the conclusions in Section 7.

2. Related Works

Various trust models have been proposed for the P2P in the past decade [3–14]. From the aspect of trust measurement and modeling method, the trust models can be divided into hybrid distributed approach [5], trust model based on weighted average method [6, 7], trust model based on game theory [8], multiple factors trust model [9, 10], Bayesian trust model [11], fuzzy inferring trust model [12, 13], trust model based evidence theory [14], and so forth. However, there are rare sound researches on how to compare trust models. Some researchers focus on the qualitative analysis or guidance of the trust models; some others focus on the quantitative evaluation.

Wojcik et al. introduced a set of criteria to analyze trust models [15]. The criteria consist of four parts: trust establishment, trust initialization, trust updating, and trust evaluation. Each part is followed by some suggestions. It provided a common framework for the development of a sound trust model, though there is no concrete realization. Rodriguez-Perez et al. discussed the main issues that a reputation framework must address and analyzed the most representative reputation systems in fully distributed peer-to-peer systems [16]. They also discussed the main advantages and drawbacks of each proposal in relation to peer-to-peer reputation system requirements. Mármol and Pérez described several trust and reputation models for distributed and heterogeneous networks and compared to provide an evaluation of the most relevant works [17]. They suggested that certain security threats and the specific features of the distributed network where a model is to be deployed should be considered carefully to improve the evaluation accuracy. He and Wu discussed the theory basics, applications, advantages, and disadvantages of some reputation systems [18]. They considered that aggregation overhead, storage efficiency, and reputation accuracy are three key issues in the design of reputation for P2P network. Azzedin proposed a reputation assessment process and used it to classify the existing reputation systems [19]. He focused on the different methods in selecting the recommendation sources and collecting the recommendations. These two phases can contribute significantly to the overall performance owing to precision, recall, and communication cost. All the above methods take the right direction to analyze trust model, but there is no concrete algorithm to compare the investigated trust models to select an optimal one to implement.

Schlosser et al. presented a formal model for describing reputation systems [20]. Based on the formal model, a generic simulation framework was implemented. The defects of this simulation framework are shortage of theoretical analysis for parameter settings and that only reputation systems are taken into account. Yang et al. proposed a method to evaluate the trust model by treating the trust model as a black box and comparing the output with the input [21]. Their work is similar to software testing in software engineering. The evaluated results are compared with two parameters: sensibility and foreseeability. However, it is difficult to model the overall features of trust models using merely two measurable parameters. In a word, the existing methods have some deficiencies, and trust model evaluation remains an open issue.

3. Parameters Extracting and Qualitative Analysis of Trust Models

3.1. Parameters Extracting. From the user perspective, the P2P can be considered as a service supporter and trust management is an integral mechanism of the network system that assists the system in providing qualified service. There are some parameters existing in trust relationship and trust models.

Subjectivity. From the definition of trust we can see that trust is a subjective concept. It is provided by observers based on their subjective judgment. Different observer, different period, different mood, and different scenario may induce different judgments. Notice that reputation is not subjective as it is based on the historical behavior. Also, not all parameters are subjective; there are some QoS parameters that can be perfectly quantified (delay, jitter, etc.)

Fuzziness. Trust is a blurry concept. Three factors induce the fuzziness: uncertainty, inaccuracy, and no clarity. A proper

trust model should be able to express these blurry concepts in trust establishing and measurement. However, most trust models proposed are based on numerical computing and the methods of quantization and inference of the fuzzy relationship varied from one to another. How to express, quantize, and infer fuzzy relationship also belongs to the scope of the fuzziness.

Time Decay. Trust should decay as time passes by. For example, trust relationship formed 3 years ago is less credible than that formed 3 days ago. However, the decayed amplitude and range have no unified conclusion in varied trust models. Whether to introduce the decay factor and what decayed range is reasonable need evaluation for the trust models.

Robustness. There are malicious attacks in distributed system, including unintentional attacks, for example, data transmission delay and block induced denial of service, and malicious attacks, for example, false feedback, collusive cheating, and malicious calumniation. An excellent trust model should be able to resist various attacks and avoid the malicious nodes from transactions.

Reward and Punishment. In a trust system, various nodes have different performance according to their ability and wish. Trust models should provide a proper mechanism to reward nodes with high trust values to encourage them to provide better service. Meanwhile, punish nodes with malicious performance by reducing their trust values or forbidding their transaction.

Sensitivity. This feature reflects the evolution speed of the trust relationship with the disturbance of the network behaviors. The evaluated factors of sensitivity include the changing speed of trust value, handing speed of malicious attack, and the speed of searching. Moreover, sensitivity is associated with application scenario; that is, higher sensitivity is suited for high precision network (i.e., military network), while lower sensitivity is popular with tolerant network.

Transitivity. When an entity needs to judge the globe trust value of another entity within the domain or in the distance, the trust transitivity is necessary. Trust or reputation transmitted mainly through recommended mechanism from a series of middle nodes. However, this recommended relationship is not always true; if node A trusts node B and node B trusts node C, we cannot infer that node A always trusts node C. A good trust model should include recommended mechanism as well as the reasonable disposal of asymmetric recommendation.

Scalability. This parameter mainly depicts the relationship between network size and network load in dealing with the trust relationship. The calculation complexity remains low or increases slowly with the increase of network nodes meaning better scalability. Specifically, trust model with lower time complexity, lower space complexity, and more efficient transmission pattern is a better model.

Other parameters could be considered in deciding the performance of a trust model, for example, the assessment or evolution of trust, usability, variable assignments.

3.2. Qualitative Analysis of Trust Models. In this section, we will address the characteristics of some traditional P2P trust models based on the extracted functions.

Rodriguez-Perez et al. [5] proposed a superpeer reputation framework for P2P network. There are single peer and sure-peer. The peers always maintain their own local reputation database; the system is fault tolerant to sure-peer unavailability. Surework introduces incentives in order to promote that nodes with higher capabilities become superpeers and assume more tasks than normal peers. Reciprocity is also promoted by encouraging peers to provide better services to most reputable client peers. Therefore, the robustness and reward-punishment are obvious. Malicious actions can be found by local reputation and clusters' opinion quickly, which reflects some sensibility. Reputation can be calculated and transferred to other sure-peers and clusters. The drawbacks of surework are increasing system complexity and computational cost, the scalability being not very good, and the other parameters being not mentioned.

EigenTrust [6] is a trust and reputation model for P2P networks where each peer is assigned a global trust value based on its transaction history. The trust value changes gradually, and the trust level determines the different transaction chance. The subjectivity, fuzziness, and time decay are not found in this model, as it adopted objective calculation of transaction results and did not consider the time decay factor. An important feature of this model is the presence of some pretrusted peers that help to break up malicious collectives, and peers can avoid transactions with partial malicious peers, so the robustness and punishment mechanism are qualified, whereas the handing speed of malicious attack and the changing speed of trust value are not obvious, which mean lower sensitivity. The transitivity is clearly presented though the asymmetric recommendation is not considered. The pretrusted peers change the convergence and achieve a significant reduction of overhead in the system, so the performance of scalability is acceptable.

PeerTrust [7] is a reputation based trust model, where more factors are introduced to compute trust value for each peer. The feedback-based evaluation, satisfaction of transaction, participating degree, community context, credibility, and so forth are considered in the trust evaluation. The subjectivity is presented with the participation of the peers' judgment of satisfaction and credibility. PeerTrust algorithm also proposed an adaptive time window-based algorithm to reflect the most recent behaviors, so time decay is considered. Good feedback will gain better results and bad feedback will be found, which means the presence of reward-punishment mechanism. The robustness and transitivity are improved compared with EigenTrust, whereas the scalability is decreased as complexity computation.

Harish et al. designed and analyzed a game theoretic model for P2P trust management [8]. The trust framework incorporated self-experience and reputation to calculate trustworthiness of a peer. Various strategies like game

tree strategy, dynamic strategy, and auditing strategy were proposed for selecting peers for doing job. The method addressed the problem of the selfish behavior; different entity uses different strategy. The intelligent entity can update the reputation values of other interactive nodes; and reward and punishment are performed directly by the payoff. Therefore, subjectivity, transitivity, and reward-punishment are obvious. It can avoid internal malicious behaviors and the robustness is presented, but the calculation of reputation and strategies and their evolution induce larger overhead. Fuzziness, time decay, and sensitivity are not mentioned.

Li et al. proposed a multidimensional trust model for large scale P2P computing [9]. It involves many factors, that is, assumptions, expectations, behaviors, and risks, to reflect the complexity of trust. Moreover, the weights of these factors are dynamically assigned by series of objective algorithms. The subjectivity, fuzziness, and time decay are not mentioned. This model gave a scene where malicious feedback is changing while the accuracy and the adaptability maintain a proper level, which means good robustness. The reward-punishment of trust value as well as sensitivity is not obvious, the transitivity is mentioned, and finally the scalability is excellent with the mechanism of direct trust tree.

Wang and Vassileva proposed a Bayesian trust model in P2P networks [11], it takes trust as a multifaceted concept, and peers need to develop differentiated trust in different aspects of other peers' capability. Bayesian network provides a flexible method to combine different aspects of trust. The subjectivity is obvious, and the calculation of differentiated trust is rapid, though the final results are measured by the number of transactions meaning low sensitivity. The transitivity is considered and the scalability is excellent as the lower load of computation. However, the fuzziness, time decay, robustness, and reward-punishment mechanism are not mentioned in this model.

There are other trust models proposed by different methods, for example, fuzzy trust model [13] and D-S evidence trust model [14]. The analysis procedure is the same as that of the above trust models and omitted here. The results of qualitative analysis are shown in Table 1. In this table, "$\surd$" stands for trust models having responding parameters and "$\times$" means not having related parameters or the merits of related parameters are not obvious. From the analysis we can see that the investigated trust models all have their advantages and disadvantages from the aspect of the parameters. And it can be inferred that the degree for one parameter owned by several trust models differs from one to another; for example, EigenTrust and PeerTrust both have robustness, whereas the intensity of robustness is varied as adopting different mechanism. The subjectivity between Jøsang model and Bayesian model is varied as different number of factors is adopted in each model.

In a word, parameter distribution and parameters degree among trust models are unbalanced, making it difficult for an interested party to decide upon a particular trust model to implement. Nevertheless, we can find the relatively optimal trust model through quantitative comparison in a concrete scene. In the next section, a quantized evaluation is addressed.

4. Hierarchical Parameters Modeling and Formal Quantization

4.1. Hierarchical Modeling of Parameters. On one hand, it is difficult to analyze trust models using more than eight parameters directly and simultaneously. Some parameters are conflicted; for example, complicated algorithms are used to deal with attacks, which increase robustness but deteriorate scalability. Some parameters are correlative; better punishment mechanisms would lead to better robustness. Moreover, different parameters concern different aspects of service, and sometime the same parameters may concern more than one aspect; for example, scalability is involved in network structure, and robustness concerns the service reliability as well as network structure.

On the other hand, there are other decision factors, such as network scene and individual policy of observer. According to previous definition, trust management is a third-party auxiliary mechanism assisting the system in providing qualified service. From the service perspective, we can extract some factors of the quality of service with a trust model: function conformance, reliability, the adaptability for network context, and the specific requirement of a user, each of which is followed by some parameters; for example, function conformance includes transitivity and flexibility which reflect the reasonability of trust modeling. The performance of each factor can be evaluated by some low-level parameters.

A natural method is establishing a hierarchical structure to combine the above two aspects; the parameters and their upper factors (criteria layer) can be considered comprehensively. It can distribute conflicting parameters to different decision criteria layer, and correlative parameters can be laid into one layer for coordinate evaluation. Then, a fusion method is designed to fuse these parameters and criteria layer to obtain the overall performance of trust models. In this paper, the decision factors in criteria layer are described as follows.

Function Conformance. This layer mainly focuses on the reasonability of trust representation, the conformance of trust attributes, and the mechanism of performing the service task properly. The most obvious functions of trust management are measuring the uncertainty of the nodes' behaviors and self-organizing a set of objects to perform the task (e.g., routing or transmitting data). Whether the uncertainty can be measured properly or not will be evaluated in this layer. Subjectivity, fuzziness, time decay, and transitivity will be used to characterize the conformance of trust mechanism and be distributed to this layer.

Service Reliability. This layer estimates whether the service provided by trust mechanism is reliable. Trust management is a third-party auxiliary mechanism. The provided service should be qualified. The robustness describes how the trusted cooperators resist malicious attacks or shield from the malicious node to maintain the stability of service. The reward and punishment (with more reasonable resource distribution) and sensitivity (reflecting the reaction speed of attacks and

Table 1: The distribution of parameters of some P2P trust models.

Trust model	Parameters							
	Subjectivity	Fuzziness	Time decay	Robustness	Reward & punishment	Sensibility	Transitivity	Scalability
EigenTrust	×	×	×	√	√	×	√	√
PeerTrust	√	×	√	√	√	×	√	×
Surework	×	×	×	√	√	√	√	×
Harish et al. [8]	√	×	×	√	√	×	√	×
Li et al. [9]	×	√	×	√	×	×	√	√
Wang and Vassileva [11]	√	×	×	×	×	√	√	√
Wang et al. [12]	√	√	√	×	×	√	×	√
Tian and Yang [14]	×	√	√	√	×	×	√	×

the sensitive degree of the changeable trust value) will be evaluated in this layer.

Structure Adaptability. This layer mainly evaluates the dynamic relationship between the trust model and network environment. The structure, size, topology, and the dynamics of the target network all will influence the execution of task. A reasonable model should be able to adjust to the change of network structure. The scalability should maintain excellent state with the expansion of nodes, the transitive path is available with the change of structure, and the reaction is timely when important nodes change or immediate service is needed. Therefore, scalability, transitivity, and sensitivity are related to this layer.

Strategies Differences of Observer. The observer may have different requirement and secure policies and even different interest and preferences, in using the trust model for certain context. In addition, it also includes some performance index, such as the usage of resource (overhead or scalability), disposal speed and quality of service of trust mechanism, the ability of surviving (mainly robustness), and the special need of sensitivity (i.e., higher accurate application).

Other criteria will be populated to add in criteria layers. The detailed hierarchical structure of parameters and factors are shown in Figure 1. Parameters act as basic layer, middle service factors act as criteria layer, and the top layer is goal layer.

The comprehensive assessment of the trust models for performing a specific task in a certain context can be derived from reasonable evaluation of the following criteria layer and basic layer in succession.

4.2. Formal Evaluation of Trust Model. The extracted parameters characterize the basic functional feature of a trust model, $P = \{p_1, p_2, \ldots, p_n\}$ denote the set of parameters, and n is the number of parameters. In order to perform quantitative analysis, we quantify the parameters and select an algorithm to fuse these parameters; that is $C :\mapsto (\mathrm{TM}, P, I)$, where TM is a trust model, I denote an integration algorithm, and C is evaluation result. A direct evaluation can be modeled into a functional form, as shown in

$$C(\mathrm{TM}) = f_I(f_1(p_1), f_2(p_2), \ldots, f_n(p_n)), \quad (1)$$

where $f_i(p_i)$, $i \in [1, n]$, denotes the quantization of ith parameter. In order to achieve the overall unified value, the normalization is embedded in the quantization of $f_i(p_i)$; in other words $Q(P) = (f_1(p_1), f_2(p_2), \ldots, f_n(p_n))$ is a dimensionless vector. $f_I(\cdot)$ denote an integrating function. A simple integration is weighted average of $Q(P)$. A more complicate but rational method is fuzzy fusion that will be used in this paper.

4.3. Quantization of Parameters. There exists a mapping f between the impact factors and the quantified value of each parameter for a given trust model. In our algorithm, the quantitative procedure of a parameter includes extracting the impact factors of each parameter, scoring the impact factors, and performing normalization and fuzzy integration of the impact factors to obtain the quantitative value of single parameter.

For a single parameter p_i, C denotes the factor set: $C = \{c_1, c_2, \ldots, c_m\}$, and m is the number of factors. The factor is extracted based on three considerations: the definition of parameter, evaluated points, range of parameter and some experience of experts. For example, scalability is related to time complexity, space complexity, transmission, and efficient storage of data; the impact factors of sensitivity include the changing speed of trust value and handing speed of malicious attack and the speed of searching and timely reaction when network topology changes.

For a single impact factor c_j, $j \in [1, m]$, we evaluate it with specific measure, range. A simple method is using fuzzy theory to determine the range and level of the evaluated factor according to the experience of observer (i.e., for the rationality, irrational, default, and lowest rationality, medium rationality, favorable rationality, and highest rationality, denoted as five intervals from 0 upper to 1, resp.; quantized step is 0.2 that denotes the uncertainty). Considering that there are some manufactured discrepancies for each factor, Delphi method can be introduced to collect and filter the divergent answers and obtain the quantified value.

The Delphi method is an interactive forecasting method that relies on a group of experts. The experts answer questions in two or more rounds. It is believed that, after several rounds, the range of the answers will decrease and the group will converge towards the "correct" answer. Finally, the process is stopped after a predefined stop criterion (e.g., stability). In

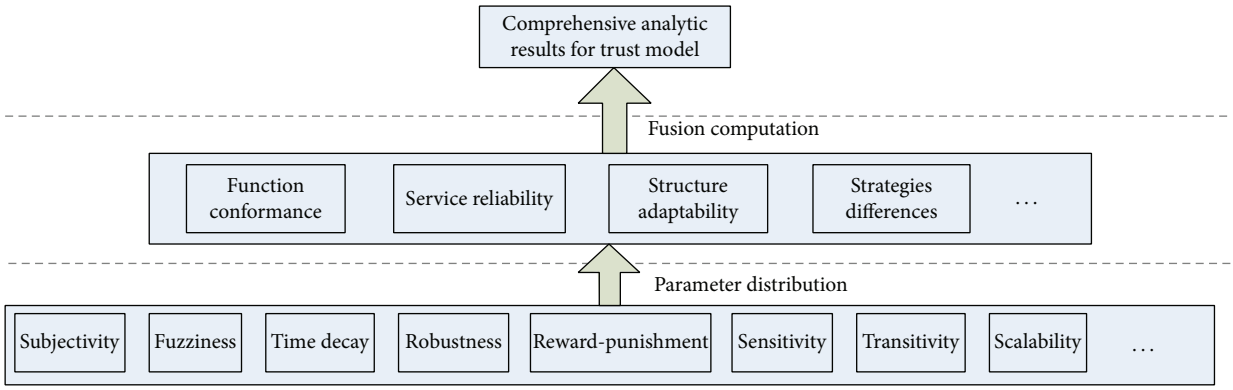


Figure 1: Hierarchical structure of parameters for trust models.

this paper, several questions are defined firstly based on the multiple factors of a particular c_j. Each question is followed by certain options that denote the level of possible answers (i.e., rationality, from being irrational to highest rationality). These questionnaires are provided to several experts. After several rounds, the final correct feedback will be determined, and the final quantized value of a factor $Q(c_j)$ can be obtained. Repeat above quantization until all m factors are quantized, denoted as $Q(C) = \{Q(c_1), Q(c_2), \ldots, Q(c_m)\}$.

Notice that m elements in $Q(c_j)$ may be measured in different unit; take the scalability as an example; the units of time complexity and space complexity are time (ms) and capacity (kb). Firstly, we normalize these different units. The popular method of normalization is max–min method:

$$Q_m\left(c_j\right) = \frac{Q\left(c_j\right) - \min\left(Q\left(c_j\right)\right)}{\max\left(Q\left(c_j\right)\right) - \min\left(Q\left(c_j\right)\right)}, \quad j \in [1, m], \quad (2)$$

where $\max(Q(c_j))$ and $\min(Q(c_j))$ are the maximum quantized value and the minimum value determined by the range of jth factor. $Q_m(c_j)$ is a numeric value between 0 and 1 after normalization. Repeat above disposal until all m factors are normalized, denoted as $Q_m(C) = \{Q_m(c_1), Q_m(c_2), \ldots, Q_m(c_m)\}$.

The final procedure is integrating the impact factors to obtain the overall quantized value of single parameter. As the impact factors are independent of each other, a simple integration is weighted sum of the quantized value of each factor. The integral is defined as follows:

$$\begin{aligned} Q\left(p_i\right) &= \sum_{j=1}^{m} w_{c_j} Q_m\left(C_j\right) \\ &= w_{c_1} Q_m\left(C_1\right) + w_{c_2} Q_m\left(C_2\right) + \cdots \\ &\quad + w_{c_m} Q_m\left(C_m\right), \end{aligned} \quad (3)$$

where w_{c_j}, $j \in [1, m]$, denote the weight of $Q_m(c_j)$ and satisfy $w_{c_1} + w_{c_2} + \cdots + w_{c_m} = 1$. The weights are determined by the experience of the experts in consideration of importance degrees of evaluation criteria. The final integration is finished through (3), and the quantized value of parameter p_i is obtained, denoted as $Q(p_i)$.

Repeat all the above procedures until all n parameters are quantized, denoted as

$$\begin{aligned} Q(P) &= \left(f_1\left(p_1\right), f_2\left(p_2\right), \ldots, f_n\left(p_n\right)\right) \\ &= \left(Q\left(p_1\right), Q\left(p_2\right), \ldots, Q\left(p_n\right)\right), \end{aligned} \quad (4)$$

where $f_i(p_i)$ is a quantification function with all the above procedures and varied for different parameter.

5. Parameter Fusion and Trust Model Evaluation

A fusion algorithm based on fuzzy inference is proposed to combine the parameters in hierarchical structure in Figure 1.

5.1. The Weights of Distributive Parameters. In Figure 1, the middle criteria layer and lower parameters have certain relation. Several parameters are related to one or more factors in criteria layer. As the parameters are dependent on each other, fuzzy integral in (3) is inappropriate. Without loss of generality, we suppose that one factor in criteria layer is related to all the parameters, and the goal layer is related to all the factors in criteria layer. Firstly, the weights of parameters to single factor in criteria layer and the weights of factors in criteria layer to the goal layer are calculated. The entropy-weight coefficient method is a quantitative objective method and will be applied in our paper.

Entropy-weight coefficient method is a quantitative risk evaluation method [22]. The relative importance of a risk factor to an evaluated system can be measured by its entropy, which is calculated by the fusion of probability values denoting the supporting degree of risk factors to indexes of evaluation set for the system. In this paper, the parameters are considered as risk factors; one factor in the criteria layer is considered as evaluated object. Set several statuses for the evaluated object, give the probability of each parameter at each status, and apply entropy-weight coefficient method calculating the relative importance (weight) of each parameter to one upper factor. The statuses can be set based on certain evaluation set (i.e., rationality, from being irrational to highest rationality) used in previous parameter

quantitation, and the probabilities that each parameter stay at certain status of the evaluation set can be determined by the same experts that used the Delphi method in Section 4.3. The detailed procedure of entropyweight coefficient method is referred to in related book (i.e., [22]) and omitted here.

Repeat the above calculation until all the weights are obtained. The weight of parameters of kth ($k \in [1, s]$, s being the number of factors in criteria layer) factor in middle criteria layer is denoted as $W_{pk} = \{w_{k1}, w_{k2}, \ldots, w_{kn}\}$, where n is the number of parameters. For the sake of simplicity, discard the weights that equaled 0 (e.g., time decay has no relation with structure adaptability of a trust model; the probability density function of status is always 0), and obtain effective weights, denoted as $W_k = \{w_{k1}, w_{k2}, \ldots, w_{kb}\}$, where $b \in [1, n]$. Repeat the above filtering; the weights of parameters to one factor in criteria layer and the weights of factors in criteria layer to the goal layer are obtained.

5.2. The Fusion of Parameters Based on Fuzzy Inference. The evaluated values of single factor in criteria layer and the evaluated value of goal layer will be fused by fuzzy inference in succession. In fuzzy set theory, a variable $V_T = \{v_1, v_2, \ldots, v_b\}$, v_k ($k = 1, 2, \ldots, b$) denoting the value of object T at the point k (k-level value) according to the defined membership functions in a given discourse domain, and the problem is how to obtain V_T under a given tree (i.e., Figure 1).

We evaluate kth factor in the criteria layer followed by b parameters. Set a discourse domain for the kth factor (e.g., reliability, from the least reliable to the most reliable and 5 levels are divided as the least reliable, little reliable, medium reliable, favorite reliable, and the most reliable and $V_q = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ stands for quantitative border value). The membership function is a trapezoidal function; the prototype is shown below:

$$\mu_T(x) = \begin{cases} \left(\dfrac{x-t}{1-t}\right)^2, & 0 \le x \le \dfrac{a+1}{2} \\ 1 - \left(\dfrac{x-t}{1-t}\right)^2, & \dfrac{a+1}{2} \le x \le 1, \end{cases} \quad (5)$$

where a is a defined threshold and t ($t \in [0, 1]$) is the offset of positive x and is set to 0.2 to form 5 curves.

The quantitative values of b parameters have been achieved in Section 4.3; each value can be mapped to a membership degree according to membership function, eventually formed into an evaluation matrix $R = (r_{gh})_{b \times l}$. The weighted vector is $W_k = \{w_{k1}, w_{k2}, \ldots, w_{kb}\}$, and then the overall vector of the kth factor is denoted as

$$V_T = \{v_1, v_2, \ldots, v_l\} = (w_{k1}, w_{k2}, \ldots, w_{kb}) \times \left(r_{gh}\right)_{b \times l}. \quad (6)$$

Define the evaluated value of the kth factor: $V_k = \max(V_T)$, where $\max(V_T)$ is the maximum membership degree of V_T.

Repeat the above procedure until all the evaluated values of the factors in the criteria layer are obtained. Based on the evaluated value of the factors in the criteria layer and the weights of factors in middle layer to the final goal layer, the fuzzy comprehensive judgment is performed with the same method as that used in calculating the evaluated value of the kth factor V_k to obtain the comprehensive analytic value for a trust model.

Then, the observer can compare the eventual evaluated value with the threshold to judge whether the trust model is qualified. The threshold is set based on some factors, for example, accuracy and fee. We can evaluate a set of trust models, sort the evaluated values, and choose an optimal trust model (usually the model with maximal evaluated value) for implementation.

5.3. The Outline of the Evaluated Procedure. The main steps of the proposed method are summarized:

(1) Based on the structure in Figure 1, apply entropy-weight coefficient method to calculate the weights of parameters to factors in criteria layer and the weights of factors in criteria layer to goal layer. Meanwhile, determine the distributive b parameters for kth factor in criteria layer.

(2) Parameter quantitation: for a trust model, quantize the extracted parameters with a series of procedures described in Section 4.3, and obtain the vector $Q(P) = (f_1(p_1), f_2(p_2), \ldots, f_n(p_n))$.

(3) Parameter fusion: the evaluated value of a trust model is calculated by the fusion of quantitative values and weights of parameters by fuzzy inference described in Section 5.2. And judge whether the given model satisfies the request according to the defined threshold.

(4) Select a set of trust models, repeat step (2) and step (3), calculate and sort the overall evaluated values, and choose an optimal trust model for implementation.

6. Method Analysis and Simulation

In this section, some discussion, a concrete evaluation experiment, and the effectiveness of the proposed method are addressed in Sections 6.1, 6.2, and 6.3, respectively.

6.1. Some Discussion of the Proposed Method. Consider the following:

(1) Notice that the hierarchical model is an open structure that other parameters and decision factors can be integrated into this model, which reflect the flexibility of the proposed method. Moreover, the hierarchical model is a reference model, and more than three layers might exist when subfactors are being linked to the parameters or the factor in criteria layer.

(2) The weights calculated by our method are stable under the condition that distributed scene and individual policy are determined. And the quantized values of parameters varied from one trust model to another.

6.2. A Concrete Evaluation Experiment. A concrete evaluation experiment is performed. Six traditional trust models

Table 2: Sorted results of evaluated trust models.

Overall score	0.79	0.78	0.65	0.63	0.62
Sorted models	MdTrust	Bayesian	EigenTrust	GTM	PeerTrust

analyzed in Section 3.2 are selected: EigenTrust, PeerTrust, game theory model (GTM for short) [8], multidimensional trust (MdTrust for short) model [9], and Bayesian model [11]. Some conditions are set as follows:

(1) A concrete scene: one user (service requester) performs the file download in P2P network. And the service requester pays more attention to the speed and quality of file download.

(2) The eight parameters and four factors in the middle layer are all considered.

(3) The number of statuses of the four factors is set to 5 (measured by rationality, I-irrational, L-lowest rationality, M-medium rationality, F-favorable rationality, and H-highest rationality); the probabilities of each status for parameters are determined by the same 7 experts in parameter quantitation with Delphi method; the range of quantized value of parameters is 0-1, with quantized step being 0.2.

According to the procedures in Section 5.3, the evaluated values of the 6 trust models are shown in Table 2.

From Table 2, we can see that multidimensional trust model reaches the highest score, as it has more parameters than others, and the robustness and scalability receive higher score in quantitation. PeerTrust compared to EigenTrust, although with better transitivity; worse scalability eventually leads to a smaller overall evaluated score, as the weight of scalability is larger than that of transitivity under the service requester policy.

We can see that none of the candidate trust models satisfies all the parameters. If the threshold is 0.8, then no trust model is qualified. Nevertheless, we can select the relatively optimal trust model (i.e., one received the highest evaluated value) to implement for a special application.

6.3. The Effectiveness of the Method. In this section, we will analyze the effectiveness of the proposed method.

The efficiency of the proposed method: for a given model, for n parameters, suppose that there are m factors mostly. Seven experts carry out two rounds of consultation, each of which needs time t_1, and the combination of m factors costs $O(m)$. The overall time complexity is $n \times (2t_1 \times m + O(m))$, m being small (around 3–5 for each parameter), so the time complexity is controlled. For the weights of parameters, it is needed to calculate the process of the entropy-weight coefficient, time complexity being $O(n)$. Moreover, the weights can be reutilized for the same scene and task.

We further validate the effectiveness of the proposed method by comparing it with previous methods [15, 20, 21]. For the convenience, [15] is denoted as Wojcik's method and [20, 21] are Schlosser's method and Yang's method.

Firstly, the proposed method adopts multiple parameters to evaluate trust model; it is more comprehensive than other works in characterizing the trust issues. Wojcik introduced a series of factors classified into four aspects in establishing a trust model, but the parameter functions were not considered. Yang's method judged the performance of trust model with two parameters: sensibility and foreseeability. In Schlosser, three parameters were used to reflect trust. These methods had failed to reflect the comprehensive characteristics of a trust model.

Secondly, in terms of accuracy, Yang proposed a black box model and compared a set of trust history sequences in the input with the output and then determined the performance of the trust model with sensibility and foreseeability. Its accuracy depends on the initialization of trust and behavioral characteristic. Wojcik displayed entire process of establishing trust comprehensively, but no specific assessment is performed. Schlosser presented a formal model for describing multiple reputation systems, but only reputation systems are taken into account. In our proposal, objective disposal of parameters as well as fuzzy inference is used to quantify the evaluated value of a trust model, the results are more objective and with higher accuracy.

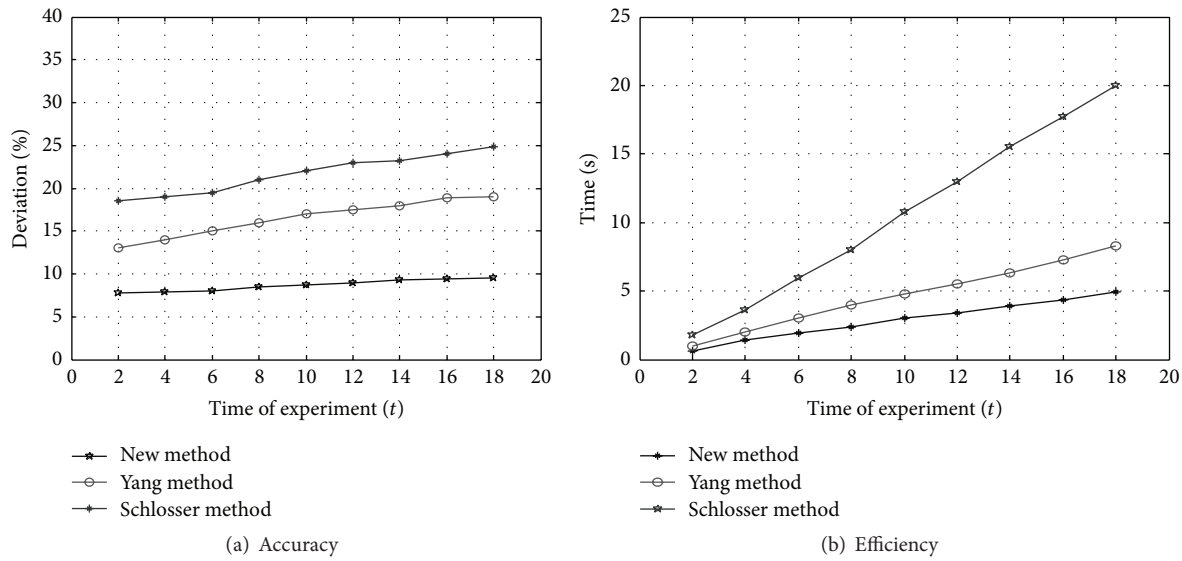
Thirdly, in terms of efficiency, the overhead for our method is controllable and man-made evaluation in Delphi method and the calculation of weights and the fuzzy inference contribute to the calculation load. Wojcik's method does not involve load, and the overhead varied with varied algorithms. Yang's method searched for the history scorings of trusted entities according to the defined behavior characteristics; the time complexity is about $O(n)$, where n is the number of behaviors collected. Schlosser simulated the reputation system in the performance of resisting attacks with the granularity of single node, and the consumption increases with the increase of nodes. The analysis results are shown in Table 3.

In Table 3, the proposed method is denoted as "new method"; the performance is denoted as three levels: good (high), medium, and bad (low). Table 3 explains the superiority of the proposed method.

Further, we present a quantitative comparison among Schlosser's method and Yang's method with simulation. The accuracy and efficiency are compared among three methods.

Accuracy simulation: reflect the change of deviation (y-axis) of evaluated results with the increasing experiment time (x-axis). The conditions are the same as that set in Section 6.2; the deviation is defined as $d = |d_e - d_t| \times 100\%$, where d_e is current evaluated value of the optimal trust model and d_t is the statistical average of its former values. We perform the experiment 20 times. The number of initial nodes of P2P network is 20, where malicious nodes are 20%. The network nodes increase by 5, where the malicious nodes increase with the same percentage (20%), when the experiment time increases by 1.

Efficiency simulation: reflect the relationship between resource consumption (i.e., time consumption) and the number of experiments. The initial number of evaluated trust models is 1 and increases by 1 when the experiment time increases by 1. The simulation results are shown in Figure 2.



(a) Accuracy (b) Efficiency

Figure 2: Simulation results.

Table 3: The comparison of the previous methods.

	Wojcik's method	Yang's method	Schlosser's method	New method
Comprehensiveness	Good	Medium	Medium	Very good
Accuracy	High	Medium	Low	Very high
Efficiency	Uncertain	High	Medium	High

Figure 2(a) describes the accuracy of the three methods. We can find that the deviation of the proposed method is smaller than Yang's method and Schlosser's method; the deviation is controlled within 10%. Therefore, the proposed method is more accurate.

Figure 2(b) describes the efficiency of the three methods; the calculation load increases with the increasing of evaluated models. The proposed method is similar to Yang's method, increasing linearly, but Schlosser's method increases rapidly. The results are in accord with analysis in Table 3.

7. Conclusions

A new method is proposed to compare and evaluate the trust models with quantitative parameters in P2P file downloading scene in this paper. The evaluated parameters are extracted from the trust related concepts and modeled into a hierarchical structure. The Delphi method, entropy-weight coefficient method, and fuzzy inference are applied to obtain a comprehensive evaluated value of a trust model. The optimal trust model is selected according to the sorted overall quantized values of candidate trust models. Analysis and simulation results show that the proposed evaluation algorithm is reasonable and effective. The proposed method resolves the individuality issues, assisting a decision maker in choosing an optimal trust model to implement in specific context. Moreover, the method also can be used to guide the newly generated trust model in theory so that it has better performance in parameter function and adaptability.

Competing Interests

The authors declare that they have no competing interests.

References

[1] Y. Zhong, B. Bhargava, Y. Lu, and P. Angin, "A computational dynamic trust model for user authorization," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 1–15, 2015.

[2] I.-R. Chen, J. Guo, and F. Bao, "Trust management for service composition in SOA-based IoT systems," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '14)*, pp. 3444–3449, Istanbul, Turkey, April 2014.

[3] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati, "Managing and sharing servants' reputations in P2P systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 840–854, 2003.

[4] L. Mekouar, Y. Iraqi, and R. Boutaba, "Detecting malicious peers in a reputation-based peer-to-peer system," in *Proceedings of the 2nd IEEE Consumer Communications and Networking Conference (CCNC '05)*, pp. 37–42, IEEE, Las Vegas, Nev, USA, January 2005.

[5] M. Rodriguez-Perez, O. Esparza, and J. L. Muñoz, "Surework: a super-peer reputation framework for p2p networks," in *Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC '08)*, pp. 2019–2023, Fortaleza, Brazil, March 2008.

[6] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks," in *Proceedings of the 12th International Conference on World Wide Web (WWW '03)*, pp. 640–651, ACM, Budapest, Hungary, May 2003.

[7] L. Xiong and L. Liu, "A reputation-based trust model for peer-to-peer eCommerce communities," in *Proceedings of the ACM Conference on Electronic Commerce (EC '03)*, pp. 275–284, San Diego, Calif, USA, June 2003.

[8] M. Harish, N. Anandavelu, N. Anbalagan, G. S. Mahalakshmi, and T. V. Geetha, "Design and analysis of a game theoretic model for P2P trust management," in *Distributed Computing and Internet Technology: 4th International Conference, ICDCIT 2007, Bangalore, India, December 17–20. Proceedings*, vol. 4882 of *Lecture Notes in Computer Science*, pp. 110–115, Springer, Berlin, Germany, 2007.

[9] X. Li, F. Zhou, and X. Yang, "A multi-dimensional trust evaluation model for large-scale P2P computing," *Journal of Parallel and Distributed Computing*, vol. 71, no. 6, pp. 837–847, 2011.

[10] L. Srour, A. Kayssi, and A. Chehab, "Reputation-based algorithm for managing trust in file sharing networks," in *Proceedings of the Securecomm and Workshops*, pp. 1–10, IEEE, Baltimore, Md, USA, September 2006.

[11] Y. Wang and J. Vassileva, "Bayesian network-based trust model in peer-to-peer networks," in *Proceedings of IEEE/WIC International Conference on Web Intelligence*, pp. 372–378, Halifax, Canada, 2003.

[12] Y. Wang, R. Wang, and Z. Han, "Dynamical trust construction schema with fuzzy decision in P2P systems," *Chinese Journal of Electronics*, vol. 18, no. 3, pp. 417–421, 2009.

[13] S. Song, K. Hwang, R. Zhou, and Y.-K. Kwok, "Trusted P2P transactions with fuzzy reputation aggregation," *IEEE Internet Computing*, vol. 9, no. 6, pp. 24–34, 2005.

[14] C. Tian and B. Yang, "A D-S evidence theory based fuzzy trust model in file-sharing P2P networks," *Peer-to-Peer Networking and Applications*, vol. 7, no. 4, pp. 332–345, 2014.

[15] M. Wojcik, H. S. Venter, and J. H. P. Eloff, "Trust model evaluation criteria: a detailed analysis of trust representation," in *Proceedings of the South African Telecommunications Networks and Applications Conference (SATNAC '06)*, Western Cape, South Africa, September 2006.

[16] M. Rodriguez-Perez, O. Esparza, and J. L. Muñoz, "Analysis of peer-to-peer distributed reputation schemes," in *Proceedings of the IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom '05)*, pp. 1811–1817, San Jose, Calif, USA, December 2005.

[17] F. G. Mármol and G. M. Pérez, "Trust and reputation models comparison," *Internet Research*, vol. 21, no. 2, pp. 138–153, 2011.

[18] C. He and M. Wu, "Comparison and analysis of different reputation systems for peer-to-peer networks," in *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE '10)*, vol. 3, pp. V3-20–V3-23, IEEE, Chengdu, China, August 2010.

[19] F. Azzedin, "Taxonomy of reputation assessment in peer-to-peer systems and analysis of their data retrieval," *Knowledge Engineering Review*, vol. 29, no. 4, pp. 463–483, 2014.

[20] A. Schlosser, M. Voss, and L. Brückner, "Comparing and evaluating metrics for reputation systems by simulation," in *Proceedings of the IAT Workshop on Reputation in Agent Societies*, 2004.

[21] M. Yang, L. Wang, and Y. Lei, "Research on evaluation of trust model," in *Proceedings of the International Conference on Computational Intelligence and Security (CIS '08)*, vol. 1, pp. 345–349, Suzhou, China, December 2008.

[22] H. Su and C. Zhu, "Application of entropy weight coefficient method in evaluation of soil fertility," in *Recent Advances in Computer Science and Information Engineering*, vol. 126 of *Lecture Notes in Electrical Engineering*, pp. 697–703, Springer, Berlin, Germany, 2012.

Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2016, Article ID 3219042, 7 pages
http://dx.doi.org/10.1155/2016/3219042

Research Article

A Privacy-Preserving Outsourcing Data Storage Scheme with Fragile Digital Watermarking-Based Data Auditing

Xinyue Cao,[1] Zhangjie Fu,[1,2] and Xingming Sun[1,2]

[1]*School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China*
[2]*Jiangsu Engineering Centre of Network Monitor, Nanjing University of Information Science and Technology, Nanjing 210044, China*

Correspondence should be addressed to Xingming Sun; sunnudt@163.com

Received 9 December 2015; Revised 6 March 2016; Accepted 12 April 2016

Academic Editor: Isao Echizen

Cloud storage has been recognized as the popular solution to solve the problems of the rising storage costs of IT enterprises for users. However, outsourcing data to the cloud service providers (CSPs) may leak some sensitive privacy information, as the data is out of user's control. So how to ensure the integrity and privacy of outsourced data has become a big challenge. Encryption and data auditing provide a solution toward the challenge. In this paper, we propose a privacy-preserving and auditing-supporting outsourcing data storage scheme by using encryption and digital watermarking. Logistic map-based chaotic cryptography algorithm is used to preserve the privacy of outsourcing data, which has a fast operation speed and a good effect of encryption. Local histogram shifting digital watermark algorithm is used to protect the data integrity which has high payload and makes the original image restored losslessly if the data is verified to be integrated. Experiments show that our scheme is secure and feasible.

1. Introduction

With the development of cloud computing, outsourcing data to cloud storage servers has become a popular way for firms and individuals. Cloud storage reduces data storage and maintenance costs. And cloud storage can provide a flexible and convenient way for users to access their data anywhere. However, the cloud service providers (CSPs) may not be honest and the data should not be disclosed to the CSPs. So the data must be encrypted before it is uploaded to the cloud. Encryption is a fundamental method to preserve data confidentiality. For privacy preserving concerned, data owner can encrypt the data before outsourcing it to CSPs. Many problems of querying over encrypted domain are discussed in research literatures [1–3]. In addition, data owners worry whether the outsourcing data is modified or revealed by the CSPs. It is necessary to add the data auditing service in outsourcing data storage scheme.

In the existing outsourcing data storage schemes, the data auditing methods can be classified into three categories: message authentication code- (MAC-) based methods, RSA-based homomorphic methods, and Boneh-Lynn-Shacham signature- (BLS-) based homomorphic methods [4]. In these methods, the data is calculated using MAC or digital signature and the verification information needs to be attached to the original data. If the data is digitally signed, any change in the data after signature invalidates the signature. Furthermore, these methods increase the data sizes and the time to sign, which is inconvenient in digital media (images, video, audio, etc.). So we use digital watermarking technology to offset the deficiency. Digital watermarking technology hides watermark information in the digital media without affecting data utilization. And it reduces the communication and computation costs. This means digital watermarking technology can provide a more effective auditing method than other cryptographic protocols for auditing.

Many works on outsourcing data storage schemes with digital watermarking are proposed. N. Singh and S. Singh [5] point out that collaboration of digital watermarking and cloud computing can significantly increase the robust of system as well as security of user's data. Boopathy and Sundaresan [6] propose a model of data storage and access process with digital watermarking technology in the cloud. Though they do not give concrete realization, it shows the broad

prospects of applying digital watermarking technology into the cloud environment. In addition, digital watermarking technology is used for data auditing in cloud environment. Wang and Lian [7] focus on the application scenarios of multiwatermarking in cloud environment by investigating the secure media distribution models. Ren et al. [8] propose a provable data possession scheme based on self-embedded digital watermark for auditing service. However, they do not provide privacy preserving with encryption methods. It is believed that supporting privacy preserving is of vital importance to outsourcing data storage.

In this paper, logistic map-based chaotic cryptography algorithm is used to preserve the privacy of outsourcing data, which has a fast operation speed and a good effect of encryption. Traditional encryption techniques such as AES, DES, and RSA have low speed to encrypt media data. And they are not suitable for high real time in media data transmission. Chaotic cryptography has many good characteristics such as sensitivity to initial value, pseudorandom properties, and ergodicity. Logistic map-based chaotic cryptography is a simple nonlinear model, but it has complex dynamics, which is widely used in image encryption. In this paper, logistic map-based chaotic cryptography method is used to permute the positions of the image pixels in the spatial domain. It is suitable for embedding watermark information with local histogram shifting digital watermark algorithm later. Local histogram shifting digital watermark algorithm is utilized to protect the data integrity. It has high payload and makes the original image restored losslessly if the data is verified to be integrated.

We propose an outsourcing data storage scheme supporting auditing service by using fragile digital watermarking technology. Meanwhile, the scheme uses encryption methods to preserve privacy. In this scheme, digital watermarking technology and encryption methods are used to enhance the integrity and privacy of outsourcing data storage. Our contributions are as follows.

(i) We propose an outsourcing data storage scheme supporting privacy-preserving and auditing service. In this scheme, we use the scrambling encryption algorithm based on logistic chaotic map, which has a fast operation speed and a good effect of encryption. Besides, local histogram shifting digital watermark algorithm [9] is used to embed the watermark, which has high payload and makes the original image restored losslessly if the data is verified to be integrated.

(ii) To reduce data owners' overhead cost, a third-party auditor (TPA) is used to verify the integrity of data in cloud. And TPA verifies the data integrity in encryption domain, which ensures the data confidentiality in the auditing process.

The rest of this paper is organized as follows. Section 2 summarizes the related work. Section 3 introduces the proposed scheme. Experiment results are given in Section 4. Section 5 concludes the paper and the future work.

2. Related Work

Many secure outsourcing data storage schemes are proposed these years. The privacy and integrity of data in cloud are the most concerns of data owners. Outsourcing data is often distributed geographically in different locations. CPSs can access the stored data if it is stored in plain format. Data owners have lost control over their data after it is uploaded to the cloud. So data privacy information [10] or sensitivity information [11] causes the outsourcing data to be encrypted in the data storage schemes.

To verify the data integrity, data auditing is considered in outsourcing data storage schemes. Ateniese et al. [12] first define the provable data possession (PDP) model for auditing service in untrusted storages. Juels and Kaliski Jr. [13] describe a proof of retrievability (POR) model, which ensures both "possession" and "retrievability" of data files. Sravan Kumar and Saxena [14] propose a proof of data integrity in the cloud, which could be agreed upon by both clients and the server via the Service Level Agreement (SLA). Hao et al. [15] propose the first protocol that provides public verifiability without TPA. Lu et al. [16] exploit the secure provenance model, which consists of the following modules: system setup, key generation, anonymous authentication, authorized access, and provenance tracking. Their scheme is based on the bilinear pairing techniques. And it records the ownership and the process history of data objects to increase the trust from public users. But all these methods have additional data to verify the data integrity and are not suitable for multimedia file. Digital watermarking technology can offset the deficiency, which is an effective method for data auditing. Digital watermarking can be divided into spatial domain and frequency domain [17]. Spatial domain digital watermark directly embeds watermark information into the image pixels. Frequency domain [18] algorithm embeds watermark information into coefficients of transform domain.

Encryption is a fundamental method to preserve data confidentiality in outsourcing data storage schemes. Digital watermarking technology is an effective method for data auditing. The methods of embedding digital watermark in encryption domain are proposed [6, 19, 20]. In medical domain, many healthcare information systems (HISs) [21] are proposed. Haas et al. [22] propose a privacy-protecting information system for controlled disclosure of personal data to third parties. This scheme uses authentic log files to check the completeness of data. And digital watermarking is used for tracing nonauthorized data disclosure. In the field of information hiding, Zhang [19] uses the simple encryption algorithm of exclusive-OR operation by a stream cipher and embedded watermark information by flipping the 3 LSBs of each encrypted pixel. Zhang [20] further proposes a scheme which makes watermark extraction independent from image decryption. That means a user can extract data from the encrypted image directly. Yin et al. [9] propose a scheme with the multigranularity encryption algorithm and local histogram shifting digital watermark algorithm, which ensures larger embedding capacity and better embedding quality. But chaotic-based scrambling encryption is widely used in

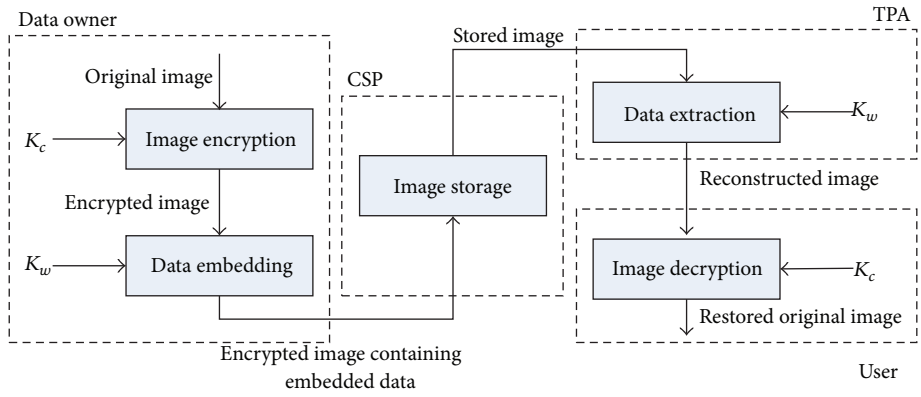


Figure 1: Sketch of the proposed scheme.

image encryption. The common encryption algorithms are one-dimensional logistic map, two-dimensional Smale and Henon map, and three-dimensional Lorenz map. The logistic map-based chaotic cryptography is a simple nonlinear model, but it has complex dynamics, which has good effect and fast speed.

In our scheme, we combine encryption technology with watermark technology. Data owner encrypts the image before transmission. CSP embeds some additional message into the encrypted image without knowing the original image content. TPA is required to extract the watermark from the encrypted image. A user can first decrypt the encrypted image containing watermark information with the decryption key and then extract the embedded watermark from the decrypted version with the extraction key. The transmission of encryption keys is assumed to be secure and is not discussed here. Here the logistic map-based chaotic cryptography method is used to permute the positions of the image pixels in the spatial domain. So the histogram of the encryption version is the same as the original image. The histogram statistical property makes the encryption method suitable for embedding watermark information with local histogram shifting digital watermark algorithm [9]. And this is a blind fragile watermark algorithm. The extraction of the watermark does not need the original image and original watermark information. Its error-free decryption can be used for military, remote sensing, and medicine data.

3. Proposed Scheme

In this section, we first analyze the framework of the system and then give the main steps of our scheme.

3.1. System Model. We first give the sketch of the proposed scheme in Figure 1. Then four parties in the scheme are described as follows.

(i) Data owner encrypts an original image with an encryption key K_c, computes a verification information as watermark information W for the encrypted image, embeds W to the encrypted image with the embedding key K_w, and upload the encrypted image to CSP.

(ii) CSP stores the watermark-embedded encrypted image.

(iii) TPA extract the watermarking information W' with K_w in the encrypted domain to verify the integrity and reconstructed the image if it is integrated.

(iv) Data user receives the reconstructed image from TPA and exactly decrypts the data to the original image with the decryption key K_c.

3.2. Main Steps of Proposed Scheme. The proposed scheme contains four modules: image encryption, watermarking embedding, watermarking extraction, and image decryption. The main steps of the proposed scheme are shown as follows.

3.2.1. Image Encryption. Data owner creates an original image I. Assume I is a gray image sized $M \times M$ pixels in uncompressed format.

The process of image encryption is as follows.

(i) Connect the jth row to the $(j-1)$th row, where $j = 2, 3, \ldots, M$, and generate the sequence of length $M \times M$.

(ii) Generate a chaotic sequence of length $M \times M$ with

$$x_{n+1} = x_n \times \mu \times (1 - x_n), \quad (1)$$

where $x_n \in (0, 1)$, $n = 0, 1, 2, \ldots$, $\mu \in (0, 4]$. x_0 is the initial value. n is the number of iterations. μ is growth parameter and when $\mu \in [3.5699456, 4]$, the generated sequence is in the state of pseudorandom distribution.

(iii) Sort the chaotic sequence and record the location set.

(iv) Scramble the sequence of image with the same location set.

The encryption key K_c consists of x_0 and μ. The encrypted image E is generated. This algorithm is simple and has good performance. The algorithm keeps the image histogram statistical properties.

3.2.2. Watermarking Embedding. The embedded watermarking information should be unpredictable and random. Arnold transforming or chaotic-based encryption can be used in this paper to improve the security of image watermarking algorithm. The above-mentioned encryption algorithm preserves the same image histogram statistical properties. Therefore, local histogram shifting watermarking algorithm is suitable for embedding data into the encrypted image [6].

When data owner embeds watermarking information W into the encrypted image E, the steps are as follows.

(i) Divide the encrypted image E into blocks $\{B_i\}_{i=1}^{N}$ of pixels in the size of $m \times m$. Two basic pixels $b_{i,L}$ and $b_{i,R}$ are randomly selected in each block B_i with the seed of random permutation k.

(ii) Calculate the difference $d_i = |b_{i,L} - b_{i,R}|$ to estimate the smoothness of each block. Blocks with smaller d_i are smoother than blocks with larger d_i. Blocks with smaller d_i have higher priority to be chosen for carrying data.

(iii) Determine the two peaks $(p_{i,L}, p_{i,R})$ in each block with

$$\begin{aligned} p_{i,L} &= \min\left(b_{i,L}, b_{i,R}\right) \\ p_{i,R} &= \max\left(b_{i,L}, b_{i,R}\right). \end{aligned} \tag{2}$$

If $p_{i,L} = p_{i,R}$, $p_{i,R} = p_{i,R} + 1$.

(iv) Saturated pixels q ($q = 0$ or $q = 255$) have to be preprocessed by modifying one grayscale unit. Then they will be recorded in a location map L to avoid saturated pixels from overflow or underflow during embedding process. Scan the pixels block by block and append bit "1" to L when $q \in \{1, 254\}$. Then append bit "0" to L when $q \in \{0, 255\}$ and modify q to q' using

$$q' = \begin{cases} 1, & q = 0 \\ 254, & q = 255 \\ q, & \text{otherwise.} \end{cases} \tag{3}$$

The embedding capacity of each block is the number of pixels whose values are equal to peak points in each block.

(v) Embedded information S consists of the location map L and the histogram information H of the image. Scan the nonbasic pixels in each block. If the scanned pixel r is valued $p_{i,L}$ or $p_{i,R}$, a bit $s \in \{0, 1\}$ from S will be embedded. Modify r to r' as

$$r' = \begin{cases} r - 1, & r < p_{i,L} \\ r - s, & r = p_{i,L} \\ r, & p_{i,L} < r < p_{i,R} \\ r + s, & r = p_{i,R} \\ r + 1, & r > p_{i,R}. \end{cases} \tag{4}$$

The encrypted image $\widehat{E}$ with embedded data is obtained. The embedding key K_w consists of the parameter m, $|L|$, $|H|$, and the seed k. The data owner outsources the encrypted image $\widehat{E}$ with embedded watermarking information to the cloud. Then the watermark embedding key K_w is transferred to TPA and the decryption key K_c is shared with the legal users.

3.2.3. Watermarking Extraction and Data Auditing. TPA extracts the watermarking information W' with the extraction key K_w before the user downloads the data from the cloud. The watermarking information can only be extracted from the encrypted domain by TPA that ensures data privacy.

This blind extracting algorithm is shown as follows.

(i) Divide the image $\widehat{E}$ into blocks $\{B_i'\}_{i=1}^{N}$ of pixels in size $m \times m$. Determine the basic pixels $b_{i,L}'$ and $b_{i,R}'$ in each block B_i'.

(ii) The difference $d_i' = |b_{i,L}' - b_{i,R}'|$ is calculated to estimate the smoothness of each block. Blocks with smaller d_i' have higher priority to be chosen for extracting data.

(iii) Determine the two peaks $(p_{i,L}', p_{i,R}')$ in each block with

$$\begin{aligned} p_{i,L}' &= \min\left(b_{i,L}', b_{i,R}'\right), \\ p_{i,R}' &= \max\left(b_{i,L}', b_{i,R}'\right). \end{aligned} \tag{5}$$

If $p_{i,L}' = p_{i,R}'$, $p_{i,R}' = p_{i,R}' + 1$.

(iv) Scan nonbasic pixels in each block B_i'. If the scanned pixel is r', embedding information S will be extracted according to

$$s = \begin{cases} 0, & r' = p_{i,L}' \text{ or } r' = p_{i,R}' \\ 1, & r' = p_{i,L}' - 1 \text{ or } r' = p_{i,R}' + 1. \end{cases} \tag{6}$$

The extracted $|S|$ bits consist of location map L and histogram information H.

TPA verifies the data integrity after extracting the watermark information W'.

The auditing process is as follows.

(i) Scan nonbasic pixels in each block B_i'. If the scanned pixel is r', the restored pixel r can be computed by

$$r = \begin{cases} r' - 1, & r' > p_{i,R} \\ r', & p_{i,L} < r' < p_{i,R} \\ r' + 1, & r' < p_{i,L}. \end{cases} \tag{7}$$

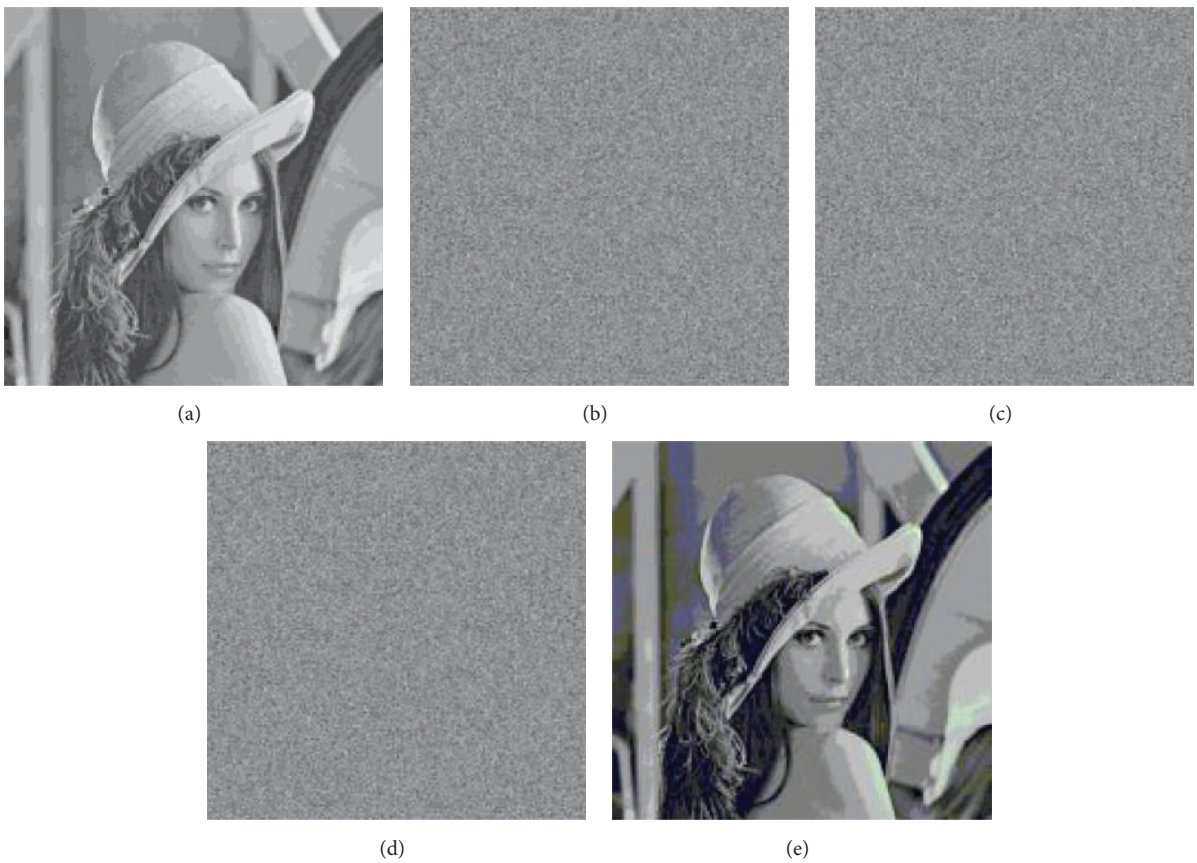


Figure 2: (a) Original image; (b) encrypted image; (c) encrypted image containing watermark; (d) reconstructed image; (e) decrypted image.

(ii) Restore the saturated pixels q with the location map L. If the pixel $q' \in \{1, 254\}$, extract a bit l from L. q is computed by

$$q = \begin{cases} 0, & l = 0,\ q' = 1 \\ 1, & l = 1,\ q' = 1 \\ 255, & l = 0,\ q' = 254 \\ 254, & l = 1,\ q' = 254. \end{cases} \tag{8}$$

The reconstructed encrypted image E' is generated.

(iii) Compute the histogram information H' of the image E'. Then compute the Euclidean distance by (9) and compare the value with the preset threshold θ:

$$D\left(H, H'\right) = \text{sqrt}\left(\sum_{i=1}^{n}\left(H\left[i\right] - H'\left[i\right]\right)^2\right), \tag{9}$$

where $H = (H(1), H(2), \ldots, H(N))$, $H' = (H'(1), H'(2), \ldots, H'(N))$.

If the value $D(H, H') < \theta$, the watermark information is correct and the data is verified to be integrated.

3.2.4. Image Decryption. The legal users can decrypt the reconstructed encrypted image E' using the decryption key K_c and can also obtain the original image I. The decryption process is as follows.

(i) Generate a chaotic sequence of length $M \times M$ with the decryption key K_c.

(ii) Sort the chaotic sequence and record the location set.

(iii) Scramble the sequence of image and restore a decrypted image with the location set.

Then the original image I is obtained by the legal users.

4. Experimental Results

To study the performance of the proposed scheme, MATLAB software 7 is used. The test image Lena of 8-bit gray level sized 512 × 512 pixels is selected as original image and it is shown in Figure 2(a). We use logistic map-based chaotic cryptography algorithm to generate an encrypted image (x_0 = 0.5, μ = 3.7), which is shown in Figure 2(b). The encrypted image containing watermarking information is shown in Figure 2(c). After the watermarking information is extracted by TPA, a reconstructed image is shown in Figure 2(d). Then the legal user can decrypt the reconstructed image. The decrypted image is shown in Figure 2(e).

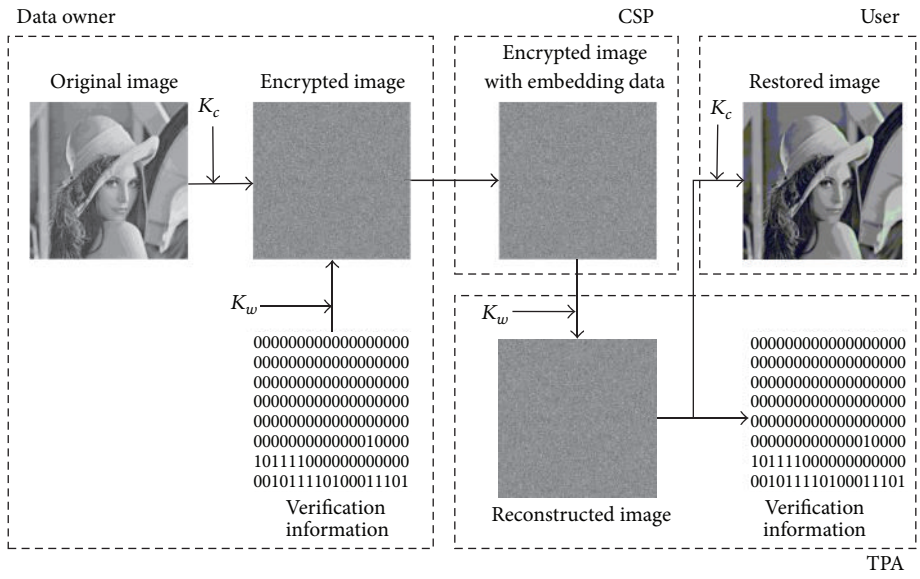


Figure 3: Experimental results of the proposed scheme.

Table 1: Payload bits and MSE.

Image	Payload bits (dB)	Payload bpp	MSE
Lena	2892	0.0110	0
Bridge	8234	0.0314	0
Aerial	4252	0.0162	0
Dollar	2892	0.0110	0

The experimental results of proposed scheme are shown in Figure 3.

The quality of encrypted image can be evaluated by Peak Signal-to-Noise Ratio (PSNR):

$$\text{PSNR} = 10 \times \log_{10}\left(\frac{255^2}{\text{MSE}}\right),$$
$$\text{MSE} = \frac{1}{M \times M}\sum_{i=1}^{M}\sum_{j=1}^{M}\left(I(i,j) - I'(i,j)\right)^2, \quad (10)$$

where I is the original image and I' is the image with watermark information. The size of image I is $M \times M$ pixels. The mean square error (MSE) can evaluate the error between the original image and decrypted image.

Table 1 lists the embedding payloads and MSEs for image Lena, bridge, aerial, and dollar without any attacks.

From Table 1, the MSEs between the decrypted version and the original image are 0. This means the encrypted image will be reconstructed error-free during watermark extraction and data auditing process if the data in cloud is not attacked. The payload is enough for embedding verification information.

In this paper, the watermark algorithm is fragile, which cannot resist any attacks. This can be used in military, remote sensing, and medicine images.

5. Conclusion and Future Work

In this paper, we propose a privacy-preserving and auditing-supporting outsourcing data storage scheme by using encryption and digital watermarking. The proposed scheme combines digital watermark technology with encryption methods for outsourcing data storage. And the scheme supports auditing service and privacy preserving. We adopt the logistic map-based chaotic cryptography algorithm for image encryption and local histogram shifting watermarking algorithm [6] for embedding data integrity verification information. This scheme has high authentication precision which can be used in high quality images.

In the future, we will add semifragile watermark to verify the integrity of images, which can resist some good image operations, such as JPEG compression. We can also apply some algorithms for the sake of supporting tamper localization and recovery.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by the NSFC (U1536206, 61232016, U1405254, 61373133, and 61502242), BK20150925, and PAPD fund.

References

[1] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in *Proceedings of the IEEE INFOCOM*, pp. 829–837, Shanghai, China, April 2011.

[2] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2015.

[3] Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Transactions on Communications*, vol. E98B, no. 1, pp. 190–200, 2015.

[4] K. Yang and X. Jia, "Data storage auditing service in cloud computing: challenges, methods and opportunities," *World Wide Web-Internet & Web Information Systems*, vol. 15, no. 4, pp. 409–428, 2012.

[5] N. Singh and S. Singh, "The amalgamation of digital watermarking & cloud watermarking for security enhancement in cloud computing," *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 4, pp. 333–339, 2013.

[6] D. Boopathy and M. Sundaresan, "Data encryption framework model with watermark security for data storage in public cloud model," in *Proceedings of the 8th International Conference on Computing for Sustainable Global Development (INDIACom '14)*, pp. 903–907, New Delhi, India, March 2014.

[7] J. Wang and S. Lian, "On multiwatermarking in cloud environment," *Concurrency Computation Practice and Experience*, vol. 24, no. 17, pp. 2151–2164, 2012.

[8] Y. Ren, J. Shen, J. Wang, J. Xu, and L. Fang, "Security data auditing based on multifunction digital watermark for multimedia file in cloud storage," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 9, no. 9, pp. 231–240, 2014.

[9] Z. Yin, B. Luo, and W. Hong, "Separable and error-free reversible data hiding in encrypted image with high payload," *The Scientific World Journal*, vol. 2014, Article ID 604876, 8 pages, 2014.

[10] N. Thiranant, M. Sain, and H. J. Lee, "A design of security framework for data privacy in e-health system using web service," in *Proceedings of the 16th International Conference on Advanced Communication Technology (ICACT '14)*, pp. 40–43, PyeongChang, South Korea, February 2014.

[11] R. L. de Souza, H. V. Netto, L. C. Lung et al., "SSICC: sharing sensitive information in a cloud-of-clouds," in *Proceedings of the 9th International Conference on Systems (ICONS '14)*, pp. 185–191, Nice, France, February 2014.

[12] G. Ateniese, R. Burns, R. Curtmola et al., "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 598–610, November 2007.

[13] A. Juels and B. S. Kaliski Jr., "Pors: proofs of retrievability for large files," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 584–597, November 2007.

[14] R. Sravan Kumar and A. Saxena, "Data integrity proofs in cloud storage," in *Proceedings of the 3rd International Conference on Communication Systems and Networks (COMSNETS '11)*, pp. 1–4, Bangalore, India, January 2011.

[15] Z. Hao, S. Zhong, and N. Yu, "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 9, pp. 1432–1437, 2011.

[16] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure provenance: the essential of bread and butter of data forensics in cloud computing," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communication Security (ASIACCS '10)*, pp. 282–292, Beijing, China, April 2010.

[17] W. N. Cheung, "Digital image watermarking in spatial and transform domains," in *Proceedings (TENCON '00)*, vol. 3, pp. 374–378, IEEE, Kuala Lumpur, Malaysia, 2000.

[18] S.-H. Wang and Y.-P. Lin, "Wavelet tree quantization for copyright protection watermarking," *IEEE Transactions on Image Processing*, vol. 13, no. 2, pp. 154–165, 2004.

[19] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.

[20] X. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 826–832, 2012.

[21] C.-L. Hsu, M.-R. Lee, and C.-H. Su, "The role of privacy protection in healthcare information systems adoption," *Journal of Medical Systems*, vol. 37, no. 5, article 9966, 2013.

[22] S. Haas, S. Wohlgemuth, I. Echizen, N. Sonehara, and G. Müller, "Aspects of privacy for electronic health records," *International Journal of Medical Informatics*, vol. 80, no. 2, pp. e26–e31, 2011.

Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2016, Article ID 6567146, 14 pages
http://dx.doi.org/10.1155/2016/6567146

Research Article

An Efficient Electronic English Auction System with a Secure On-Shelf Mechanism and Privacy Preserving

Hong Zhong,[1] Song Li,[1] Ting-Fang Cheng,[2] and Chin-Chen Chang[2]

[1]*School of Computer Science and Technology, Anhui University, Hefei, Anhui 230601, China*
[2]*Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan*

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

Received 16 December 2015; Accepted 24 March 2016

Academic Editor: Isao Echizen

With the rapid development of the Internet, electronic commerce has become more and more popular. As an important element of e-commerce, many Internet companies such as Yahoo! and eBay have launched electronic auction systems. However, like most electronic commerce products, safety is an important issue that should be addressed. Many researchers have proposed secure electronic auction mechanisms, but we found that some of them do not exhibit the property of unlinkability, which leads to the leakage of users' privacy. Considering the importance of privacy preservation, we have designed a new auction mechanism. Through symmetrical key establishment in the registration phase, all messages transmitted over the Internet would be protected and, meanwhile, achieve the property of unlinkability. The security analysis and performance analysis show that our protocol fulfills more security properties and is more efficient for implementation compared with recent works.

1. Introduction

With the development of network technology, more and more people are searching for information on the Internet. In 2013, the number of Internet users of the whole world reached 2.92 billion [1]. Now, people not only read the news and search for information on the Internet but also do business with others. Not surprisingly, e-commerce has grown rapidly in recent decades. As the founder and CEO of China's most famous e-commerce company Alibaba, Jack Ma was ranked as the second richest man in China in 2014. Many famous Internet companies launched electronic auction products several years ago such as Yahoo! and eBay. On these websites, users can play the role of not only auctioneer but also bidder after they register their accounts successfully. They can participate in the auction anytime and anywhere when their devices have access to the Internet.

Auctions can be classified into many types [2]. According to numbers of sellers and buyers, auctions can be classified as forward auctions and reverse auctions; according to the determinant of the winner, auctions can be classified as single attribute auctions and multiattribute auctions; and according to whether the bidding price is made open, we can classify auctions as sealed-bid auctions and open auctions. The sealed-bid auctions can be subdivided into sealed-bid first-price auctions and sealed-bid second-price auctions. In the first mode, all bidders submit their sealed bids to the auctioneer of an auction at the same time, and the auctioneer or arbiter of the auction secretly calculates the highest price. After that, the auctioneer or arbiter announces the winner as the owner with the highest price bid. In a sealed-bid second-price auction (also called Vickrey auction), the highest price bidder wins the auction but only needs to pay the second highest price. The designer of the sealed-bid second price auction thinks that everyone will submit their bidding price rationally in this mode; however, this method confronts a lot of problems in practice, such as bidder collusion. Open auctions can be further classified into English auctions and Dutch auctions. In an English auction, the auctioneer changes the current price dynamically, with the bidding price increasing. This means that, if one bidder's bid price is higher than the current price, then the auctioneer uses this price as the new current price and waits for someone to offer a higher price than the current price. If someone submits a higher price, the current price is changed simultaneously; if no one offers a higher price when the auction is closed, the

owner of the current price bid wins the auction. A Dutch auction is bidding process contrary to the English auction. In a Dutch auction, the auctioneer decreases the current price until one bidder can afford it and is the winner of the auction.

A lot of problems arise when we introduce the auction process in reality on the Internet. An important issue is the information security of the system. As an Internet product is connected with money and goods, users are concerned with the safety of the online auction. Many researchers have proposed their protocols to solve security problems in auction systems. In 1999, based on millionaires' problem, Cachin [3] proposed a private bidding and auction scheme with an oblivious third party. In 2000, Nguyen and Traoré [4] used a group signature to protect bidders' anonymity. However, the huge computational cost and the special authority of group managers still cannot be resolved. In 2001, Omote and Miyaji [5] applied a bulletin board to overcome these problems, but their scheme does not publish the winner's information at the end. In 2003, C.-C. Chang and Y.-F. Chang [6] proposed an anonymous English auction protocol to ensure that bidders can bid arbitrarily. In 2005, Jiang et al. [7] considered that the scheme in [6] cannot protect against man-in-the-middle attacks. In 2006, Y.-F. Chang and C.-C. Chang [8] proposed another anonymous auction scheme to resolve the problem in [6]. In 2005, Suzuki and Yokoo proposed safety problem in the multiattribute auction systems [9]. In 2007, Shih et al. proposed a privacy preserving multi-item auction mechanism with a shared key chain [10]. In 2008, Parkes et al. used a homomorphic cryptograph to achieve bid privacy in multi-item auction [11]. In 2009, Xiong et al. [12] proposed an anonymous auction scheme based on the ring signature. However, like the group signature-based protocol in [4], huge computational cost is an essential problem that should be resolved. In 2012, Xiong et al. [13] proposed another protocol based on revocable ring signature to solve the problem of high computational cost in [12] and added a dispute section.

In 2013, Chang et al. [14] pointed out that [13] is vulnerable to denial of service attacks and designed an ECC-based protocol which provides a secure on-shelf phase. Unfortunately, we found that Chang et al.'s proposal [14] does not provide unlinkability. In [14], users' privacy can be leaked by monitoring the communication of system and linking the captured information. Although some important data such as public key and certificate are encrypted before transmission, unencrypted messages such as on-shelf information can link different packets together. For example, in the on-shelf phase, although attackers cannot calculate the true public key of the auctioneer using an encrypted message transmitted from the auctioneer and the published anonymous public key on website, attackers still can link transmitted packets and website information together through on-shelf information. This is because the on-shelf information is transmitted in plaintext. On the other hand, bidders' privacy is in more serious danger during the bidding phase. At the end of the bidding phase, the agent center (*AC*) publishes the information with the bidder's true public key on its website. Attackers can link this public key with a certain IP address and know what the user bids on.

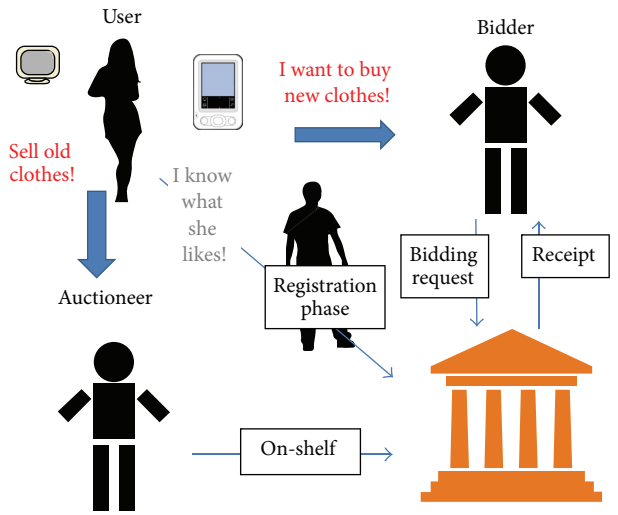


Figure 1: Privacy problem in electronic auction system.

Then, attackers can trace this public key owner's transaction history.

Until now, many researchers have stressed the importance of privacy preservation in online auctions [15–17]. Hence, in order to overcome the privacy problem in [14], in this paper, we provide a new English auction system with privacy preservation. The overview of the auction system architecture is shown in Figure 1. Using the same system mode with Chang et al.'s proposal [14], our system has 5 phases: the registration phase, on-shelf phase, bidding phase, product-claim phase, and dispute phase. We utilize a trusted third party as the *AC* of our system, a well-meaning role that will not initiate attacks. However, the safety of the *AC*'s database cannot be completely guaranteed. We do not consider the case of hackers controlling the *AC*, but hackers may get access to read data in the *AC*. The only data we consider to be completely safe is the private key of *AC*. Besides this, any leaked data cannot do serious harm to the safety of the whole system. In addition, we built our protocol on Elliptic Curve Cryptosystem. According to the analysis in Section 8, our proposal has an efficiency advantage over the original proposal. In our scheme, messages transmitted between any two entities are encrypted with a symmetrical encryption algorithm such as AES. By utilizing this method, monitors cannot link different messages together. Security analysis shows that our proposal provides more security properties than the original one.

The rest of this paper is organized as follows. In Section 2, we introduce the Elliptic Curve Cryptosystem and ECDLP (elliptic curve discrete logarithm problem). Subsequently, we briefly review Chang et al.'s auction protocol [14] in Section 3. In Section 4, we talk about network model, adversary model, and properties that an auction mechanism needed. We describe our proposal in detail and use BAN logic to analyze authentication accuracy of on-shelf phase and bidding phase in Sections 5 and 6, respectively. In Sections 7 and 8, we compare the security and efficiency of our protocol with related schemes.

2. Preliminaries

In this part, we introduce some basic ideas of Elliptic Curve Cryptosystem including the definition of elliptic curve, basic operations on the Elliptic Curve, and ECDLP on ECC that can be used to construct cryptographic algorithms.

2.1. Elliptic Curve Cryptosystem. Elliptic Curve Cryptosystem (ECC) is an asymmetrical cryptosystem. It was independently proposed by Miller [19] and Koblitz [20] in 1985 and 1987. Compared to RSA, ECC can achieve the same security requirement with a shorter key-length [21]. Hence, it has been widely used in many cryptographic schemes recently.

An elliptic curve [22, 23] is defined over a finite field F_p by equation $E_p(a,b) : y^2 = x^3 + ax + b$, where p is a large prime and $4a^3 + 27b^2 \neq 0 \bmod p$. The points on this elliptic curve form a cyclic group. Addition in this group is defined as if points $P, Q, R \in E_p(a,b)$ on one line, and then $P + Q + R = O$ (O is infinite point). Given an integer $s \in F_p^*$ and a point $P \in E_p(a,b)$, the multiplication operation $s \cdot P$ over $E_p(a,b)$ is defined as $P + P + \cdots + P$ in s times. If P is symmetrical with P' on the x-axis, then $P + P' = O$. Furthermore, point P is a base point with an order n if and only if $n \cdot P = O$.

2.2. ECDLP. Every cryptosystem has its own difficult problem, such as the integer factorization used in RSA. The most important difficult problem in ECC is the elliptic curve discrete logarithm problem (ECDLP) [24]. Based on ECDLP, we can develop many other difficult problems, such as the computational Diffie-Hellman problem (CDLP) and the elliptic curve factorization problem (ECFP). In our scheme, we will use ECDLP.

Definition 1 (elliptic curve discrete logarithm problem (ECDLP)). Given two points P and Q over $E_p(a,b)$, it is very hard to find an integer $s \in F_p^*$ such that $Q = sP$.

3. Related Work

In this part, we describe Chang et al.'s proposal [14] briefly and analyze the security and privacy preserving in it.

Chang et al.'s protocol includes five phases: registration phase, on-shelf phase, bidding phase, product claiming phase, and dispute phase. A new user should register to an agent center (AC) and AC will issue a certificate for him/her such that he/she can play the role of both auctioneer and bidder. If a registered user wants to initiate an auction, he/she should send some basic information about auction (i.e., product identity, basic price of the auction, and deadline) to AC via the on-shelf phase. Upon checking the legality of user's identity and the information of the new auction, AC publishes an advertisement on its website (BB_{AC}) such that anyone can read it after the auction has started. If someone wants to bid for the product of this auction via the bidding phase, he/she can send bidding information to AC. After AC checked the legality of bidder's identity and the bidding price is higher than current price, AC should change the value of current price to the bidding price and publish the bidder's public key and bidding price on BB_{AC}; else, AC ignores this bid and waits for new bid. After the auction deadline, AC selects the bidder with highest price as the winner, sends a receipt to the winner, and publishes final result on BB_{AC} such that anyone can read and verify it. After getting the receipt, the winner can claim the product from the auctioneer by showing the receipt via the product claiming phase. Then, the auctioneer can verify the correctness of this receipt and send the product to the user. With regard to the dispute phase, if a user takes action illegally, the user trading with him/her can submit dispute information to AC and wait for the arbitration of AC. If illegal action existed, AC can trace the identity of malicious user and punish him/her.

The notations used in Chang et al.'s protocol are as follows and the details of their protocol are described in the following except of the dispute phase:

AC: a trusted agent center.

B_Z: a Bidder Z.

A_Z: an auctioneer Z.

ID_Z: the identity of Z.

SK_{AC}: AC's private key.

PK_Z: Z's public key.

PW_Z: Z's password.

$CERT_Z$: the certificate of PK_Z signed by AC.

BB_{AC}: the Bulletin board of AC.

P: the base point of ECC group over a finite field F_p with order p, where p is a large prime.

$(\cdot)_x$: an operation using x-coordinate in ECC.

$(\cdot)_y$: an operation using y-coordinate in ECC.

$E_k[\cdot]/D_k[\cdot]$:AES-based encryption/decryption with key k.

$f(\cdot)$: a secure one-way hash function.

3.1. Registration Phase. If a user wants to play the role of an auctioneer or a bidder in the auction, he/she should register an account at AC by the following steps.

(1) User chooses ID_u, $PW_u \in Z_p^*$ and a random number $r_u \in Z_p^*$, and computes his/her public key $PK_u = PW_u \cdot P \bmod p$. Then, the user calculates $R_0 = r_u \cdot P \bmod p$, $R_1 = ID_u + (r_u \cdot PK_{AC})_x \bmod p$, and $R_2 = PK_u + r_u \cdot PK_{AC} \bmod p$ and sends (R_0, R_1, R_2) to AC.

(2) AC retrieves user's identity $ID_u = R_1 - (SK_{AC} \cdot R_0)_x \bmod p$ and public key $PK_u = R_2 - SK_{AC} \cdot R_0 \bmod p$. Subsequently, AC chooses a random number $r_{AC} \in Z_q^*$ calculates $R_0' = r_{AC} \cdot P \bmod p$ and $CERT_u = r_{AC} + (PK_u)_x \cdot SK_{AC} \cdot ID_u \bmod p$, and sends them to the user. User checks the correctness of equation $CERT_u \cdot P - R_0' ? \equiv (PK_u)_x \cdot ID_u \cdot PK_{AC} \pmod p$ to verify the validation of this certificate. If valid, then user stores it with R_0'.

3.2. On-Shelf Phase. If a registered user wants to host an auction, then he/she becomes an auctioneer A_j and performs the following.

(1) A_j chooses a random number $r_1 \in Z_p^*$ and a nonce $n_1 \in Z_p^*$ and calculates $O_0 = r_1 \cdot P \bmod p$, $O_1 = (PK_{A_j} + r_1 \cdot PK_{AC}) \bmod p$, $K_{aa} = (PW_{A_j} \cdot PK_{AC} \cdot n_1)_x \bmod p$, and $O_2 = E_{K_{aa}}(R_0' \parallel CERT_{A_j})$. A_j generates auction information $AI = (pid_k \parallel BasicPrice \parallel Deadline)$, that is, the identity of a product, basic price of this auction, and the deadline of this auction. Finally, A_j computes $Sig = (r_1 + AI \cdot PW_{A_j} \cdot n_1) \bmod p$ and sends $(ID_{A_j}, n_1, O_0, O_1, O_2, AI, Sig)$ to AC.

(2) AC retrieves A_j's public key $PK_{A_j} = O_1 - SK_{AC} \cdot O_0$. Then, AC computes the symmetrical key shared with A_j : $K_{aa} = (SK_{AC} \cdot PK_{A_j} \cdot n_1)_x \bmod p$. AC can decrypt O_2 with K_{aa} to extract A_j's certificate. AC should check the correctness of the certificate by verifying if equation $CERT_{A_j} \cdot P - R_0' \equiv (PK_{A_j})_x \cdot ID_{A_j} \cdot PK_{AC} (\bmod p)$ holds and the integrity of AI by verifying if equation $Sig \cdot P - O_0 \equiv PK'_{A_j} \cdot AI (\bmod p)$ holds where $PK'_{A_j} = n_1 \cdot PK_{A_j} \bmod p$. If both are valid, AC generates a unique gid for A_j's product. AC then computes a verifier $V = h(gid \parallel K_{aa})$ and sends (V, gid) to A_j.

(3) After A_j received message from AC, he/she computes and checks if $V = h(gid \parallel K_{aa})$. If the equation holds, A_j responds *Approval* message to AC. Upon receiving *Approval* from A_j, AC publishes $(gid, ID_{A_j}, AI, O_0, PK'_{A_j}, Sig, N_1)$ on BB_{AC} and anyone can verify this message by checking equation $SigP - O_0 \equiv PK'_{A_j} \cdot AI (\bmod p)$.

3.3. Bidding Phase. If someone is interested in this auction, he/she could play the role of a bidder B_i and send bidding message to AC by the following.

(1) B_i chooses a random number $r_2 \in Z_q^*$ and computes $D_0 = r_2 \cdot P \bmod p$, $D_1 = (PK_{B_i} + r_2 \cdot PK_{AC} \bmod p$, $K_{ba} = (PW_{B_i} \cdot PK_{AC} \cdot (D_0)_y)_x \bmod p$, and $D_2 = E_{K_{ba}}(R_0' \parallel CERT_{B_i} \parallel ID_{B_i})$. B_i generates bidding message $BM = (gid \parallel price)$ and signs it as $BSig_{B_i} = (r_2 + BM \cdot PW_{B_i}) \bmod p$. B_i sends $(D_0, D_1, D_2, BM, BSig_{B_i})$ to AC.

(2) Upon receiving these messages, AC retrieves B_i's public key $PK_{B_i} = D_1 - D_0 \cdot SK_{AC} \bmod p$ and computes $K_{ba} = (SK_{AC} \cdot PK_{B_i} \cdot (D_0)_y)_x$. Then, AC decrypts D_2 using K_{ba} to extract B_i's certificate. AC can verify the correctness of $CERT_{B_i}$ by checking equation $CERT_{B_i} \cdot P - R_0' ? \equiv (PK_{B_i})_x \cdot ID_{B_i} \cdot PK_{AC} (\bmod p)$. If it is valid, AC checks the correctness of BM by checking equation $BSig_{B_i} \cdot P - D_0 ? \equiv PK_{B_i} \cdot BM (\bmod p)$. If it is valid too, AC further checks *Deadline* and *BasicPrice* of the auction by using gid to search the corresponding information on its website. If the auction has closed or the *price* is not larger than *BasicPrice*, AC rejects this bid; otherwise, AC generates a unique transaction identity tid for B_i and chooses a random number $r_3 \in Z_q^*$. AC then computes $D_3 = r_3 \cdot P \bmod p$, $BM' = (tid \parallel BM)$, $Rec = r_3 + (BM' \cdot SK_{AC})_x \bmod p$, and $D_4 = E_{K_{ba}}(tid \parallel D_3 \parallel Rec)$. Finally, AC updates *BasicPrice* = *price*, publishes $(tid, price, date)$ on BB_{AC}, stores $(tid, price, date, D_0, BM, BSig_{B_i}, PK_{B_i}, D_2)$ in database, and sends D_4 to bidder.

(3) After B_i received the message from AC, B_i can extract the transaction information $(tid \parallel D_3 \parallel Rec)$ from D_4. Then, B_i checks the validation of receipt Rec by verifying $Rec \cdot P - D_3 ? \equiv (tid \parallel BM) \cdot PK_{AC} \bmod p$. If the equation holds, B_i stores Rec and D_3.

(4) After deadline, AC sets the winner as the bidder with the highest price. AC then publishes the winner's information on BB_{AC} including several items $(tid, price, date, D_0, bid, BSig_{B_i}, PK_{B_i}, D_2)$ such that anyone can verify its validity by checking $BSig_{B_i} \cdot P - D_0 ? \equiv PK_{B_i} \cdot BM (\bmod p)$.

3.4. Product Claiming Phase. After the auction deadline, the winner can claim the product from the auctioneer by showing his/her receipt and performing the following steps. After the auctioneer checked the correctness of receipt, the auctioneer should send the product to the winner.

(1) B_i selects a random number r_c and a nonce n_2 from Z_q^*, computes $C_0 = r_c \cdot P \bmod p$, $C_1 = (Rec + r_c \cdot n_2 \cdot PK'_{A_j}) \bmod p$, and $C_2 = (D_3 + r_c \cdot n_2 \cdot PK'_{A_j}) \bmod p$, and sends $(gid, C_1, C_2, C_0, n_2)$ to A_j.

(2) Once receiving B_i's claiming request, A_j retrieves $Rec = (C_1 - (C_0 \cdot n_2 \cdot n_1 \cdot PW_{A_j})_x) \bmod p$ and $D_3 = (C_2 - C_0 \cdot n_2 \cdot n_1 \cdot PW_{A_j}) \bmod p$. Then, A_j verifies $Rec \cdot P - D_3 ? \equiv (tid \parallel gid \parallel price) \cdot PK_{AC} \bmod p$. If the verification holds, A_j is convinced that B_i is the actual winner and sends the product to B_i.

3.5. Security Defect of Chang et al.'s Protocol. According to our analysis, we found that Chang et al.'s protocol meets most of the security requirements of an auction protocol except for unlinkability. In Chang et al.'s protocol, although some important data such as users' public keys and certificates are encrypted before being transferred, unencrypted messages such as the on-shelf information still can be linked together with different packets. Figures 2 and 3 show the linkage between transferred messages and published messages on AC's website. The solid border represents the message transferred on the Internet, and the dotted border represents the message published on AC's website.

From Figures 2 and 3, we can see that these messages can be linked together through certain parameters. For example, in the on-shelf phase (Figure 2), although attackers cannot calculate the public key or certificate of an auctioneer A_j from

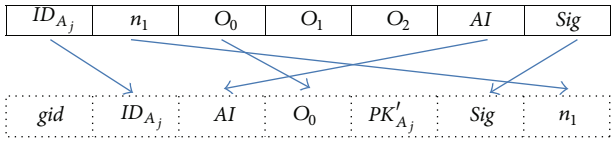


Figure 2: Linkage of messages in on-shelf phase.

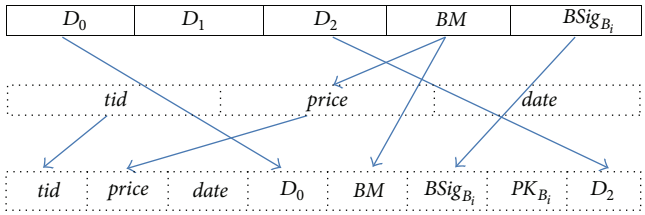


Figure 3: Linkage of messages in bidding phase.

the encrypted message (solid border) based on the definition of ECDLP and the security assumption of underlying AES algorithms, attackers can still find the relevance of transferred messages (solid message) and published messages (dotted message) to analyze the specific user's privacy. As shown in Figure 2, we find that ID_{A_j}, n_1, AI, O_0, and Sig are linkable. Obviously, an adversary can link the captured message on the Internet with the published message if they have the same parameters ID_{A_j}, n_1, AI, O_0, andSig. Furthermore, the adversary can decide that if these messages come from the specific IP address. If so, the adversary obtained this IP address owner's identity ID_{A_j}.

On the other hand, as mentioned previously, bidders' privacy is in more serious danger in the bidding phase. At the end of the bidding phase, the *AC* publishes the information with the winner's true public key on *AC*'s website. As shown in Figure 3, similar to Figure 2, attackers have the ability to determine the IP addresses of captured messages on the Internet and to link the specific transferred encrypted bidding information with the published message (the winner's information). Consequently, attackers would find the relevance of the winner's public key and the specific IP address. If this user continues to anticipate other auctions, his/her transaction history will be traced. For example, an adversary used Sniffer (a network tool to capture the packets transferred in local network) to capture packets transferred in the local network of his/her office environment and found that some packets are transferred to the auction server (*AC*). After analyzing the important parts of these packets and comparing them with the messages published on *AC*'s website, the adversary can know what the specific IP address owner had bidden for or sold out. Obviously, the adversary can easily know who uses the IP address in this office if he/she wishes.

According to the aforementioned analyses, we can see that the main problem that results in privacy disclosure is that some published data is transferred in plaintext before such that an adversary can easily link them together. Hence, in our proposal, we establish a shared symmetrical key between each user and the *AC* to encrypt the message transferred on the Internet to avoid linkability. By using symmetrical encryptions and fewer ECC operations, our proposal can not only improve the security but also reduce the system cost. In the registration phase, we connect a user's identity and the corresponding symmetrical key so that the *AC* can compute the shared symmetrical key easily with the user's identity. Besides, each user's identity is encrypted with the *AC*'s public key so it cannot be revealed without the knowledge of *AC*'s private key. In our proposal, we fulfill the verifiability of all messages published on the *AC*'s website; however, Chang et al.'s proposal only partially achieved this feature.

4. System and Adversary Model

In this section, we describe the definitions of our network model, adversary model, and requirements as follows before introducing our proposed scheme.

4.1. Network Model. We consider a network composed of an agent center (*AC*) and users. The *AC* is a trusted third party that undertakes most tasks of system running, such as user registration, generation of users' certificates, and product on-shelf. With regard to the application of the trusted third party, it is widespread in e-commerce systems [25–28]. The *AC* is also an arbiter of an auction and has the right to determine who the winner is or punish the illegal user's identity. Hence, we think that the *AC* has large capability of computation and storage. The *AC* should maintain a database on its machine to store users' data. Herein, we do not consider this database to be completely secure. Hackers may have methods to access it, but such actions will not destroy the security of the whole system. Furthermore, the *AC*'s private key should be kept in a trusted place such as a bank security box.

On the other hand, users use their devices such as computers or mobile devices to communicate with the *AC* through the Internet. When a user wants to connect with the *AC*, he/she should type in a password or insert a smart card with a password into the machine. The client in the user's machine erases the password after the user goes offline.

4.2. Adversary Model. In this paper, we assume that an adversary can launch a passive attack to monitor the communication channel of the system. The adversary has knowledge of the format of a packet, so he/she can analyze what content is included in the packet. We also think that the adversary has the ability to access the database stored in the *AC* and can read the *AC*'s website as the other users. In addition, some registered users may attack the system jointly.

4.3. Security Requirements. In order to provide a secure auction scheme, the following security properties are critical [29].

Anonymity. User's identification should be kept secret as the system is running. No one can obtain the user's identification from messages transmitted or published.

Easy Revocation. A user's right should be easily and correctly revoked in the auction system by the *AC*.

Fairness. All bidders should have the ability to verify whether their bids have indeed been included in the auction.

Nonrepudiation. Users including bidders and auctioneers cannot deny the actions that have been taken before. For example, if an auctioneer puts a product on-shelf, he/she cannot deny that he/she did it and off-shelf this product arbitrarily.

Onetime Registration. Any registered user can host or participate in any auction without reregistration at the *AC*.

Traceability. If some illegal actions are taken such as a nonpaying bid (*NPB*), the *AC* should have the ability to identify and publish the dishonest user's true identity anytime.

Unforgeability. No one can forge a valid message or impersonate any legal users to do illegal things.

Unlinkability. No one can link different messages together to trace a specific user's transaction history even if he/she does not know the true identification of the user.

Verifiability. All winning bids and product information published on a bulletin board should be publicly verifiable without revealing the bidders' identities.

5. Proposed Scheme

In this section, we propose a secure English auction system with privacy preservation. Our proposal consists of six phases: the system setup phase, registration phase, on-shelf phase, bidding phase, product claiming phase, and dispute phase. There are two kinds of participants in this system: the *AC* and the users. In the system setup phase, the *AC* inputs a security parameter and generates a set of system parameters. The *AC* then requests a certificate from an acknowledged certificate authority and publishes these data on its website. Anyone can get these data from the *AC*'s website. If a new user wants to register an account, he/she can connect with the *AC* during the registration phase. After registration, the user can either host or participate in an auction. If he/she wants to host an auction, he/she can on-shelf his/her products in the on-shelf phase; if he/she wants to bid for some on-shelved products, he/she can send the bidding information to the *AC* during the bidding phase. The *AC* should publish every bidder's bidding information except for the bidder's identity on its website such that anyone can read it. If a bidder wins the auction, he/she will receive a receipt and a session key shared with the auctioneer from the *AC*. Then, the winning bidder can obtain his/her product from the auctioneer during the product claiming phase. In our proposal, the *AC* has the capability of tracing the whole transaction and identifying the participants in an auction if necessary. If a dispute has occurred, anyone participating in an auction can submit the dispute request to the *AC* in the dispute phase. The notations used in our proposal are as follows and the details of our proposal are described in the following subsections:

B_i: a bidder i.

A_j: an auctioneer j.

AC: a trusted agent center.

κ: the system security parameter chosen by *AC*.

F_q: finite filed of order q.

E/F_q: an elliptic curve based on q order finite field.

G_q: a cyclic group of the elliptic curve.

P: the base point of ECC group.

$(\cdot)_x$: an operation using x-coordinate in ECC.

SK_{AC}: *AC*'s private key.

PK_{AC}: *AC*'s public key.

$Cert_{AC}$: the certificate of PK_{AC} signed by the certificate authority.

id_U: the identity of a user U.

pw_U: U's password.

N_n: a nonce.

P_U: U's partial key used for computing symmetrical key shared with *AC*.

K_{AU}: the symmetrical key of *AC* and a user.

K_{AB}: the symmetrical key of *AC* and a bidder.

K_{AA}: the symmetrical key of *AC* and an auctioneer.

$E_K(\cdot)$: an AES-based encryption with key K.

$D_K(\cdot)$: an AES-based decryption with key K.

$H(\cdot)$: a secure one-way hash function.

BB_{AC}: the bulletin board of *AC*.

5.1. System Setup Phase. Before the system is running, the *AC* chooses $SK_{AC} \in_R Z_q^*$ as the private key. The *AC* then inputs a security parameter $\kappa \in Z_q^*$ and generates a set of system parameters $\Omega = \{F_q, E/F_q, G_q, P, PK_{AC}, Cert_{AC}, H(\cdot)\}$ through the steps below:

(1) Choose a κ-bit prime q.

(2) Determine the tuple $\{F_q, E/F_q, G_q, P\}$; F_q is a finite filed modular q, E/F_q is an elliptic curve defined over F_q, G_q is a cyclic group defined over F_q, and P is the generator of G_q.

(3) Compute the *AC*'s public key $PK_{AC} = SK_{AC} \cdot P \bmod q$.

(4) Choose a cryptographic hash function $H(\cdot)$.

(5) Request a certificate $Cert_{AC}$ of PK_{AC} from the certificate authority.

Then, the *AC* publishes Ω on its online BB_{AC} and keeps SK_{AC} secret.

5.2. Registration Phase. If a user wants to host or participate in an auction, he/she must perform the following steps to register an account at the *AC*.

(1) The user obtains system parameters on BB_{AC} and checks the validation of the *AC*'s public key by $Cert_{AC}$. If PK_{AC} is invalid, he/she terminates the subsequent operations; otherwise, he/she chooses his/her password $pw_U \in Z_q^*$ and identity $id_U \in Z_q^*$ and generates a random number r_0 and a nonce N_1. Afterwards, he/she computes $R_0 = r_0 \cdot P \bmod q$, $P_U = pw_U \cdot P \bmod q$, $C = (r_0 \cdot PK_{AC} + P_U) \bmod q$, and $C_0 = ((r_0 \cdot PK_{AC})_x + M) \bmod q$ and sends (C, C_0, R_0) to *AC*, where $M = (id_U, N_1)$.

(2) Upon the receipt of the message sent from the user, the *AC* can extract P_U and M by computing $P_U = C - SK_{AC} \cdot R_0 \bmod q$ and $= (id_U, N_1) = C_0 - (SK_{AC} \cdot R_0)_x \bmod q$. The *AC* then checks the freshness of N_1. If N_1 is valid, the *AC* chooses a nonce N_2 and computes $RESPONSE = ((success, N_2) + (SK_{AC} \cdot P_U)_x) \bmod q$. The *AC* then sends *RESPONSE* to the user and stores (id_U, v_U) in its database, where $v_U = P_U \oplus H(SK_{AC})$.

(3) After receiving *RESPONSE*, the user retrieves $(success, N_2) = (RESPONSE - (PK_{AC} \cdot pw_U)_x) \bmod q$ and checks if *success* is contained in it. If so, the user checks the freshness of N_2. If both are valid, the user ascertains that he/she has registered successfully and then can either host or participate in an auction.

Note that each registered user has a partial key P_U and can share a key $K_{AU} \equiv (pw_U \cdot PK_{AC})_x \equiv (SK_{AC} \cdot P_U)_x (\bmod q)$ with the *AC*. For convenience, in the following, we use K_{AA} to imply the key shared between the *AC* and a user who plays as an auctioneer A_j, and A_j's partial key is P_{A_j}. In contrast, the keys of a user who plays as a bidder B_i are K_{AB} and P_{B_i}.

5.3. On-Shelf Phase. If a registered user wants to host an auction to sell some products, he/she becomes an auctioneer A_j and performs the following steps.

(1) A_j generates on-shelf information $\gamma = (ID_{\text{product}} \parallel Basic\_price \parallel Deadline)$ firstly, where ID_{product} refers to the identity of a product, *Basic_price* indicates the basic price of this auction (every bidder's bidding price should not be less than this value), and *Deadline* represents the deadline of this auction. Then, A_j computes $K_{AA} = (pw_{A_j} \cdot PK_{AC})_x \bmod q$. After that, A_j randomly chooses a nonce N_3 and calculates $e_1 = E_{K_{AA}}(\gamma, N_3, id_{A_j})$, $h_1 = H(e_1, K_{AA})$, $C_1 = (h_1 \cdot PK_{AC})_x + id_{A_j} \bmod q$, and $R_1 = h_1 \cdot P \bmod q$. Then, A_j sends (C_1, R_1, e_1) to the *AC*.

(2) The *AC* retrieves id_{A_j} from C_1 by computing $id_{A_j} = C_1 - (R_1 \cdot SK_{AC})_x \bmod q$. Then, the *AC* searches its database to extract the corresponding P_{A_j} ($P_{A_j} = H(SK_{AC}) \oplus v_{A_j}$) of id_{A_j} and calculates $K_{AA} = (SK_{AC} \cdot P_{A_j})_x \bmod q$. The *AC* extracts (γ, N_3, id_{A_j}) by decrypting operation $D_{K_{AA}}(e_1)$. The *AC* checks whether the identity id_{A_j} in e_1 is equal to that retrieved from C_1. If they are not equal, the *AC* may confront a replay attack, and the *AC* should terminate sequent operations immediately; otherwise, the *AC* checks the freshness of N_3. If both are valid, the *AC* generates a unique on-shelf identity $ID_{\text{on-shelf}}$ for A_j and sends $E_{K_{AA}}(ID_{\text{on-shelf}}, id_{A_j}, N_4)$ to A_j, where N_4 is a nonce.

(3) After receiving the message from the *AC*, A_j retrieves $(ID_{\text{on-shelf}}, id_{A_j}, N_4)$ by decryption with key K_{AA}. A_j checks the integrity of the message by checking if a correct id_{A_j} is in it and checks the freshness of this message by checking N_4. If both are valid, the auctioneer then sends *APPROVAL* to *AC*.

(4) Upon receiving *APPROVAL* from A_j, the *AC* chooses a random number $r_1 \in_R Z_q^*$ and computes $R_2 = r_1 \cdot P \bmod q$ and $S_1 = r_1^{-1} \cdot (M_1 - SK_{AC} \cdot (R_2)_x) \bmod q$, where $M_1 = (\gamma, ID_{\text{on-shelf}}, id_{A_j})$. Waiting τ minutes ($\tau \in_R (0, t)$), the *AC* publishes (M_1, R_2, S_1) on BB_{AC}. Once the *AC* publishes the on-shelf information, the auction has been held, and anyone can verify the validation of the auction information by checking the correctness of equation $S_1 \cdot R_2 + (R_2)_x \cdot PK_{AC}? \equiv M_1 \cdot P \bmod q$.

5.4. Bidding Phase. If a bidder B_i wants to bid for an on-shelved product, he/she should send the bidding information to the *AC*, including the corresponding identity $ID_{\text{on-shelf}}$ of the product and a bidding price *price*. We denote $(ID_{\text{on-shelf}} \parallel price)$ as η here.

(1) B_i computes $K_{AB} = pw_{B_i} \cdot PK_{AC} \bmod q$ firstly. Then, B_i computes $e_2 = E_{K_{AB}}(\eta, N_5, id_{B_i})$, $h_2 = H(e_2, K_{AB})$, $C_2 = ((h_2 \cdot PK_{AC})_x + id_{B_i}) \bmod q$, and $R_3 = h_2 \cdot P \bmod q$, where N_5 is a nonce. Then, B_i sends (C_2, R_3, e_2) to the *AC*.

(2) After receiving the bid sent from B_i, the *AC* can retrieve id_{B_i} from C_2 by computing $id_{B_i} = C_2 - (R_3 \cdot SK_{AC})_x \bmod q$. Then, the *AC* searches the id_{B_i} in the database. If the id_{B_i} does not exist, the *AC* terminates the sequent operation; otherwise, the *AC* retrieves the corresponding P_{B_i} by computing $P_{B_i} = H(SK_{AC}) \oplus v_{B_i}$ and computes the symmetrical key shared with B_i as $K_{AB} = SK_{AC} \cdot P_{B_i} \bmod q$. The *AC* decrypts e_2 with K_{AB} to extract (η, N_5, id_{B_i}). The *AC* can check the integrity of this bid by seeing if id_{B_i} is equal to the one retrieved from C_2. If they are not equal, the *AC* may confront a replay attack, and the *AC* should terminate the sequent operations; otherwise, the *AC* checks the freshness of N_5. If both are valid, the *AC* uses $ID_{\text{on-shelf}}$ in η to search the corresponding auction information in BB_{AC}.

(3) After finding the corresponding auction, the *AC* compares *price* in η with *Basic_price* of the auction. If *price* is not larger than *Basic_price*, the *AC* ignores

this bid and returns to the waiting state; otherwise, the AC updates $Basic\_price = price$ and generates a unique identity ID_{trans} for this transaction. Then, the AC generates a nonce N_6 and computes $k_{dis} = H(SK_{AC} \parallel N_6)$ and $dis = E_{k_{dis}}(C_2, R_3)$. Let $(ID_{\text{trans}}, date, price, dis, N_6)$ be M_2; the AC then generates $r_2 \in_R Z_q^*$ and computes $R_4 = r_2 \cdot P \bmod q$ and $S_2 = r_2^{-1} \cdot (M_2 - SK_{AC} \cdot (R_4)_x) \bmod q$. Waiting τ minutes, $\tau \in_R (0, t)$, the AC publishes (M_2, R_4, S_2) on BB_{AC} such that anyone can verify it by checking if equation $S_2 \cdot R_4 + (R_4)_x \cdot PK_{AC} \equiv M_2 \cdot P \bmod q$ holds.

(4) The AC then finds out the corresponding P_{A_j} of the auctioneer's id_{A_j} of this auction in the database. The AC computes the symmetrical key shared with A_j : $K_{AA} = SK_{AC} \cdot P_{A_j} \bmod q$. Then, the AC generates a session key shared between B_i and A_j as $K_{\text{session}} = H(ID_{\text{trans}}, price, date, (K_{AA})_x)$. The AC generates a receipt for B_i as $RECEIPT = (K_{AA})_x + H(SK_{AC} \parallel N_6) \cdot M_2$ and sends $E_{K_{AB}}(RECEIPT, K_{\text{session}}, id_{B_i}, N_7, ID_{\text{trans}})$ to B_i, where N_7 is a nonce.

(5) Upon receiving the message from the AC, B_i decrypts it with K_{AB} and checks the id_{B_i} and N_7 in it. If id_{B_i} is correct and the nonce is fresh, B_i stores $(RECEIPT, K_{\text{session}}, ID_{\text{trans}})$.

(6) After the *DeadLine* arrives, the AC closes the auction and determines the bidder with the highest price as the winner. The AC determines from the corresponding winning bid information on BB_{AC} whose *price* of the bid is equal to $Basic\_price$. Let $M_3 = (ID_{\text{trans}}, price, date, H(SK_{AC} \parallel N_6) \cdot P, dis, N_6)$; the AC chooses $r_3 \in_R Z_q^*$ and computes $R_5 = r_3 \cdot P \bmod q$ and $S_3 = r_3^{-1} \cdot (M_3 - SK_{AC} \cdot (R_5)_x) \bmod q$. Then, the AC publishes the winning information (M_3, R_5, S_3) on BB_{AC}. Anyone can verify the winner's information by checking $S_3 \cdot R_5 + (R_5)_x \cdot PK_{AC}? \equiv M_3 \cdot P (\bmod q)$.

5.5. Product Claiming Phase. After the AC publishes the auction result, every bidder can check whether he/she is a winner by checking whether his/her transaction identity ID_{trans} is equal to the winner's ID_{trans}. If a bidder B_i wins the auction, then he/she can claim the product from the auctioneer A_j using the following.

B_i computes $E_{K_{\text{session}}}(RECEIPT \cdot P \bmod q, N_8)$ and sends it to A_j, where N_8 is a nonce. Upon receiving the claiming request, A_j computes $K_{AA} = pw_{A_j} \cdot PK_{AC}$ firstly. Then, he/she browses the BB_{AC} and computes the session key $K_{\text{session}} = H(ID_{\text{trans}}, price, date, (K_{AA})_x)$ shared with the winner and extracts $(RECEIPT \cdot P \bmod q, N_8)$ by decrypting the ciphertext sent from B_i using K_{session}. Then, A_j checks the freshness of N_8. If N_8 is fresh, A_j verifies the validity of $RECEIPT$ by checking $RECEIPT \cdot P - (K_{AA})_x \cdot P? \equiv M_2 \cdot H(SK_{AC} \parallel N_6) \cdot P \bmod q$. Note that M_2 and $H(SK_{AC} \parallel N_6) \cdot P$ can be found in the winner's information on BB_{AC}. If the equation holds, A_j is convinced that B_i is the winner and sends the product to him/her.

5.6. Dispute Phase. In our proposal, the AC has the capability of tracing the whole transaction and identifying the participants in an auction if necessary. If a dispute has occurred, anyone who participates in an auction can submit a dispute request to the AC in this phase. In the following, we consider two situations to explain the dispute phase.

Situation 1 (auctioneer appeals to AC). If a nonpaying auction happened, which means that the winner B_i did not contact the auctioneer A_j after he/she won the auction, A_j can appeal to the AC and the AC will trace the transaction to deal with this dispute.

A_j browses BB_{AC} and finds the winner's ID_{trans} firstly. Then, A_j selects a nonce N_9 and computes $K_{AA} = pw_{A_j} \cdot PK_{AC}$, $e_3 = E_{K_{AA}}(ID_{\text{trans}}, id_{A_j}, N_9)$, $h_3 = H(e_3, K_{AA})$, $C_3 = (h_3 \cdot PK_{AC})_x + id_{A_j} \bmod q$, and $R_6 = h_3 \cdot P \bmod q$. Subsequently, A_j sends (C_3, R_6, e_3) to AC. Once receiving the dispute request from A_j, the AC retrieves $id_{A_j} = C_3 - (R_6 \cdot SK_{AC})_x \bmod q$ and checks the validation of id_{A_j} in the database. If it is valid, then the AC computes $K_{AA} = (H(SK_{AC}) \oplus v_{A_j}) \cdot SK_{AC} \bmod q$ and decrypts e_3 to extract $ID_{\text{trans}}, id_{A_j}$, and N_9. The AC checks the freshness of N_9. If the nonce is fresh, the AC checks if id_{A_j} in C_3 equals id_{A_j} in e_3. If they are not equal, the AC terminates operation immediately; otherwise, the AC searches the ID_{trans} on BB_{AC} and traces the transaction history to determine whether the problem really exists. If the AC ascertains that the problem claimed by A_j is true, the AC then finds out the corresponding dis and N_6 among the winner's information from BB_{AC}. Afterwards, the AC computes $k_{dis} = H(SK_{AC} \parallel N_6)$ and extracts (C_2, R_3) from dis. Finally, the AC can retrieve the dishonest winner's identity as $id_{B_i} = C_2 - (R_3 \cdot SK_{AC})_x \bmod q$ and publish it.

Situation 2 (bidder appeals to AC). If the auctioneer A_j of an auction refuses to send the product to the winner B_i, B_i also can appeal to the AC.

Firstly, B_i computes $K_{AB} = pw_{B_i} \cdot PK_{AC} \bmod q$, $e_4 = E_{K_{AB}}(RECEIPT \parallel ID_{\text{trans}} \parallel N_{10} \parallel id_{B_i})$, $h_4 = H(e_4, K_{AB})$, $C_4 = (h_4 \cdot PK_{AC})_x + id_{B_i} \bmod q$, and $R_7 = h_4 \cdot P \bmod q$, where N_{10} is a nonce. B_i then sends (C_4, R_7, e_4) to the AC. Upon receiving the dispute request from B_i, the AC retrieves $id_{B_i} = C_4 - (R_7 \cdot SK_{AC})_x \bmod q$ and checks if id_{B_i} exists in the database. If it exists, the AC then computes $K_{AB} = (H(SK_{AC}) \oplus v_{B_i}) \cdot SK_{AC} \bmod q$ and decrypts e_4 to extract $RECEIPT, ID_{\text{trans}}, N_{10}$, and id_{B_i}. The AC checks the freshness of N_{10}. If the nonce is fresh, the AC checks if id_{B_i} in C_4 equals id_{B_i} in e_4. If they are not equal, the AC terminates the operation immediately; otherwise, the AC then finds the corresponding N_6 and id_{A_j} of ID_{trans} on BB_{AC} and the corresponding v_{A_j} of id_{A_j} in the database. Then, the AC calculates $P_{A_j} = H(SK_{AC}) \oplus v_{A_j}$ and $K_{AA} = P_{A_j} \cdot SK_{AC} \bmod q$. The AC checks the validation of RECEIPT by verifying $RECEIPT ? = (K_{AA})_x + H(SK_{AC} \parallel N_6) \cdot M_2$. Note that M_2 and N_6 can be found in the winner's information on BB_{AC}. If RECEIPT is valid, the AC traces the transaction history to determine whether the problem really exists. If the AC

ascertains that the problem claimed by B_i is true, the AC publishes the identity of the dishonest auctioneer A_j.

6. BAN Logic Analysis

In this part, we will use BAN logic to analyze the authentication accuracy. Burrows-Abadi-Needham logic (also known as the BAN logic) is a set of rules for defining and analyzing information exchange protocols. Specifically, BAN logic helps its users to determine whether exchanged information is trustworthy, secured against eavesdropping, or both [30]. The notations of BAN logic are as follows:

$P \mid\equiv X$: P believes X.

$P \triangleleft X$: P sees X.

$P \mid\sim X$: P once said X.

$\mid \xrightarrow{PK} P$: P has PKas a public key.

$\#(X)$: the formula X is fresh.

$P \mid\Rightarrow X$: Phas jurisdiction over X.

$P \overset{K}{\longleftrightarrow} Q$: P and Q have a shared key K for communication.

$\{X\}_K$: the formula X encrypted under key K.

We mainly focus on the proof of on-shelf phase and bidding phase.

6.1. On-Shelf Phase. We idealized the on-shelf phase as below:

(I1) $A_j \rightarrow AC : \{id_{A_j}\}_{PK_{AC}}, \{\gamma, N_3, id_{A_j}\}_{K_{AA}}$.

(I2) $AC \rightarrow A_j : \{ID_{\text{on-shelf}}, N_4, id_{A_j}\}_{K_{AA}}$.

In this phase, we want to ensure that AC believes in the on-shelf information γ and A_j believes in the $ID_{\text{on-shelf}}$ transferred from AC, so we can conclude the two goals below:

(G1) $AC \mid\equiv \gamma$.

(G2) $A_j \mid\equiv ID_{\text{on-shelf}}$.

We need to assume that AC believes the nonce N_3 is fresh and A_j believes the nonce N_4 is fresh; AC believes that A_j does not send fake γ and id_{A_j}, so if A_j believes γ and id_{A_j} are true then AC will believe them too; K_{AA} is computed by AC self, so A_j believes that K_{AA} is shared between him/her and AC; and A_j believes that AC does not send fake $ID_{\text{on-shelf}}$, so if AC believes $ID_{\text{on-shelf}}$ is true then A_j will believe them too. All 6 assumptions are listed as below:

(A1) $AC \mid\equiv \#(N_3)$.

(A2) $AC \mid\equiv A_j \mid\Rightarrow \gamma$.

(A3) $AC \mid\equiv A_j \mid\Rightarrow id_{A_j}$.

(A4) $A_j \mid\equiv (AC \overset{K_{AA}}{\longleftrightarrow} A_j)$.

(A5) $A_j \mid\equiv \#(N_4)$.

(A6) $A_j \mid\equiv AC \Rightarrow ID_{\text{on-shelf}}$.

Proof. For goal (G1), our deduction is shown as the following formulas:

(F1) $AC \triangleleft \{id_{A_j}\}_{PK_{AC}}$.

(F2) $AC \triangleleft id_{A_j}$ (message decryption rule).

(F3) $AC \mid\equiv id_{A_j}$(our hypothesis, we will prove that it is true later).

(F4) $AC \mid\equiv (AC \overset{K_{AA}=SK_{AC}\cdot P_{A_j}}{\longleftrightarrow} A_j)$ ((F3), the computing property of believing in operator).

(F5) $AC \triangleleft \{\gamma, N_3, id_{A_j}\}_{K_{AA}}$.

(F6) $AC \mid\equiv A_j \mid\sim \{\gamma, N_3, id_{A_j}\}$ ((F4), (F5), message meaning rule).

(F7) $AC \mid\equiv \#(\{\gamma, N_3, id_{A_j}\})$ ((A1), freshness propagation rule).

(F8) $AC \mid\equiv A_j \mid\equiv \{\gamma, N_3, id_{A_j}\}$ ((F6), (F7), nonce verification rule).

(F9) $AC \mid\equiv A_j \mid\equiv id_{A_j}$ ((F8), believing rule).

(F10) $AC \mid\equiv id_{A_j}$ ((F9), (A3), jurisdiction rule, our hypothesis in (F3) is proved).

(F11) $AC \mid\equiv A_j \mid\equiv \gamma$ ((F8), believing rule).

(F12) $AC \mid\equiv \gamma$ ((F11), (A3), jurisdiction rule).

According to formula (F12), the proof of (G1) is completed.

For goal (G2), our deduction is shown as following formulas:

(F13) $A_j \triangleleft \{ID_{\text{on-shelf}}, N_4, id_{A_j}\}_{K_{AA}}$.

(F14) $A_j \mid\equiv AC \mid\sim \{ID_{\text{on-shelf}}, N_4, id_{A_j}\}$ ((F13), (A4), message meaning rule).

(F15) $A_j \mid\equiv \#(\{ID_{\text{on-shelf}}, N_4, id_{A_j}\})$ ((A5), freshness propagation rule).

(F16) $A_j \mid\equiv AC \mid\equiv \{ID_{\text{on-shelf}}, N_4, id_{A_j}\}$ ((F14), (F15), nonce verification rule).

(F17) $A_j \mid\equiv AC \mid\equiv ID_{\text{on-shelf}}$ ((F16), believing rule).

(F18) $A_j \mid\equiv ID_{\text{on-shelf}}$ ((F16), (A6), jurisdiction rule).

According to formula (F18), the proof of (G2) is completed. □

6.2. Bidding Phase. Our bidding phase can be idealized as below:

(I3) $B_i \rightarrow AC : \{id_{B_i}\}_{PK_{AC}}, \{\eta, N_5, id_{B_i}\}_{K_{BA}}$.

(I4) $AC \rightarrow B_i : \{RECEIPT, K_{\text{session}}, N_7, id_{B_i}\}_{K_{AB}}$.

In bidding phase, we want to ensure that AC can be convinced that bidding information η transferred from B_i is true; additionally, B_i should be convinced that $RECEIPT$ and K_{session} transferred from AC are true. So the three goals below can be concluded:

(G3) $AC \mid\equiv \eta$.

(G4) $B_i \mid\equiv RECEIPT$.

(G5) $B_i \mid\equiv K_{\text{session}}$.

We need to assume that AC believes N_5 is fresh and B_i believes N_7 is fresh; if B_i believes η and id_{B_i} are true, AC believes in them too because they are generated by B_i; similarly, if AC believes in $RECEIPT$ and K_{session} then B_i believes in them too; because K_{AB} is generated by B_i, B_i believes that he/she shared K_{AB} with AC. All assumptions are listed as below:

(A7) $AC \mid\equiv \#(N_5)$.

(A8) $AC \mid\equiv B_i \mid\Rightarrow \eta$.

(A9) $AC \mid\equiv B_i \mid\Rightarrow id_{B_i}$.

(A10) $B_i \mid\equiv (AC \overset{K_{AB}}{\longleftrightarrow} B_i)$.

(A11) $B_i \mid\equiv \#(N_7)$.

(A12) $B_i \mid\equiv AC \Rightarrow RECEIPT$.

(A13) $B_i \mid\equiv AC \Rightarrow K_{\text{session}}$.

Proof. For goal (G3), our deduction is shown as the following formulas:

(F19) $AC \triangleleft \{id_{B_i}\}_{PK_{AC}}$.

(F20) $AC \triangleleft id_{B_i}$ (message decryption rule).

(F21) $AC \mid\equiv id_{B_i}$ (our hypothesis, we will prove that it is true later).

(F22) $AC \mid\equiv (AC \overset{K_{AB}=SK_{AC}\cdot P_{B_i}}{\longleftrightarrow} B_i)$ ((F21), operator believes in property).

(F23) $AC \triangleleft \{\eta, N_5, id_{B_i}\}_{K_{BA}}$.

(F24) $AC \mid\equiv B_i \mid\sim \{\eta, N_5, id_{B_i}\}$ ((F22), (F23), message meaning rule).

(F25) $AC \mid\equiv \#(\{\eta, N_5, id_{B_i}\})$ ((A7), freshness propagation rule).

(F26) $AC \mid\equiv B_i \mid\equiv \{\eta, N_5, id_{B_i}\}$ ((F24), (F25), nonce verification rule).

(F27) $AC \mid\equiv B_i \mid\equiv id_{B_i}$ ((F26), believing rule).

(F28) $AC \mid\equiv id_{B_i}$ ((F27), (A9), jurisdiction rule, our hypothesis in (F21) is proved).

(F29) $AC \mid\equiv B_i \mid\equiv \eta$ ((F26), believing rule).

(F30) $AC \mid\equiv \eta$ ((F29), (A8), jurisdiction rule).

According to formula (F30), the proof of (G3) is completed.

For goal (G4), our deduction is shown as the following formulas.

(F31) $B_i \triangleleft \{RECEIPT, K_{\text{session}}, N_7, id_{B_i}\}_{K_{AB}}$.

(F32) $B_i \mid\equiv AC \mid\sim \{RECEIPT, K_{\text{session}}, N_7, id_{B_i}\}$ ((F31), (A10), message meaning rule).

(F33) $B_i \mid\equiv \#(\{RECEIPT, K_{\text{session}}, N_7, id_{B_i}\})$ ((A11), freshness propagation rule).

(F34) $B_i \mid\equiv AC \mid\equiv \{RECEIPT, K_{\text{session}}, N_7, id_{B_i}\}$ ((F32), (F33), nonce verification rule).

(F35) $B_i \mid\equiv AC \mid\equiv RECEIPT$ ((F34), believing rule).

(F36) $B_i \mid\equiv RECEIPT$ ((F34), (A12), jurisdiction rule).

According to formula (F36), the proof of (G4) is completed. Finally, the goal of (G5) is deduced by the following formulas:

(F37) $B_i \mid\equiv AC \mid\equiv K_{\text{session}}$ ((F34), believing rule).

(F38) $B_i \mid\equiv RECEIPT$ ((F37), (A13), jurisdiction rule).

The proof of (G5) is completed. □

According to our proofs, we can see that our protocol completes the mutual authentication between users and AC. In on-shelf phase, AC can be sure that on-shelf information is sent from legal user and auctioneer can be sure that $ID_{\text{on-shelf}}$ is transmitted from AC; in bidding phase, AC can be sure that bidding information is sent from legal user and bidder can be sure that receipt and session key are generated and sent by AC. As the results, our protocol achieves authentication accuracy.

7. Security Analysis

In this part, we analyze the security of our proposal. We assume that an adversary can eavesdrop on public communications on the Internet and read the information on BB_{AC}; however, he/she cannot read ciphertexts without getting the encrypting keys. Of course, the adversary cannot calculate the keys from ciphertexts. All roles in the system except for the AC can conspire to do something illegally for their benefit. In the following, we provide different scenarios to explain our security defense.

(1) Impersonation Attack. It is impossible for an adversary to impersonate a user. When a user wants to communicate with the AC, he/she should send his/her identity ciphertext (C_n, R_n, e_n) to the AC. The AC retrieves and checks the user's identity from this message. We can see that the adversary cannot generate a legal (C_n, R_n, e_n) to pass this verification without the knowledge of the AC's private key SK_{AC} and the key K_{AU} shared between the AC and the user. Even if the adversary eavesdrops on the other users' communications and stores their (C_n, R_n, e_n), he/she still cannot use these to do anything because the AC will check if the identity included in C_n equals that in e_n. The only method for the adversary is to replay the whole (C_n, R_n, e_n); however, this will not work because the AC can check the freshness of the nonce in e_n.

(2) Collusion Attack. In some cases, bidders and auctioneers may conspire to break the secret of the AC. For example, a bidder may send $RECEIPT$ instead of $RECEIPT \cdot P$ to an auctioneer. However, with this knowledge, they can only compute $H(SK_{AC} \parallel N_6) = (RECEIPT - (K_{AA})_x)/M_2$, where K_{AA} can be computed by the auctioneer and M_2 can be found in BB_{AC}. Even obtaining $H(SK_{AC} \parallel N_6)$, they cannot compute SK_{AC} because $H(\cdot)$ is a one-way hash function.

(3) Forging Receipt. Firstly, we consider the case that a malicious bidder wants to forge a receipt to impersonate the winner of an auction by himself/herself. Because $RECEIPT =$

$(K_{AA})_x + H(SK_{AC} \parallel N_6) \cdot M_2$, the malicious bidder obviously cannot forge a valid receipt without knowing the *AC*'s private key SK_{AC} and the key K_{AA} shared between the *AC* and the auctioneer. Secondly, it is also impossible for the winner and auctioneer to conspire and create the receipt of another auction. As mentioned in the Collusion Attack, they only can compute $H(SK_{AC} \parallel N_6)$. However, for different bidders, N_6's are different, so $H(SK_{AC} \parallel N_6)$ are different in different auctions. Obviously, they cannot create a useful receipt of the other auction with $H(SK_{AC} \parallel N_6)$ by collusion. Besides, it is meaningless for the winner and auctioneer of an auction to forge a receipt of the auction.

(4) Denial of Service Attack. If an adversary wants to mount a deny of service attack on a bidder (B_i) such that he/she cannot claim the product in the product claiming phase when he/she wins an auction, the adversary should generate a fake receipt and send it to the bidder during the bidding phase. However, the receipt transmitted to B_i needs to be encrypted with B_i's identity by the symmetrical key shared between B_i and the *AC* as $E_{K_{AB}}(RECEIPT, K_{\text{session}}, id_{B_i}, N_7, ID_{\text{trans}})$. After receiving the ciphertext, the bidder should decrypt it and check if his identity id_{B_i} is correct. So it is impossible for an adversary to deceive a bidder to accept a fake receipt.

(5) Publishing Fake Information. In our proposal, all published information, including new auction information in the on-shelf phase and bidding information and winner information in the bidding phase are signed by the private key of *AC*, so anyone can verify the published information with *AC*'s public key. Let us consider a situation: if a bidder wants to win an auction, he/she may try to publish a fake bidding message with a very high bidding price on BB_{AC} so that no one else wants to compete with him/her. However, he/she cannot be successful without knowing the private key of the *AC*, because every user can check the signature of the *AC* on this message.

(6) Privacy Preserving. In our scheme, an adversary cannot link any information together to analyze a user's privacy. Only auctioneers' identities need to be published in BB_{AC} for enhancing users' willingness to bid. All packets transmitted in the public channel are encrypted, so the adversary cannot know whether these packets can be linked to certain information published on BB_{AC} even if they captured these packets. It is noteworthy that, in our schemes, we use a delay operation in bidding and on-shelf phases for increasing difficulty of linkage of different messages. For example, if the adversary detects that Alice sends an on-shelf message to the *AC* and the *AC* publishes an auction advertisement on the website including the auctioneer's identity immediately, the adversary would ascertain that Alice is the auctioneer and obtain Alice's identity. If we delay publishing the auction message several minutes later (i.e., the *AC* publishes it with other auction information together), then the adversary cannot distinguish a specific user from them. The larger the τ we use, the stronger the privacy preserving function is. However, we should balance the privacy preserving and user experience. Moreover, in the bidding phase, a bidder's identity can be retrieved by $id_{B_i} = C_2 - (R_3 \cdot SK_{AC})_x \bmod q$, where C_2 and R_3 are involved in the information published on BB_{AC}. However, these parameters are protected by *AC*'s private key as $dis = E_{k_{dis}}(C_2, R_3)$ before being published on BB_{AC}, where $k_{dis} = H(SK_{AC} \parallel N_6)$, such that no one can extract id_{B_i} except for the *AC*. In the product claiming phase, an auctioneer cannot know the winner's identity from $RECEIPT \cdot P$ but can only know the validity of the receipt. As indicated in the results, our scheme indeed achieves privacy preserving.

(7) Accessing to the Database of the AC. In the definition of a network model (Section 4), we have mentioned that we do not consider the database of the *AC* to be completely secure in our proposal. An adversary may access the messages stored in it by certain methods. However, accessing the database would not destroy the security of the whole system. Furthermore, the *AC*'s private key should be kept in a trusted place, such as a bank. There are mainly four data sets maintained in the *AC*: the first is (id_U, v_U) generated in the registration phase, the second is $(ID_{\text{product}}, Basic\_price, Deadline, ID_{\text{on-shelf}}, id_{A_j}, R_2, S_1)$ generated in the on-shelf phase, and the final are $(ID_{\text{trans}}, date, price, dis, N_6, R_4, S_2)$ and $(ID_{\text{trans}}, price, date, H(SK_{AC} \parallel N_6) \cdot P, dis, N_6)$ generated in the bidding phase. Only the first data set is unpublished, so we only consider the safety of the first data set here. If a hacker reads the content of the first data set, he/she only can obtain all users' identities and their partial keys (P_Us) which have been protected by *AC*'s private key as the form $v_U = P_U \oplus H(SK_{AC})$. With only protected partial keys, the hacker cannot compute the symmetrical key K_{AU} ($K_{AU} = pw_u \cdot PK_{AC} \bmod q = SK_{AC} \cdot P_U \bmod q$) shared between users and the *AC*, because the private key of the *AC* is kept secret. Furthermore, he/she cannot retrieve a user's password from the partial key based on ECDLP. So we can see that accessing the database does not destroy the safety of the whole system.

(8) Password Guessing Attack. In our scheme, users' passwords are needed to protect system security. However, because the user's password is not very long and there is a certain routine mode, attackers may try to mount the password guessing attack to the password-based communication schemes. To prevent this attack, in our scheme, users' passwords will not be transmitted or stored in plaintext. As mentioned in the registration phase (Section 5.2), the password related information P_U is transformed to ciphertext $C = (r_0 \cdot PK_{AC} + P_U) \bmod q$ before transferring and is stored as the format of $v_U = P_U \oplus H(SK_{AC})$ in *AC*'s database. Obviously, even an attacker collects ciphertext or v_U by eavesdropping the whole communication channel or intruding *AC*'s database; he/she can not obtain any user's password by password guessing attack without the knowledge of *AC*'s secret key SK_{AC}.

Table 1 ("○" means fully achieved, "△" means partially achieved, and "×" means not achieved) shows the security comparison among our proposal and three other schemes: Chang et al.'s [14], Xiong et al.'s [13], and Chung et al.'s [18]. As shown in Table 1, our scheme fulfills the most secure properties of an electronic auction system. In the analysis of "*privacy preserving,*" our scheme can achieve the

Table 1: Comparison in security properties of the three schemes.

Property	Proposal			
	Our proposal	Chang et al.'s proposal [14]	Xiong et al.'s proposal [13]	Chung et al.'s proposal [18]
Anonymity	○	○	○	○
Easy revocation	○	○	△	△
Fairness	○	○	○	○
Nonrepudiation	○	○	○	○
Onetime registration	○	○	○	○
Traceability	○	○	○	○
Unforgeability	○	○	○	○
Unlinkability	○	×	○	○
Verifiability	○	△	×	○
On-shelf mechanism	○	○	×	×
Without secure channel	○	○	×	×
Password guessing attack defending	○	○	○	○

Table 2: Comparison in computational cost of the three schemes.

Property	Proposal			
	Our proposal	Chang et al.'s proposal [14]	Xiong et al.'s proposal [13]	Chung et al.'s proposal [18]
Registration phase	8PM + H	9PM	4PM + 2H	5PM + 3H
On-shelf phase	7PM + 2E + 2D + 2H	10PM + 2E + 2D + 2H	×	×
Bidding phase	$(10n + 2)$PM + $3n$E + $2n$D + $5n$H	$12n$PM + $2n$E + $2n$D	$(4n + 1)$PM + 3BM + 2H	$(8n + 2)$PM + $(3n + 1)$H
Product claiming phase	2PM + E + D + 1H	6PM	×	×
Dispute phase (bidder/auctioneer)	6PM + E + 1D + 3H/6PM + E + 2D + 3H	3PM + 2H/3PM + E + D	$0.5n$BM/×	×

properties of anonymity and unlinkability; in the analysis of "*Publishing Fake Information,*" our scheme can achieve the properties of fairness and verifiability; and in the analyses of "*Impersonation Attack*, *Forging Receipt*, and *Denial of Service Attack,*" our scheme can achieve the property of unforgeability. Similar to [13, 14, 18], the bidding information should be published on an open website, so the bidder can check if his/her bid appeared on it to ensure fairness. Based on the digital signature, bidders cannot deny their bids so nonrepudiation is achieved in four proposals. By providing the dispute mechanism, our scheme achieves the properties of easy revocation, nonrepudiation, and traceability. Furthermore, our scheme overcomes the security problem (leakage of unlinkability) and meanwhile inherits the advantage of without a secure channel in Chang et al.'s scheme [14]. It is noteworthy that, compared to [14], our protocol needs to store a little of the verification table in the server side. However, as we analyzed before, leakage of these verification tables will not influence system security, so we say that our protocol achieves partial property of no verification table.

With regard to verifiability, in the bidding phase of Chang et al.'s proposal [14], only the winner's bidding information will be signed by the *AC* and can be publicly verified, which may lead attackers to modify any new bidding information with a higher price such that no one else wants to bid for it further. So verifiability is not fully achieved in [14]. As [14] analyzed, the designed revocation function in [13, 18] only partially achieved easy revocation. Our protocol and [14] have a secure on-shelf phase compared to [13, 18]. Our protocol does not need a secure channel to achieve verification in any phase similar to [14]. However, in [13, 18], a secure channel is needed to assist in the completion of the whole protocol.

8. Performance Analysis

In this section, we compare the cost of our scheme with the other three schemes [13, 14, 18] with regard to computational cost, traffic, and communication rounds.

Table 2 shows the comparison of computational cost, where "PM" means a point multiplication operation of ECC, "BM" means a bilinear mapping which can be implemented on ECC such as Weil pairing, "E" means a symmetrical encryption, "D" means a symmetrical decryption, "H" means a hash function, and "n" means the number of bidders in an auction.

As we know, a bilinear map is an expensive operation; the more it is used, the more computation cost is needed. Compared with point multiplication, the hash function only consumes negligible computation cost. Symmetrical cryptographic operation is also cheaper than point multiplication. From the comparison in Table 2, we can see that our proposal needs less computation cost compared with Chang et al.'s proposal [14] in each phase, except for the dispute phase.

Table 3: Comparison in traffic of the three schemes (B: bytes).

Property	Proposal			
	Our proposal	Chang et al.'s proposal [14]	Xiong et al.'s proposal [13]	Chung et al.'s proposal [18]
Registration phase	82 B	192 B	108 B	128 B
On-shelf phase	144 B	266 B	×	×
Bidding phase	$176n$ B	$290n$ B	$(96n + 120)$ B	$(119n + 248)$ B
Product claiming phase	48 B	149 B	×	×
Dispute phase (auctioneer/bidder)	112 B/128 B	37 B/85 B	48 B/×	×

Table 4: Comparison in communication rounds of the three schemes.

Property	Proposal			
	Our proposal	Chang et al.'s proposal [14]	Xiong et al.'s proposal [13]	Chung et al.'s proposal [18]
Registration phase	2	2	1	1
On-shelf phase	3	3	×	×
Bidding phase	$2n$	$2n$	1	$n + 2$
Product claiming phase	1	1	×	×
Dispute phase (auctioneer/bidder)	1/1	1/1	1	×

However, in reality, the first four phases are more frequently executed.

To make the traffic comparison, we determine the elliptic curve operation as NistP 192 [24], hash function as SHA-256 [31], and symmetrical encryption/decryption as AES-256 [32]. The length of the user's identity is 4 bytes, product identity is 5 bytes, transaction identify is 5 bytes, *GID* in [14] is 5 bytes, *price* is 5 bytes, and *Deadline* is 4 bytes. Note that we ignore the length of nonce transmitted in communications. As shown in Table 3, we can see that our protocol has a traffic advantage compared with related works [13, 14, 18], except for dispute phase of Chang et al.'s scheme [14]. The main reason for this is that our scheme finished the key establishment in the registration phase for saving the traffic cost of public keys and certificates. Comparing with Chang et al.'s proposal, though our dispute phase needs more traffic, this phase is not often used compared with the other four phases. As a result, we believe that our proposal has traffic advantages still. Finally, we compare the communication rounds of our scheme and the related schemes in Table 4. As shown in this table, our protocol needs the same communication rounds as those in Chang et al.'s protocol [14].

9. Conclusions

In this paper, we pointed out that Chang et al.'s protocol does not fulfill the unlinkability such that users' privacy will be leaked by linking different messages together. Considering the importance of privacy preservation, we proposed a novel electronic English auction system. By using symmetrical encryptions and fewer ECC operations, our protocol can not only improve the security but also reduce the system cost. In the registration phase, we connect a user's identity and the corresponding symmetrical key so that the *AC* can compute the shared symmetrical key easily with the user's identity. Furthermore, each user's identity is encrypted with the *AC*'s public key so it cannot be revealed without the knowledge of the *AC*'s private key. In the bidding phase, we improve the efficiency of generating bidding receipts. In our proposal, we fulfill the verifiability of all messages published on the *AC*'s website; however, Chang et al.'s proposal only achieved this feature partially. We used BAN logic to prove that our protocol indeed realizes mutual authentication. The security analysis and performance analysis show that our protocol fulfills more security properties and is more efficient for implementation compared with recent works.

Competing Interests

The authors declare that they have no competing interests.

References

[1] Statista.com, *Number of Worldwide Internet Users from 2000 to 2014 (in millions)*, Statista, New York, NY, USA, 2015, http://www.statista.com/statistics/273018/number-of-internet-users-worldwide/.

[2] S. Parsons, J. A. Rodriguez-Aguilar, and M. Klein, "Auctions and bidding: a guide for computer scientists," *ACM Computing Surveys*, vol. 43, no. 2, article 10, 2011.

[3] C. Cachin, "Efficient private bidding and auctions with an oblivious third party," in *Proceedings of the 6th ACM Conference on Computer and Communication Security (CCS '99)*, pp. 120–127, Singapore, November 1999.

[4] K. Q. Nguyen and J. Traoré, "An online public auction protocol protecting bidder privacy," in *Proceedings of the 5th Australasian Conference on Information Security and Privacy (ACISP '00), Brisbane, Australia, July 2000*, vol. 1841 of *Lecture Notes in Computer Science*, pp. 427–442, Springer, 2000.

[5] K. Omote and A. Miyaji, "A practical English auction with one-time registration," in *Information Security and Privacy: 6th*

Australasian Conference, ACISP 2001 Sydney, Australia, July 11–13, 2001 Proceedings, vol. 2119 of *Lecture Notes in Computer Science*, pp. 221–234, Springer, Berlin, Germany, 2001.

[6] C.-C. Chang and Y.-F. Chang, "Efficient anonymous auction protocols with freewheeling bids," *Computers & Security*, vol. 22, no. 8, pp. 728–734, 2003.

[7] R. Jiang, L. Pan, and J.-H. Li, "An improvement on efficient anonymous auction protocols," *Computers & Security*, vol. 24, no. 2, pp. 169–174, 2005.

[8] Y.-F. Chang and C.-C. Chang, "Enhanced anonymous auction protocols with freewheeling bids," in *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*, vol. 1, pp. 353–358, April 2006.

[9] K. Suzuki and M. Yokoo, "Secure multi-attribute procurement auction," in *Information Security Applications: 6th International Workshop, WISA 2005, Jeju Island, Korea, August 22–24, 2005, Revised Selected Papers*, vol. 3786 of *Lecture Notes in Computer Science*, pp. 306–317, Springer, Berlin, Germany, 2005.

[10] D.-H. Shih, C.-H. Cheng, and J.-C. Shen, "A secure protocol of reverse discriminatory auction with bid privacy," in *Proceedings of the 6th International Conference on the Management of Mobile Business (ICMB '07)*, 52 pages, Toronto, Canada, July 2007.

[11] D. C. Parkes, M. O. Rabin, S. M. Shieber, and C. Thorpe, "Practical secrecy-preserving, verifiably correct and trustworthy auctions," *Electronic Commerce Research and Applications*, vol. 7, no. 3, pp. 294–312, 2008.

[12] H. Xiong, Z. Qin, and F. Li, "An anonymous sealed-bid electronic auction based on ring signature," *International Journal of Network Security*, vol. 8, no. 3, pp. 235–242, 2009.

[13] H. Xiong, Z. Chen, and F. Li, "Bidder-anonymous English auction protocol based on revocable ring signature," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7062–7066, 2012.

[14] C. C. Chang, T. F. Cheng, and W. Y. Chen, "A novel electronic english auction system with a secure on-shelf mechanism," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 4, pp. 657–668, 2013.

[15] K. Peng, C. Boyd, E. Dawson, and K. Viswanathan, "Robust, privacy protecting and publicly verifiable sealed-bid auction," in *Information and Communications Security: 4th International Conference, ICICS 2002 Singapore, December 9–12, 2002 Proceedings*, vol. 2513 of *Lecture Notes in Computer Science*, pp. 147–159, Springer, Berlin, Germany, 2002.

[16] K. Sako, "An auction protocol which hides bids of losers," in *Public Key Cryptography: Third International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2000, Melbourne, Victoria, Australia, January 18–20, 2000. Proceedings*, vol. 1751 of *Lecture Notes in Computer Science*, pp. 422–432, Springer, Berlin, Germany, 2000.

[17] H. Kikuchi, M. Hakavy, and D. Tygar, "Multi-round anonymous auction protocols," *IEICE Transactions on Information and Systems*, vol. 82, no. 4, pp. 769–777, 1999.

[18] Y.-F. Chung, Y.-T. Chen, T.-L. Chen, and T.-S. Chen, "An agent-based English auction protocol using Elliptic Curve Cryptosystem for mobile commerce," *Expert Systems with Applications*, vol. 38, no. 8, pp. 9900–9907, 2011.

[19] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology—CRYPTO '85 Proceedings*, vol. 218 of *Lecture Notes in Computer Science*, pp. 417–426, Springer, Berlin, Germany, 1985.

[20] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[21] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," in *Cryptographic Hardware and Embedded Systems—CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11–13, 2004. Proceedings*, vol. 3156 of *Lecture Notes in Computer Science*, pp. 119–132, Springer, Berlin, Germany, 2004.

[22] J. Hoffstein, J. Pipher, and J. H. Silverman, *An Introduction to Mathematical Cryptography*, Springer, New York, NY, USA, 2008.

[23] J. W. Bos, A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, "Elliptic curve cryptography in practice," in *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3–7, 2014, Revised Selected Papers*, vol. 8437 of *Lecture Notes in Computer Science*, pp. 157–175, Springer, Berlin, Germany, 2014.

[24] Z. Tan, Z. Liu, and C. Tang, "Digital proxy blind signature schemes based on DLP and ECDLP," in *MM Research Preprints*, no. 21, pp. 212–217, MMRC, AMSS, Chinese Academy of Sciences, Beijing, China, 2002.

[25] H.-T. Liaw, W.-S. Juang, and C.-K. Lin, "An electronic online bidding auction protocol with both security and efficiency," *Applied Mathematics and Computation*, vol. 174, no. 2, pp. 1487–1497, 2006.

[26] C.-C. Chang and T.-F. Cheng, "An efficient proxy raffle protocol with anonymity-preserving," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 772–778, 2009.

[27] E. Pinker, A. Seidmann, and Y. Vakrat, "Using bid data for the management of sequential, multi-unit, online auctions with uniformly distributed bidder valuations," *European Journal of Operational Research*, vol. 202, no. 2, pp. 574–583, 2010.

[28] J.-S. Lee and K.-S. Lin, "An innovative electronic group-buying system for mobile commerce," *Electronic Commerce Research and Applications*, vol. 12, no. 1, pp. 1–13, 2013.

[29] M.-J. Li, J. S. T. Juan, and J. H. C. Tsai, "Practical electronic auction scheme with strong anonymity and bidding privacy," *Information Sciences*, vol. 181, no. 12, pp. 2576–2586, 2011.

[30] M. Burrows, M. Abadi, and R. Needham, "Authentication: a practical study in belief and action," in *Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning about Knowledge*, pp. 325–342, March 1988.

[31] H. Gilbert and H. Handschuh, "Security analysis of SHA-256 and sisters," in *Proceedings of the 10th Annual International Workshop on Selected Areas in Cryptography (SAC '03), Ottawa, Canada, August 2003*, vol. 3006 of *Lecture Notes in Computer Science*, pp. 175–193, Springer, 2003.

[32] F. P. Miller, A. F. Vandome, and J. McBrewster, *Advanced Encryption Standard*, Alphascript, Beau Bassin, Mauritius, 2009.

Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2016, Article ID 2750268, 7 pages
http://dx.doi.org/10.1155/2016/2750268

Research Article

Analysis and Improvement of Key Distribution Scheme for Secure Group Communication

Jia Ning Luo[1] and Ming Hour Yang[2]

[1]*Department of Information and Telecommunication Engineering, Ming Chuan University, Taoyuan 333, Taiwan*
[2]*Department of Information and Computer Engineering, Chung Yuan Christian University, Taoyuan 320, Taiwan*

Correspondence should be addressed to Jia Ning Luo; deer@mail.mcu.edu.tw

Received 16 November 2015; Accepted 2 March 2016

Academic Editor: Isao Echizen

In a secure group communication, messages between a group coordinator and members are protected by session keys. If a group's membership changes, the session keys should be updated to insure forward secrecy and backward secrecy. Zhou and Huang proposed a key-updated scheme based on ciphertext-policy attribute encryption algorithm to improve the security of key-update mechanism, but their scheme is vulnerable: a malicious group member may send forged key-update messages to control the group. In this paper, we analyze the vulnerability in Zhou and Huang's scheme and propose an enhanced scheme. In our scheme, only the group initiator can update group keys and the verification of key-update mechanism is improved to prevent malicious insiders from controlling the group. We also give a security and performance analysis of our scheme.

1. Introduction

In recent years, social networks such as Google+, Facebook, and Twitter are receiving wide popularity and provide much convenience in people's daily life. In a social network, people can maintain their own social circles freely, such as adding or removing their friends and sharing messages to specific members within a group.

Social networks are based on cloud computing technology. In a cloud service, users' data and documents are not stored in their computers, but on cloud servers. In most cases, users' data is not encrypted and, therefore, is vulnerable to system vulnerabilities, unauthorized access, and privacy leak under government's censorship [1–3].

The secure multicast technique can be used to enhance users' privacy in social networks [4]. For secure multicast, the confidentiality of group communication is secured because group members share a session key to de/encrypt their communications. Once a group member leaves a group or a new member joins a group, their group key must be renewed. A group coordinator has to rekey the group. He needs to ensure that a new group key is delivered to every new member through a secure channel and that every old member's key is updated simultaneously. By this, new members are not able to access previous messages, and those who have left a group cannot access the group's new messages. This guarantees the forward and backward secrecy in group communication and has made key updating an important issue for secure multicast.

Chang et al. [5] propose Flat Table (FT) for key management in a binary tree. A group coordinator only needs to store $\log N$ keys and each member can join $N - 1$ subgroups. To have higher flexibility of data encryption, Attribute-Based Encryption (ABE) [6–9] and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [10] are proposed. Bethencourt et al. [11] propose CP-ABE for multicast key management. It features complex access control on encrypted data. Zhou and Huang [12] combine FT and CP-ABE to present Efficient Group Keying (EGK). EGK is more efficient for a group coordinator to rekey his group, and the size of its ciphertext is fixed. No matter how its membership changes, the size of its ciphertext remains the same. EGK is also able to keep its storage within $O(\log N)$, where N denotes the maximum members. Huang et al. also come up with secure virtual trust routing and multicasting, aiming to apply EGK to networking

routing [13]. Jia et al. propose a layered EGK architecture [14]. Huang et al. proposed a cloud solution of EGK [15]. As each mobile device is seen as a service node, they propose to process mobile cloud data through trust management and private data isolation.

However, we have discovered certain security issues in the EGK model. For example, a malicious group member may launch the following attacks within EGK. (1) He may perform Man-in-the-Middle (MITM) attacks to intercept group communications and modify the group coordinator's messages. (2) He may masquerade as the coordinator and forge the key-update message and become the new coordinator. In Jia et al.'s EGK-based scheme [14], we have also found the threat of desynchronization because the group members cannot verify whether the messages are sent from the coordinator.

In this paper, we analyze the security issues in EGK and propose a new secure key management scheme. Our scheme resists the MITM attack and masquerade attack. The rest of this paper is organized as follows. In Section 2, we analyze the vulnerability of EGK. We describe our scheme in Section 3 and give a performance analysis in Section 4. A conclusion is drawn in Section 5.

2. EGK Scheme

Zhou and Huang [12] propose an Efficient Group Keying (EGK) method that is based on Ciphertext-Policy Attribute-Based Encryption (CP-ABE). Following CP-ABE's policy, EGK takes each bit of group members' ID as an attribute. The group key is calculated by using the combination of specific group members.

When a new member joins a specific group, the coordinator generates a new group key and sends it to the new member by a secure channel and uses the old group key to encrypt the new one and sends it to the members. When a member leaves the group, the coordinator uses the Quine-McCluskey algorithm [16] to update the new group key.

The notations used in EGK are listed in Notations.

2.1. Initial Stage. At this stage, a group coordinator creates a group and the maximum membership of the group is N. The length of an ID is designed as $n = \log N$. The total number of L_B's is $2n$, and the total number of P_B's is $2n$ too. Consider $L_B = \{B_0, \overline{B_0}, B_1, \overline{B_1}, \ldots, B_{n-1}, \overline{B_{n-1}}\}$; B_i denotes that the ith bit of ID is 1; $\overline{B_i}$ denotes that the ith bit of ID is 0. Consider $P_B = \{p_{B_0}, \overline{p_{B_0}}, p_{B_1}, \overline{p_{B_1}}, \ldots, p_{B_{n-1}}, \overline{p_{B_{n-1}}}\}$; consider $p_{B_i}, \overline{p_{B_i}} \in Z_q$; P_B maps L_B, p_{B_0} maps B_0, $\overline{p_{B_0}}$ maps B_0, and so on. Then, the coordinator generates a master key $\mathrm{MK} = \{\beta, g^\alpha, g^\beta, e(g,g)^\alpha, L_B, P_B\}$ and sets the parameters of a bilinear map function, such as q, G_0, G_1, g, and e. At last, he generates a hash function H and defines the public parameter $\mathrm{Params} = \{G_0, e, g, H\}$.

2.2. Assign Private Key to Group Member. When a new member joins a group, the group coordinator gives to the member a unique ID and associates the member with an attribute A_{ID}. For example, there is a group whose maximum membership is 64, and the length of the ID is $\log_2 64$. The coordinator assigns $\mathrm{ID} = 001011$ to the new member and accordingly associates the member with the attribute $A_{\mathrm{ID}} = \{\overline{B_5}, \overline{B_4}, B_3, \overline{B_2}, B_1, B_0\}$. Then, he runs the algorithm $\mathrm{KeyGen}(\mathrm{MK}, A_{\mathrm{ID}})$ to generate a private key for the new member:

(1) He generates a random value $\gamma \in Z_q^*$.

(2) He calculates $D = g^{(\alpha+\gamma)/\beta}$.

(3) According to A_{ID}, he calculates $\forall B_i \in A_{\mathrm{ID}} : D_B = g^{\gamma B_i}$.

(4) He performs the mapping of L_B and P_B, $\forall B_i \in A_{\mathrm{ID}} : D_B = g^{\gamma p_{B_i}}$.

(5) He generates a private key $\mathrm{SK}_{\mathrm{ID}} = \{D = g^{(\alpha+\gamma)/\beta}, \forall B_i \in A_{\mathrm{ID}} : D_B = g^{\gamma p_{B_i}}\}$ for the new member.

(6) At last, the coordinator sends ID, A_{ID}, $\mathrm{SK}_{\mathrm{ID}}$, and current session key SEK to the new member through a secure channel.

2.3. Encryption/Decryption Algorithms. When a group coordinator is going to multicast a message, he can use a different bit-assignment to assign specific members to decrypt the message. Then, he has to run the Quine-McCluskey algorithm to minimize all group members' IDs. The IDs are compared in pairs and reduced until there is no pair with only one bit different. The irreducible IDs are taken as attributes for encryption: S_i, where $1 \le i \le m$ and m denotes the number of irreducible IDs. After the coordinator runs the encryption algorithm for each attribute, he generates the encrypted messages for multicast: $\mathrm{ENC}(\mathrm{Params}, \mathrm{MK}, S_1, M)$, $\mathrm{ENC}(\mathrm{Params}, \mathrm{MK}, S_2, M), \ldots, \mathrm{ENC}(\mathrm{Params}, \mathrm{MK}, S_m, M)$.
For example, a group coordinator is going to multicast three members whose IDs are 001001, 001011, and 111000, respectively. After minimization through Quine-McCluskey algorithm, the IDs are reduced as 111000 and 0010x1, where x denotes a negligible bit. Next, the coordinator maps the reduced IDs with L_B as attributes S_1 and S_2. Consider $S_1 = \{B_0, \overline{B_2}, B_3, \overline{B_4}, \overline{B_5}\}$ and $S_2 = \{\overline{B_0}, \overline{B_1}, \overline{B_2}, B_3, B_4, B_5\}$.

With the attributes, the coordinator runs the encryption algorithm $\mathrm{ENC}(\mathrm{Params}, \mathrm{MK}, S_i, M)$ in detailed steps as follows:

(1) converting S_i's L_B into P_B,

(2) calculating $\mathrm{Sum}_{S_i} = \sum_{B_i \in S_i} p_{B_i}$,

(3) generating a random value $t_{S_i} \in Z_q^*$,

(4) calculating $C_{0S_i} = Me(g,g)^{\alpha t_{S_i} \mathrm{Sum}_{S_i}}$,

(5) calculating $C_{1S_i} = g^{\beta t_{S_i} \mathrm{Sum}_{S_i}}$,

(6) calculating $C_{2S_i} = g^{t_{S_i}}$,

(7) generating a ciphertext $\mathrm{CT}_{S_i} = \{S_i, C_{0S_i}, C_{1S_i}, C_{2S_i}\}$.

After group members receive the encrypted messages, they use their private key and the public parameter to run the decryption algorithm, $\mathrm{DEC}(\mathrm{Params}, \mathrm{SK}, \mathrm{CT}_{S_i})$:

(1) The group member verifies whether CT_{S_i}'s S_i matches his own attribute A_{ID}. If the two match, the member

proceeds to the next step. If they do not match, the decryption algorithm ends.

(2) He converts S_i's L_B into P_B and calculates the sum: $F = \prod_{A_{\mathrm{ID}} \in S_i} g^{\gamma p_{B_i}} = g^{\gamma \sum_{A_{\mathrm{ID}} \in S_i} p_{B_i}} = g^{\gamma \mathrm{Sum}_{S_i}}$.

(3) He calculates $A_1 = e(C_{1S_i}, D) = e(g^{\beta t_{S_i} \mathrm{Sum}_{S_i}}, g^{(\alpha+\gamma)/\beta}) = e(g, g)^{(\alpha+\gamma) t_{S_i} \mathrm{Sum}_{S_i}}$.

(4) He calculates $A_2 = e(C_{2S_i}, F) = e(g^{t_{S_i}}, g^{\gamma \mathrm{Sum}_{S_i}}) = e(g, g)^{\gamma t_{S_i} \mathrm{Sum}_{S_i}}$.

(5) He calculates $A_3 = A_1 / A_2 = e(g, g)^{\alpha t_{S_i} \mathrm{Sum}_{S_i}}$. $C_{0S_i} / A_3 = M e(g, g)^{\alpha t_{S_i} \mathrm{Sum}_{S_i}} / e(g, g)^{\alpha t_{S_i} \mathrm{Sum}_{S_i}} = M$.

2.4. Key-Update Stage. When a new member joins a group, the group coordinator generates a new group key SEK′ and uses the previous group key SEK to encrypt SEK′ to the new member. Then, he sends ID, A_{ID}, $\mathrm{SK}_{\mathrm{ID}}$, and SEK to the new member through a secure channel. If a member leaves a group, the rekeying process will be divided into two parts: the group coordinator and the group member.

The group coordinator has to run the following steps:

(1) Generating a random value $\alpha' \in Z_q^*$.

(2) Updating his master key $\mathrm{MK} = \{\beta, g^{\alpha'}, g^{\beta}, e(g, g)^{\alpha'}, L_B, P_B\}$.

(3) Generating a rekeying message $\mathrm{KU} = g^{(\alpha' - \alpha)/\beta}$.

(4) Running Quine-McCluskey algorithm to minimize the remaining members' IDs until there are m irreducible IDs; these IDs are taken as attributes S_i, $1 \le i \le m$.

(5) Using each attribute to run the encryption algorithm ENC(Params, MK, S_i, M).

After receiving the ciphertext, the group members run the following steps:

(1) Performing the decryption algorithm DEC(Params, SK, CT_{S_i}) and retrieving $\mathrm{KU} = M$.

(2) Multiplying KU by his private key's D and obtaining D', $D' = \mathrm{KU} \cdot D = g^{(\alpha' - \alpha)/\beta} \cdot g^{(\alpha+\gamma)/\beta} = g^{(\alpha' + \gamma)/\beta}$.

(3) Updating his session key $\mathrm{SEK}' = H_1(\mathrm{KU})$.

3. Security Issues in EGK

3.1. Malicious Member and Man-in-the-Middle Attack. In EGK, a malicious member may decrypt the group coordinator's message and send the forged message to the members who are associated with the same attributes. The victim members may update their keys and consequently cause desynchronization with the coordinator. Thus, the group coordinator is replaced and the malicious member plays the man in the middle. He takes control of the group. Detailed steps are as follows.

A malicious member decrypts CT_{S_i}, a valid ciphertext of earlier sessions. He calculates F, A_1, A_2, and A_3:

(1) He retrieves $e(g, g)^{\alpha t_{S_i} \mathrm{Sum}_{S_i}}$ from CT_{S_i} as A_3; that is, $A_3 = e(g, g)^{\alpha t_{S_i} \mathrm{Sum}_{S_i}}$.

(2) He generates a random value x and requires $\exists \alpha_2 \in Z_q^*$, so that $x = (\alpha_2 - \alpha)/\beta$; that is, $x\beta = \alpha_2 - \alpha$. He also forges a rekeying message $M' = g^x$.

(3) He multiplies M' by A_3 and the result is $C_0' = M' e(g, g)^{\alpha t_{S_i} \mathrm{Sum}_{S_i}}$.

(4) He sends a forged ciphertext $\mathrm{CT}_{S_i}' : \{S_i, C_{0S_i}', C_{1S_i}, C_{2S_i}\}$.

Therefore, the malicious member can bypass the encryption algorithm ENC and sends a forged ciphertext CT_{S_i}' without the group coordinator's master key MK. After receiving CT_{S_i}', the members run the decryption algorithm DEC. Detailed steps are as follows:

(1) Calculating $F = \prod_{A_{\mathrm{ID}} \in S_i} g^{\gamma P_{B_i}} = g^{\gamma \sum_{A_{\mathrm{ID}} \in S_i} P_{B_i}} = g^{\gamma \mathrm{Sum}_{S_i}}$.

(2) Calculating $A_1 = e(C_{1S_i}, D) = e(g, g)^{(\alpha+\gamma) t_{S_i} \mathrm{Sum}_{S_i}}$.

(3) Calculating $A_2 = e(C_{2S_i}, F) = e(g, g)^{\gamma t_{S_i} \mathrm{Sum}_{S_i}}$.

(4) Calculating $A_3 = A_1 / A_2 = e(g, g)^{\alpha t_{S_i} \mathrm{Sum}_{S_i}}$.

(5) Calculating $C_{0S_i}' / A_3 = M' e(g, g)^{\alpha t_{S_i} \mathrm{Sum}_{S_i}} / e(g, g)^{\alpha t_{S_i} \mathrm{Sum}_{S_i}} = M'$.

After decryption, they retrieve the rekeying message $M' = g^{(\alpha_2 - \alpha')/\beta}$ and process to rekey themselves. They calculate $D' = D \cdot M' = g^{(\alpha_2 + \gamma)/\beta}$ and then calculate the current session key $\mathrm{SEK}' = H(M')$.

Under such MITM attacks, the compromised members' private keys become $D = g^{(\alpha_2+\gamma)/\beta}$, which is desynchronous with their group coordinator's $g^{\alpha'}$ in his master key MK. Here, we assume that the group coordinator sends a new rekeying message $\mathrm{KU} = M = g^{(\alpha'' - \alpha')/\beta}$, uses Quine-McCluskey algorithm to minimize the members' IDs until there are n irreducible IDs and their attributes are S_i, $1 \le i \le n$, and multicasts the ciphertext to each member ENC(Params, MK, S_1, M), ENC(Params, MK, S_2, M), …, ENC(Params, MK, S_n, M). The compromised members receive their ciphertext and run the following steps for decryption:

(1) Calculation of $F = \prod_{A_{\mathrm{ID}} \in S_i} g^{\gamma p_{B_i}} = g^{\gamma \sum_{A_{\mathrm{ID}} \in S_i} p_{B_i}} = g^{\gamma \mathrm{Sum}_{S_i}}$.

(2) Calculation of $A_1 = e(C_{1S_i}, D) = e(g, g)^{(\alpha_2+\gamma) t_{S_i} \mathrm{Sum}_{S_i}}$.

(3) Calculation of $A_2 = e(C_{2S_i}, F) = e(g, g)^{\gamma t_{S_i} \mathrm{Sum}_{S_i}}$.

(4) Calculation of $A_3 = A_1 / A_2 = e(g, g)^{\alpha_2 t_{S_i} \mathrm{Sum}_{S_i}}$.

(5) Calculation of $C_{0S_i} / A_3 = M e(g, g)^{\alpha' t_{S_i} \mathrm{Sum}_{S_i}} / e(g, g)^{\alpha_2 t_{S_i} \mathrm{Sum}_{S_i}}$.

After the division in step (5), the compromised members are unable to retrieve the key-update message M. However, the malicious member can repeat the same rekeying process

and successfully update the group's private keys and session key. Then, the malicious member is able to replace the coordinator and take control of the group.

3.2. Full Modification of Group Coordinator's Ciphertext. A malicious member may modify the whole ciphertext, except S_i, in the following steps:

(1) Calculating $B_{1S_i} = e(C_{1S_i}, M) = e(g^{\beta t_{S_i} \mathrm{Sum}_{S_i}}, g^{(\alpha'-\alpha)/\beta}) = e(g,g)^{(\alpha'-\alpha) t_{S_i} \mathrm{Sum}_{S_i}}$.

(2) Calculating $B_{2S_i} = C_{0S_i} \cdot B_1 / M = e(g,g)^{\alpha t_{S_i} \mathrm{Sum}_{S_i}} \cdot e(g,g)^{(\alpha'-\alpha) t_{S_i} \mathrm{Sum}_{S_i}} = e(g,g)^{\alpha' t_{S_i} \mathrm{Sum}_{S_i}}$.

(3) Calculating $B_{3S_i} = (B_{2S_i})^x = e(g, g)^{\alpha' t_{S_i} \mathrm{Sum}_{S_i} \cdot ((\alpha_2 - \alpha')/\beta)} = e(g,g)^{((\alpha_2-\alpha')\alpha' t_{S_i} \mathrm{Sum}_{S_i})/\beta}$.

(4) Multiplying M' by B_{3S_i} to forge $C'_{0S_i} = M' \cdot B_{3S_i} = M' e(g,g)^{((\alpha_2-\alpha')\alpha' t_{S_i} \mathrm{Sum}_{S_i})/\beta}$.

(5) Calculating $C'_{1S_i} = (C_{1S_i})^x = (C_{1S_i})^{(\alpha_2-\alpha')/\beta} = g^{(\alpha_2-\alpha') t_{S_i} \mathrm{Sum}_{S_i}}$.

(6) Calculating $C'_{2S_i} = (C_{2S_i})^x = (C_{2S_i})^{(\alpha_2-\alpha')/\beta} = g^{((\alpha_2-\alpha') t_{S_i})/\beta}$.

(7) Sending the forged message CT'_{S_i}: $\{S_i, C'_{0S_i}, C'_{1S_i}, C'_{2S_i}\}$.

After receiving CT'_{S_i}, the group members run the decryption algorithm DEC(Params, SK, CT_{S_i}):

(1) The group members verify whether the received S_i matches their own A_{ID}. If the two are verified, they proceed to the next step. Otherwise, they end the algorithm.

(2) They convert the S_i-associated L_B into P_B and then calculate the sum $F = \prod_{A_{\mathrm{ID}} \in S_i} g^{\gamma p_{B_i}} = g^{\gamma \sum_{A_{\mathrm{ID}} \in S_i} p_{B_i}} = g^{\gamma \mathrm{Sum}_{S_i}}$.

(3) They calculate $F = \prod_{A_{\mathrm{ID}} \in S_i} g^{\gamma p_{B_i}} = g^{\gamma \sum_{A_{\mathrm{ID}} \in S_i} p_{B_i}} = g^{\gamma \mathrm{Sum}_{S_i}}$.

(4) They calculate $A_1 = e(C'_{1S_i}, D) = e(g^{(\alpha_2-\alpha') t_{S_i} \mathrm{Sum}_{S_i}}, g^{(\alpha'+\gamma)/\beta}) = e(g,g)^{((\alpha_2-\alpha')(\alpha'+\gamma) t_{S_i} \mathrm{Sum}_{S_i})/\beta}$.

(5) They calculate $A_2 = e(C'_{2S_i}, F) = e(g^{((\alpha_2-\alpha') t_{S_i})/\beta}, g^{\gamma \mathrm{Sum}_{S_i}}) = e(g,g)^{((\alpha_2-\alpha')\gamma t_{S_i} \mathrm{Sum}_{S_i})/\beta}$.

(6) They calculate $A_3 = A_1/A_2 = e(g,g)^{((\alpha_2-\alpha')(\alpha'+\gamma) t_{S_i} \mathrm{Sum}_{S_i})/\beta} / e(g,g)^{((\alpha_2-\alpha')\gamma t_{S_i} \mathrm{Sum}_{S_i})/\beta} = e(g,g)^{((\alpha_2-\alpha')(\alpha'+\gamma) t_{S_i} \cdot \mathrm{Sum}_{S_i} - (\alpha_2-\alpha')\gamma t_{S_i} \mathrm{Sum}_{S_i})/\beta} = e(g,g)^{((\alpha_2-\alpha')\alpha' t_{S_i} \mathrm{Sum}_{S_i})/\beta}$.

(7) They calculate $C'_{0S_i}/A_3 = M' e(g,g)^{((\alpha_2-\alpha')\alpha' t_{S_i} \mathrm{Sum}_{S_i})/\beta} / e(g,g)^{((\alpha_2-\alpha')\alpha' t_{S_i} \mathrm{Sum}_{S_i})/\beta} = M'$.

After the decryption procedure, they retrieve $M' = g^{(\alpha_2-\alpha')/\beta}$, calculate $D' = D \cdot M' = g^{(\alpha_2+\gamma)/\beta}$, and update their current session key $\mathrm{SEK}' = H(M')$.

In EGK, even though a malicious member is unable to decrypt his group coordinator's ciphertext, he can forge a message and then try the steps of Sections 3.1 and 3.2. Maybe some members are able to decrypt two ciphertexts, for example, S_i and S_j, but they can still receive the forged message. Since they do not verify the sender, they just retrieve M' and update their keys and hence the desynchronization between them and their group coordinator.

4. Secure Key Distribution Scheme for Multicast Communication

4.1. Initial Stage. At this stage a group coordinator creates a multicast group with N maximum members, generates system parameters G_0, G_1, q, g, and e, defines two hash functions H_1 and H_2, generates three random values α, β and t, requires $\alpha, \beta, t \in Z_q^*$, sets the length of ID as $n = \log N$, creates $2\log_2 N$ bit-assignment sets $L_B = \{B_0, \overline{B_0}, B_1, \overline{B_1}, \ldots, B_{n-1}, \overline{B_{n-1}}\}$, where B_i denotes that the ith bit of ID is 1 and $\overline{B_i}$ denotes that the ith bit of ID is 0, and creates sets of bit-assignment secrets $P_B = \{p_{B_0}, \overline{p_{B_0}}, s_{B_1}, \overline{p_{B_1}}, \ldots, p_{B_{n-1}}, \overline{p_{B_{n-1}}}\}$, where $p_{B_i}, \overline{p_{B_i}} \in Z_q$. P_B maps L_B. That is, p_{B_0} maps B_0, $\overline{p_{B_0}}$ maps $\overline{B_0}$, and so on. Further, a group coordinator generates his master key $\mathrm{MK} = \{\beta, t, g^\alpha, g^\beta, e(g,g)^\alpha, L_B, P_B\}$ and the public parameter $\mathrm{Params} = \{G_0, e, g, P_{\mathrm{pub}}, H_1, H_2\}$, where $P_{\mathrm{pub}} = g^t$.

4.2. Key Distribution. When a new member joins the group, the group coordinator assigns a unique ID to the member and associates him with an attribute A_{ID}. Then, the group coordinator runs KeyGen(MK, A_{ID}) to generate a key:

(1) He generates a random value $\gamma \in Z_q^*$.

(2) He calculates $D = g^{(\alpha+\gamma)/\beta}$.

(3) He uses A_{ID} to compute $\forall B_i \in A_{\mathrm{ID}}: D_B = g^{\gamma B_i}$.

(4) He maps P_B with L_B, $\forall B_i \in A_{\mathrm{ID}} = g^{\gamma p_{B_i}}$.

(5) He generates a private key $\mathrm{SK}_{\mathrm{ID}} = \{D = g^{(\alpha+\gamma)/\beta}, \forall B_i \in A_{\mathrm{ID}} : D_B = g^{\gamma p_{B_i}}\}$.

At last, the group coordinator multicasts ID, A_{ID}, $\mathrm{SK}_{\mathrm{ID}}$, and current session key SEK to the new member through a secure channel.

4.3. Encryption and Decryption Algorithm. When a group coordinator needs to multicast his group, he assigns a unique set of bit-assignment to each group member, so that each member is able to decrypt their ciphertext. Also, the coordinator uses the Quine-McCluskey algorithm to minimize remaining members' IDs and then calculates attributes S_i. He runs ENC to encrypt S_i and members run DEC to decrypt it.

The group coordinator runs ENC(Params, MK, S_i, M) that

(1) calculates $h = H_2(M)$,

(2) converts the S_i-associated L_B into P_B,

(3) calculates $\mathrm{Sum}_{S_i} = \sum_{B_i \in S_i} p_{B_i}$,

(4) calculates $C_{0S_i} = Me(h, g)^{\alpha t \mathrm{Sum}_{S_i}}$,

(5) calculates $C_{1S_i} = h^{\beta t \mathrm{Sum}_{S_i}}$,

(6) calculates $C_{2S_i} = h^t$,

(7) generates CT_{S_i}: $\{S_i, C_{0S_i}, C_{1S_i}, C_{2S_i}\}$.

After receiving the ciphertext, group members verify whether their own A_{ID} matches S_i's bit-assignment. If verified, they run the decryption algorithm DEC(Params, SK, CT_{S_i}). Besides, during decryption, they have to check if the sender of M is their group coordinator. Steps are as follows:

(1) Conversion of the S_i-associated L_B into P_B.

(2) Calculating $F = \prod_{A_{\mathrm{ID}} \in S_i} g^{\gamma p_{B_i}} = g^{\gamma \sum_{A_{\mathrm{ID}} \in S_i} p_{B_i}} = g^{\gamma \mathrm{Sum}_{S_i}}$.

(3) Calculating $A_1 = e(C_{1S_i}, D) = e(h, g)^{(\alpha+\gamma) t \mathrm{Sum}_{S_i}}$.

(4) Calculating $A_2 = e(C_{2S_i}, F) = e(h, g)^{\gamma t \mathrm{Sum}_{S_i}}$.

(5) Calculating $A_3 = A_1 / A_2 = e(h, g)^{\alpha t \mathrm{Sum}_{S_i}}$.

(6) Dividing C_{0S_i} by A_3 and the result is $M = C_{0S_i} / A_3$.

(7) Calculating $h = H_2(M)$.

(8) Verifying if $e(h, P_{\mathrm{pub}}) = e(h, g^t) = e(C_{2S_i}, g)$.

4.4. Key-Update Stage. If a new member joins a group, the group coordinator generates a new session key SEK′ and uses the previous session key SEK to encrypt it to his group members. Also, through a secure channel, the coordinator multicasts ID, AID, $\mathrm{SK}_{\mathrm{ID}}$, and SEK to the members. If a member leaves a group, the coordinator and his members proceed to different steps.

Group coordinator performs the following:

(1) He generates a random value $\alpha' \in Z_q^*$.

(2) He updates his master key $\mathrm{MK} = \{\beta, g^{\alpha'}, g^{\beta}, e(g, g)^{\alpha'}, L_B, P_B\}$.

(3) He generates a rekeying message $\mathrm{KU} = g^{(\alpha' - \alpha)/\beta}$.

(4) He uses the Quine-McCluskey algorithm to minimize remaining members' IDs until there are m irreducible IDs and these IDs are taken as attributes S_i, $1 \le i \le m$.

(5) He runs ENC(Params, MK, S_i, M) for each attribute.

Group members perform the following:

(1) After receiving the ciphertext, each group member runs DEC(Params, SK, CT_{S_i}) and retrieves $M = \mathrm{KU}$.

(2) They multiply KU by D and the result is $D' = \mathrm{KU} \cdot D = g^{(\alpha' - \alpha)/\beta} \cdot g^{(\alpha+\gamma)/\beta} = g^{(\alpha'+\gamma)/\beta}$.

(3) They update their session key $\mathrm{SEK}' = H_1(\mathrm{KU})$.

5. Security Analysis

5.1. Malicious Member and MITM Attack. We assume a malicious member is in a group coordinator's communication group. He receives the latest rekeying message and then forges a fake one to other group members. The forged message may arrive before the group coordinator's rekeying message and successfully rekey the members. By this, the malicious member can cause asynchrony between group members and their coordinator.

After receiving group coordinator's CT_{S_i}, the malicious member retrieves S_i, C_{0S_i}, C_{1S_i}, and C_{2S_i} and calculates A_1, A_2, and A_3. He tries to forge a message $C'_{0S_i} = M' \cdot A_3$ and sends the fake ciphertext $\mathrm{CT}'_{S_i} = \{S_i, C'_{0S_i}, C_{1S_i}, C_{2S_i}\}$ to other members. Following the decryption algorithm, the members decrypt the fake ciphertext CT'_{S_i} and retrieve M'. According to the 7th step of decryption, they calculate $h = H_2(M')$. At the 8th step, they calculate $e(h, P_{\mathrm{pub}}) = e(h, g^t)$, which does not match $e(C_{2S_i}, g)$, hence an unsuccessful attack.

5.2. Malicious Member and Asynchrony. We assume a malicious member belongs to a group coordinator's communication group. He is able to decrypt and retrieve the coordinator's rekeying message. He forges a rekeying message and sends it to other group members. CT_{S_i} is the previous ciphertext from the group coordinator:

$$\mathrm{CT}_{S_i}: \left\{S_i, C_{0S_i} = Me\,(h, g)^{\alpha t \mathrm{Sum}_{S_i}}, C_{1S_i} = h^{\beta t \mathrm{Sum}_{S_i}}, C_{2S_i} = h^t\right\}, \quad (1)$$

$$M = \mathrm{KU} = g^{(\alpha' + \alpha)/\beta}.$$

With CT_{S_i} and M, the group members have updated their private keys $D = g^{(\alpha'+\gamma)/\beta}$. The malicious member forges a rekeying message M'. Since he has had H_2, he uses it to calculate $H_2(M')$ and tries to generate $C'_{0S_i} = M' e(H_2(M'), g)^{\alpha' t \mathrm{Sum}_{S_i}}$. He has to use CT_{S_i} to run the following steps:

(1) Summing up the previous attributes S_i; thus, $F = \prod_{A_{\mathrm{ID}} \in S_i} g^{\gamma p_B} = g^{\gamma \sum_{A_{\mathrm{ID}} \in S_i} p_B} = g^{\gamma \mathrm{Sum}_{S_i}}$.

(2) Calculating $A'_1 = e(C_{1S_i}, D) = e(H_2(M), g)^{(\alpha'+\gamma) t \mathrm{Sum}_{S_i}}$.

(3) Calculating $A_2 = e(C_{2S_i}, F) = e(H_2(M), g)^{\gamma t \mathrm{Sum}_{S_i}}$.

(4) Calculating $A'_3 = A'_1 / A_2 = e(H_2(M), g)^{\alpha' t \mathrm{Sum}_{S_i}}$.

Since $A'_3 = e(H_2(M), g)^{\alpha' t \mathrm{Sum}_{S_i}}$, it does not match $e(H_2(M'), g)^{\alpha' t \mathrm{Sum}_{S_i}}$.

5.3. Backward Secrecy. If a group coordinator removes a member from the group, he updates his own master key and runs the encryption algorithm ENC to generate a rekeying message KU to the remaining members. After receiving KU, the members update their private keys $\mathrm{SK} \cdot D' = \mathrm{SK} \cdot D \cdot \mathrm{KU}$

Table 1: Comparison of computational load.

	EGK		Our scheme	
	Encryption	Decryption	Encryption	Decryption
Coordinator	$(e + 3Ep)m$	0	$(h + e + 3Ep)m$	0
Member	0	$3e$	0	$h + 4e$

Table 2: Storage.

	Master key	Public parameter	Session key
Coordinator	1	0	1
Member	0	N	N
Total	1	N	$N + 1$

and calculate a new session key $\text{SEK}' = H_2(\text{KU})$. Because the removed member does not have KU, he cannot calculate a new private key and session key.

5.4. Forward Secrecy. When a new member joins a group, the group coordinator generates a new session key SEK' and uses the previous session key SEK to encrypt SEK' to his group members. Then, he sends SEK' to the new member through a secure channel. In doing so, the new member is unable to access previous messages.

6. Performance Evaluation

In our secure multicast group communication protocol, a group coordinator has to store a set of a public parameter, a master key, and a session key, that is, {Params, MK, SEK}. It consists of five real numbers, two hash functions, one mapping function, and two sets of bit-assignments.

As for our computational load, a group coordinator calculates one hash function, one mapping, and three exponentials when running the encryption algorithm. When a group member decrypts a ciphertext, the decryption algorithm requires four mappings and one hash. As listed in Table 1, if there are m attributes after the minimization, our key-updating requires m times of encryption, that is, m hash functions, m mappings, and $3m$ exponentials.

Compared with EGK, our encryption algorithm requires sender verification; therefore, it needs to compute one more hash. And our decryption algorithm requires group members to verify messages, so it needs to compute one more hash and one more mapping.

Take Facebook as an example. We assume that it uses our scheme; its member limit is 5000 ($N = 5000$); each user is a group coordinator; each user is also a member of his friends' groups. Thus, each user has a master key, $(N + 1)$ public parameters, and $(N + 1)$ session keys, as shown in Table 2.

In our scheme, a group coordinator's communication load is in linear increase with the minimized attributes. If there are m attributes after minimization, he has to generate m rekeying messages. Each message consists of one S_i and three variables C_{0S_i}, C_{1S_i}, and C_{2S_i} (see Table 3).

Despite having one more hash and one more public parameter compared to EGK, our system's storage requirement is the same as EGK's. And the two schemes' communication loads are also the same.

Table 3: Communication loads for rekeying.

	EGK	Our scheme
Coordinator	$(S_i, C_{0S_i}, C_{1S_i}, C_{2S_i})m$	$(S_i, C_{0S_i}, C_{1S_i}, C_{2S_i})m$

7. Conclusion

In this paper, we analyze Zhou and Huang's key distribution scheme for multicast group communication and find it prone to MITM and desynchronization attacks. These security issues can cause a group coordinator to lose the control over his group and fail to rekey his members. We propose a scheme to enhance the security of EGK. Our scheme guarantees forward and backward secrecy, prevents message modification and forgery during rekeying, requires sender verification, and therefore prevents MITM attacks in group communication.

We hope our secure multicast group communication scheme can be applied to cloud services and social networking sites, so that users can enjoy these services in a secure environment without further loads. And users' privacy and data integrity can also be secured.

Notations

q: A large prime number
G_0: A cyclic additive group of prime order q and generator g
G_1: A cyclic multiplicative group of prime order q
g: Generator, $g \in G_0$
e: A bilinear map $e, G_0 \times G_0 \to G_1$
H: One-way hash function
α, β, γ: Random values, $\alpha, \beta, \gamma \in Z_q^*$
ID: Identifier of group members, $\text{ID} = b_n b_{n-1} \cdots b_0$
L_B: A set of bit-assignments $L_B = \{B_0, \overline{B_0}, B_1, \overline{B_1}, \ldots, B_{n-1}, \overline{B_{n-1}}\}$; B_i: the ith bit of ID is 1; $\overline{B_i}$: the ith bit of ID is 0.
P_B: A set of bit-assignments secrets $P_B = \{p_{B_0}, \overline{p_{B_0}}, p_{B_1}, \overline{p_{B_1}}, \ldots, p_{B_{n-1}}, \overline{p_{B_{n-1}}}\}$; $p_{B_i}, \overline{p_{B_i}} \in Z_q$; P_B maps L_B
Params: Public parameter $\text{Params} = \{G_0, e, g, H\}$
MK: Group coordinator's master key $\text{MK} = \{\beta, g^\alpha, g^\beta, e(g, g)^\alpha, L_B, S_B\}$
SK_{ID}: Private keys that a group coordinator multicasts to members of specific ID, $\text{SK}_{\text{ID}} = \{D = g^{(\alpha+\gamma)/\beta}, \forall B_i \in A_{\text{ID}} : D_B = g^{rs_{B_i}}\}$
SEK: Session key
A_{ID}: An attribute that a group coordinator gives to a member of a specific ID (each attribute is the bit-assignment of the ID).

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research was supported by the National Science Council of Taiwan under Grant no. MOST104-2221-E-130-009.

References

[1] X. Liang, K. Zhang, X. Shen, and X. Lin, "Security and privacy in mobile social networks: challenges and solutions," *IEEE Wireless Communications*, vol. 21, no. 1, pp. 33–41, 2014.

[2] L. A. Cutillo, R. Molva, and T. Strufe, "Safebook: a privacy-preserving online social network leveraging on real-life trust," *IEEE Communications Magazine*, vol. 47, no. 12, pp. 94–101, 2009.

[3] H. Abelson, R. Anderson, S. M. Bellovin et al., "Keys under doormats: mandating insecurity by requiring government access to all data and communications," *Journal of Cybersecurity*, 2015.

[4] S. Deering, "Host extensions for IP multicasting," *RFC 1112, IETF*, August 1989.

[5] I. Chang, R. Engel, D. Kandlur, D. Pendarakis, and D. Saha, "Key management for secure Internet multicast using Boolean function minimization techniques," in *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societie (INFOCOM '99)*, pp. 689–698, Sharm El Sheik, Egypt, March 1999.

[6] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[7] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1214–1221, 2011.

[8] Z. Wan, J. Liu, and R. H. Deng, "HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 743–754, 2012.

[9] B. Qin, R. H. Deng, S. Liu, and S. Ma, "Attribute-based encryption with efficient verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1384–1393, 2015.

[10] Z. Liu, Z. Cao, and D. S. Wong, "White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 76–88, 2013.

[11] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, Calif, USA, 2007.

[12] Z. Zhou and D. Huang, "An optimal key distribution scheme for secure multicast group communication," in *Proceedings of the IEEE INFOCOM*, pp. 1–5, San Diego, Calif, USA, March 2010.

[13] D. Huang, S. Ata, and D. Medhi, "Establishing secure virtual trust routing and provisioning domains for future internet," in *Proceedings of the 53rd IEEE Global Communications Conference*, pp. 1–6, IEEE, Miami, Fla, USA, December 2010.

[14] H. Jia, Y. Chen, X. Mao, and R. Dou, "Efficient and scalable multicast key management using attribute based encryption," in *Proceedings of the IEEE International Conference on Information Theory and Information Security (ICITIS '10)*, pp. 426–429, Beijing, China, December 2010.

[15] D. Huang, Z. Zhou, L. Xu, T. Xing, and Y. Zhong, "Secure data processing framework for mobile cloud computing," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM '11)*, pp. 614–618, Shanghai, China, April 2011.

[16] E. McCluskey, "Minimization of Boolean functions," *The Bell System Technical Journal*, vol. 35, no. 6, pp. 1417–1444, 1956.

Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2016, Article ID 9029745, 16 pages
http://dx.doi.org/10.1155/2016/9029745

Research Article

A New Scalar Quantization Method for Digital Image Watermarking

Yevhen Zolotavkin and Martti Juhola

Computer Science, School of Information Sciences, University of Tampere, Kanslerinrinne 1, 33014 Tampere, Finland

Correspondence should be addressed to Yevhen Zolotavkin; zhzolot@countermail.com

Received 7 October 2015; Revised 18 January 2016; Accepted 24 January 2016

Academic Editor: Mazdak Zamani

A new technique utilizing Scalar Quantization is designed in this paper in order to be used for Digital Image Watermarking (DIW). Efficiency of the technique is measured in terms of distortions of the original image and robustness under different kinds of attacks, with particular focus on Additive White Gaussian Noise (AWGN) and Gain Attack (GA). The proposed technique performance is affirmed by comparing with state-of-the-art methods including Quantization Index Modulation (QIM), Distortion Compensated QIM (DC-QIM), and Rational Dither Modulation (RDM). Considerable improvements demonstrated by the method are due to a new form of distribution of quantized samples and a procedure that recovers a watermark after GA. In contrast to other known quantization methods, the detailed method here stipulates asymmetric distribution of quantized samples. This creates a distinctive feature and is expressed numerically by one of the proposed criteria. In addition, several realizations of quantization are considered and explained using a concept of Initial Data Loss (IDL) which helps to reduce watermarking distortions. The procedure for GA recovery exploits one of the two criteria of asymmetry. The accomplishments of the procedure are due to its simplicity, computational lightness, and sufficient precision of estimation of unknown gain factor.

1. Introduction

In modern communications, multimedia plays significant role. Ownership of multimedia data is important and needs to be protected [1]. As a part of nowadays popular multimedia content, digital images are an important class. A protection of digital rights of an owner is implemented by Digital Image Watermarking (DIW). A watermark that is inserted into an image has to be robust [2] as well as invisible [3].

Among the popular and efficient techniques in DIW, Quantization Index Modulation (QIM) is widely used in blind watermarking where neither original media nor watermark is known to the receiver [4, 5]. One of the aspects of robustness of QIM is evaluated by attacking a watermarked image with Additive White Gaussian Noise (AWGN). Unfortunately, all the known on practice implementations of QIM are far from achieving the channel capacity limit that was first derived in [6].

Several different QIM-related approaches are known. Some state-of-the-art realizations will be outlined briefly. According to QIM, intervals of equal length Δ are mapped on the real number line. The oldest known approach is to replace all the original coefficients inside every interval with one of the two endpoints of that interval. The selection of the endpoint depends on a bit of a watermark [7]. The main disadvantage is that for high intensity of noise and the capacity of the oldest QIM is much lower than the theoretical limit. In a more advanced realization of DC-QIM, coefficients from every original interval are mapped into two disjoint subintervals. The gap between the subintervals is controlled by parameter α, $0 \leq \alpha \leq 1$ [8]. Assuming that initial distribution inside original interval and target distributions in subintervals are uniform, the mapping in accordance to DC-QIM is optimal in terms of Mean Square Error (MSE) of quantization. In order to maximize capacity for a given MSE under AWGN of different intensity, parameters Δ, α have to be adjusted. Nevertheless, the limit defined in [6] is still well above the one achievable by DC-QIM.

Not all the original coefficients in each interval need to be quantized. This idea has been explored by the authors of Forbidden Zone Data Hiding (FZDH) [9]. Another idea was proposed by the authors of Thresholded Constellation

Modulation (TCM) that uses two different quantization rules to modify coefficients inside the original interval [10].

Despite sufficient robustness of QIM under AWGN, the limitation is that synchronization is required in order to reconstruct intervals that are necessary to extract (or decode) a watermark. A type of distortion which scales all the watermarked coefficients is called Gain Attack (GA). The scaling factor might be close to 1 and cause very little visual distortion, but it is unknown to the receiver which causes asynchronous extraction. Retrieval of the watermark is usually complicated by AWGN that follows GA [11].

Improvement of QIM performance under GA is the task of numerous known approaches [12]. Most of them can be classified into two groups where the main idea of the first group is to estimate the unknown factor [13] while the idea of the second is to quantize coefficients of a different kind that are invariant to scaling of original signal.

The solution proposed in [11] contributes to robustness enhancement in case of GA and a constant offset attack followed by AWGN. A pilot signal is embedded for this purpose. Fourier analysis is used during extraction to estimate the gain factor and the offset. Another method of recovery after GA and AWGN is proposed in [14]. It uses information about dither sequence and applies Maximum Likelihood (ML) procedure to estimate the scaling factor.

Watermarking that is invariant to GA demands more complex transform of original signal (e.g., nonlinear) to obtain coefficients. One of the most popular watermarking methods in that category is Rational Dither Modulation (RDM) [15]. For a particular coefficient, a ratio that depends on a norm of other coefficients is being quantized instead of a coefficient itself. In order to quantize the ratio, RDM utilizes the simplest QIM scheme. This implies that the performance of RDM under AWGN (without GA) is close to the simplest QIM. Among others recent blind watermarking methods robust to GA are, for example, detailed in [16–18].

A new scalar QIM-based watermarking method is proposed in this paper. It provides high robustness under conditions of AWGN and GA. Among the new features of the method are IDL and a new form of distribution of quantized samples.

The organization of the rest of the paper is as follows. Section 2 explains the choice of the distribution of quantized samples and contains description of the procedure of recovery after GA. Concept of IDL and quantization model are described in Section 3 using formal logic approach. The aspects of analytic-based estimation of robustness under AWGN are discussed in Section 4. Next, Section 5 contains experimental results obtained under AWGN and GA. Discussion of the details of the experiment and comparison of the performance are given in Section 6. Section 7 concludes the paper. The list of the key variables and their meaning is given in Nomenclature section.

2. Distribution of Quantized Samples and Procedure for Recovery after GA

An asymmetric distribution of quantized samples is proposed and parametrized in this section. Asymmetry is the quality that can be easily expressed quantitatively. Under symmetric attack, like AWGN, such quantitative index remains sufficiently indicative. On the other hand, it can be affected by GA. Such semifragility is favorable for restoration of the right condition for decoding. The restoration is done by the procedure for recovery after GA which uses criterion of asymmetry. Compared to the known estimation procedures [14], the one proposed in this section depends on a single variable which is the unknown gain factor. This makes the technique simple and more precise.

For encoding, in our case, asymmetric distribution requires substantially more variables for description compared to common QIM methods. Because of that, it is advisable to refer to Nomenclature section.

2.1. Distribution of Quantized Samples. Symbol Σ will be used to denote a random variable whose domain is the space of original coefficients of a host. A particular realization of Σ will be denoted as ς. We will further consider manipulation of original values ς that are in some kth interval of size Δ and its left endpoint is l_Δ^k. Such an interval is referred further as embedding interval. For any $\varsigma \in [l_\Delta^k, l_\Delta^k + \Delta]$ we define $x = \varsigma - l_\Delta^k$ and X will be used to denote a random variable which represents x. The value of Δ should be small enough so that the distribution of X can be considered uniform. A random variable that represents quantized coefficients inside kth interval is denoted as X' and its realization is denoted as x'. Each pair of an original x and corresponding quantized x' belongs to the same kth embedding interval so that an absolute shift is never larger than Δ. Correspondingly, a random variable that represents quantized coefficients on the whole real number line is denoted as Σ' and its realization is denoted as ς'.

In order to provide efficient recovery after GA, we propose the following asymmetric distribution of quantized samples x' inside kth embedding interval (Figure 1(a)):

$$f\left(x'\right) = \begin{cases} \left(\gamma_0 + \eta_1\right) f_0\left(x'\right), & \text{if } x' \in \left[0, \Delta\left(\beta - \alpha\right)\right], \\ \left(\varphi_1 + \vartheta_0\right) f_1\left(x'\right), & \text{if } x' \in \left[\Delta\beta, \Delta\right], \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where $f_0(x')$ and $f_1(x')$ are two different kinds of truncated distributions defined as

$$f_0\left(x'\right) = \begin{cases} cx' + \tau, & \text{if } x' \in \left[0, \Delta\left(\beta - \alpha\right)\right], \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

$$f_1\left(x'\right) = \begin{cases} g, & \text{if } x' \in \left[\Delta\beta, \Delta\right], \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The other parameters are constrained in the following way: $0 \le \alpha \le \beta \le 1$, $\gamma_0 + \vartheta_0 + \varphi_1 + \eta_1 = 1$ (see Nomenclature section). The meaning of parameters $\gamma_0, \vartheta_0, \varphi_1, \eta_1$ will be discussed later in Section 3. In Figure 1(b) we can see the distribution of the quantized coefficients outside kth embedding interval as well.

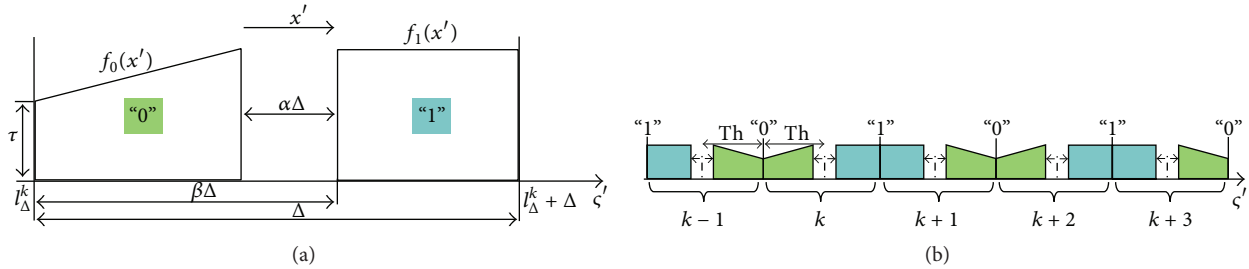


FIGURE 1: Distribution of the quantized coefficients: (a) Inside kth embedding interval. (b) In five consecutive intervals.

2.2. Procedure for GA Recovery. It is assumed that under GA the original length of embedding interval Δ is altered by unknown gain factor λ and the resulting length is $\tilde{\Delta} = \lambda\Delta$. In addition to that, AWGN attack is applied. The procedure for GA recovery is the estimator whose result is based on a criterion having higher values for the right length $\tilde{\Delta}$ of embedding interval. The uniqueness of the distribution of quantized samples is exploited by two different criteria C_1 and C_2. The procedure itself represents a brute force approach that substitutes guessed values $\tilde{\Delta}'$ of the length of embedding interval into a criterion. Guessed value of $\tilde{\Delta}'$ which maximizes it (C_1 or C_2) should be selected:

$$\tilde{\Delta}'' = \arg\max_{\{\tilde{\Delta}'\}} C_{1,2}\left(\tilde{\Delta}'\right), \tag{4}$$

where $\tilde{\Delta}''$ is the final output of the procedure. Some interval $[\tilde{\Delta}'_{\min}, \tilde{\Delta}'_{\max}]$ for guessed values $\tilde{\Delta}'$ should be defined in advance. For instance, $\tilde{\Delta}'_{\min} = 0.9\Delta$ and $\tilde{\Delta}'_{\max} = 1.1\Delta$ works well in most cases because the diapason of scaling factor λ is quite limited on practice.

For each particular value $\tilde{\Delta}'$, the index defined according to the criterion is calculated by projecting noisy quantized samples ς'_n on a single embedding interval:

$$x'_n = \begin{cases} \varsigma'_n \bmod \tilde{\Delta}', & \text{if } \left\lfloor \dfrac{\varsigma'_n - l_\Delta^k}{\tilde{\Delta}'} \right\rfloor \bmod 2 = 0, \\ \tilde{\Delta}' - \left(\varsigma'_n \bmod \tilde{\Delta}'\right), & \text{otherwise.} \end{cases} \tag{5}$$

This is needed to reconstruct the distribution of quantized samples inside embedding interval.

Two criteria are proposed for the assessment of the distribution of random variable $X'_n \in [0, \tilde{\Delta}']$ (subscript "n" means affected by noise):

$$C_1\left(\tilde{\Delta}'\right) = \left| \frac{\text{median}\left(X'_n\right)}{\tilde{\Delta}'} - 0.5 \right|,$$
$$C_2\left(\tilde{\Delta}'\right) = \left| \frac{\mu_w\left(X'_n\right)}{\left(\tilde{\Delta}'\right)^w} \right|, \quad w = 2m+1,\ m \in \mathbb{N}. \tag{6}$$

Here, μ_w is the wth central moment. Odd moments are zero for symmetric distributions, but for asymmetric distributions their values can be sufficiently large. If the assumption about $\tilde{\Delta}$ is wrong, then the values of both criteria are low. In that case the distribution of X'_n is very close to uniform (which is symmetric). This is because of the effect caused by GA on calculation of x'_n in (5). Nevertheless, the distribution of X'_n demonstrates asymmetry in case $\tilde{\Delta}'$ is close to $\tilde{\Delta}$. The explanation is that the distribution of quantized samples inside embedding interval (before GA is introduced) is indeed asymmetric. In spite of utilization of brute force optimization, the procedure is simple and the computational demand is low. On practice, the number of brute force steps is much smaller than the number of quantized elements. Therefore, the complexity is $O(n)$ in that case. For instance, for recovery with high accuracy it is enough to perform 10^3 brute force steps with values from the interval $[\tilde{\Delta}'_{\min}, \tilde{\Delta}'_{\max}]$.

3. Quantization

A quantization model is introduced in this section. In order to represent it in a compact form, we combine all the quantization conditions in a single logical expression. Previously proposed distribution of quantized samples is assured. However, additional parameter of the quantization model implies different distribution of the samples associated with labels "0" and "1."

3.1. Two Approaches for Quantization. Quantized samples are modified according to the model described in this subsection. A watermark bit is denoted as b. Each sample with value x inside kth embedding interval has index $i \in \mathbb{N}$ according to its order in the host sequence. During watermarking a bit is assigned to each index i. Different frameworks might be used for description of the quantization model. We will use first order predicate logic to describe our approach. This choice can be reasoned as follows. A closed-from expression has to be defined for quantization and it is important to show that the derived solution minimizes MSE between initial and target distribution. The kind of proposed target distribution is not common for QIM-based watermarking methods. Therefore, we find it necessary to explain in detail the process of derivation of quantization expression. Also, samples interpreting "0" should be quantized in a different way to samples interpreting "1." Predicate logic is a suitable

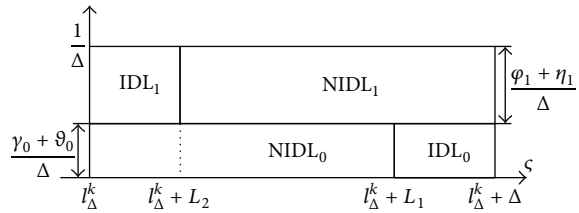


Figure 2: Scheme of labeling and distribution of original samples prior to quantization.

tool for description of embedding because logical construction can incorporate all the possible quantization conditions in a compact form.

Two-place predicate E is to denote correspondence between some index and the value of coefficient. For example, Eix is true if a coefficient with order i has value x. We will further use notation of the set $\mathbf{E}$ which contains all the pairs (x, i) that provide true value of Eix. One-place predicate B is to denote bit value assigned to a coefficient with particular index. For instance, Bi is true if watermark bit $b = 1$ is assigned to a coefficient with index i and $\sim Bi$ is true if $b = 0$. Two-place predicates X_0 or X_1 will be used to define that some ith sample with value x has label "0" or "1," respectively:

$$\left(X_0 ix \equiv (Eix \& \sim Bi)\right), \quad (\forall i)(\forall x), \tag{7}$$

$$\left(X_1 ix \equiv (Eix \& Bi)\right), \quad (\forall i)(\forall x). \tag{8}$$

Sets $\mathbf{X}_0$ and $\mathbf{X}_1$ contain all the pairs (x, i) that provide true values of $X_0 ix$ and $X_1 ix$, respectively. Initial PDFs of X inside $\mathbf{X}_0$, $\mathbf{X}_1$, and $\mathbf{E}$ are considered to be uniform: $f_{\mathbf{X}_0}(x) = f_{\mathbf{X}_1}(x) = f_{\mathbf{E}}(x) = 1/\Delta$ (Figure 2).

Also, each coefficient is labeled either as IDL or non-IDL depending on its value x and index i. Samples labeled as IDL are quantized in a different way which reduces the total embedding distortion. Both types of coefficients (IDL and non-IDL) are being modified during quantization. However, after quantization, interpretation of a bit of each IDL coefficient is incorrect. The purpose of quantization is to provide that all the non-IDL samples can be extracted correctly and the resulting distribution of all the samples is the one depicted in Figure 1(a). Parameters η_1 and ϑ_0 represent fractions of IDL for $b = 1$ and $b = 0$, respectively. Parameters φ_1 and γ_0 represent fractions of non-IDL samples for $b = 1$ and $b = 0$, respectively. The fraction of zeros in a watermark data is $\gamma_0 + \vartheta_0$ and fraction of ones is $\varphi_1 + \eta_1$. It is required that $\gamma_0 + \vartheta_0 + \varphi_1 + \eta_1 = 1$.

We define IDL and non-IDL samples using two-place predicates IDL_0, IDL_1, NIDL_0, and NIDL_1 in the following way (Figure 2):

$$\begin{aligned}
&\left(\mathrm{IDL}_0 ix \equiv \left(X_0 ix \& (x > L_1)\right)\right), \quad (\forall i)(\forall x),\\
&\left(\mathrm{IDL}_1 ix \equiv \left(X_1 ix \& (x < L_2)\right)\right), \quad (\forall i)(\forall x),\\
&\left(\mathrm{NIDL}_0 ix \equiv \left(X_0 ix \& (x \le L_1)\right)\right), \quad (\forall i)(\forall x),\\
&\left(\mathrm{NIDL}_1 ix \equiv \left(X_1 ix \& (x \ge L_2)\right)\right), \quad (\forall i)(\forall x),
\end{aligned} \tag{9}$$

where $L_1 = \Delta\gamma_0/(\gamma_0 + \vartheta_0)$, $L_2 = \Delta\eta_1/(\varphi_1 + \eta_1)$, and $L_1 \ge L_2$.

Sets $\mathbf{IDL}_0$, $\mathbf{IDL}_1$, $\mathbf{NIDL}_0$, and $\mathbf{NIDL}_1$ will be used in order to specify all the coefficients that satisfy IDL_0, IDL_1, NIDL_0, and NIDL_1, respectively. Fractions γ_0, ϑ_0, φ_1, and η_1 can be expressed in terms of cardinalities of sets $\mathbf{IDL}_0$, $\mathbf{IDL}_1$, $\mathbf{NIDL}_0$, $\mathbf{NIDL}_1$, and $\mathbf{E}$. For example, $|\mathbf{IDL}_0|/|\mathbf{E}| = \vartheta_0$.

In this paper, two different quantization techniques are proposed. Since predicate logic is used to describe watermark embedding, a suitable logical construction should be able to distinguish between the techniques. According to our model, each kind of quantization can be represented by setting a corresponding logical value ("0" or "1") for zero-place predicate Ω. Hence, Ω is used to define one out of two possible quantization techniques. For each kind of quantization, $\mathbf{E}$ is split on two subsets $\mathbf{E}_0$ and $\mathbf{E}_1$. For two-place predicates E_0 and E_1 formulas $E_0 ix$ and $E_1 ix$ are defined in the following way:

$$\begin{aligned}
&\left(E_0 ix \equiv \left(\mathrm{NIDL}_0 ix \vee \left(\mathrm{IDL}_1 ix \& \Omega\right) \vee \left(\mathrm{IDL}_0 ix \& \sim \Omega\right)\right)\right), \quad (\forall i)(\forall x),\\
&\left(E_1 ix \equiv \left(Eix \& \sim E_0 ix\right)\right), \quad (\forall i)(\forall x).
\end{aligned} \tag{10, 11}$$

Using information about distribution inside $\mathbf{IDL}_0$, $\mathbf{IDL}_1$, $\mathbf{NIDL}_0$, and $\mathbf{NIDL}_1$ it is easy to derive distribution inside $\mathbf{E}_0$ and $\mathbf{E}_1$. Let us introduce variable $\omega \in \{0, 1\}$ of natural numbers domain $\mathbb{N}$ (not a logical variable) which satisfies $(\Omega \supset (\omega = 1)) \& (\sim \Omega \supset (\omega = 0))$. Common arithmetical operations can be performed with ω which makes it possible to express PDF $f_{\mathbf{E}_0}(x)$ in the following compact form:

$$f_{\mathbf{E}_0}(x) = \begin{cases} \dfrac{(\gamma_0 + \vartheta_0) f_{\mathbf{X}_0}(x) + \omega(\varphi_1 + \eta_1) f_{\mathbf{X}_1}(x)}{DN_0}, & \text{if } x \le L_2,\\ \dfrac{(\gamma_0 + \vartheta_0) f_{\mathbf{X}_0}(x)}{DN_0}, & \text{if } L_2 < x \le L_1,\\ \dfrac{(1-\omega)(\gamma_0 + \vartheta_0) f_{\mathbf{X}_0}(x)}{DN_0}, & \text{otherwise,} \end{cases} \tag{12}$$

where $DN_0 = (\omega\eta_1 + \gamma_0 + (1 - \omega)\vartheta_0)$.

Therefore $f_{\mathbf{E}_1}(x)$ can be expressed as (Figures 3 and 4)

$$f_{\mathbf{E}_1}(x) = \frac{f_{\mathbf{E}}(x) - DN_0 f_{\mathbf{E}_0}(x)}{1 - DN_0}. \tag{13}$$

Elements of sets $\mathbf{E}_0$ and $\mathbf{E}_1$ are modified during quantization so that new sets $\mathbf{E}_0'$ and $\mathbf{E}_1'$ are obtained, respectively.

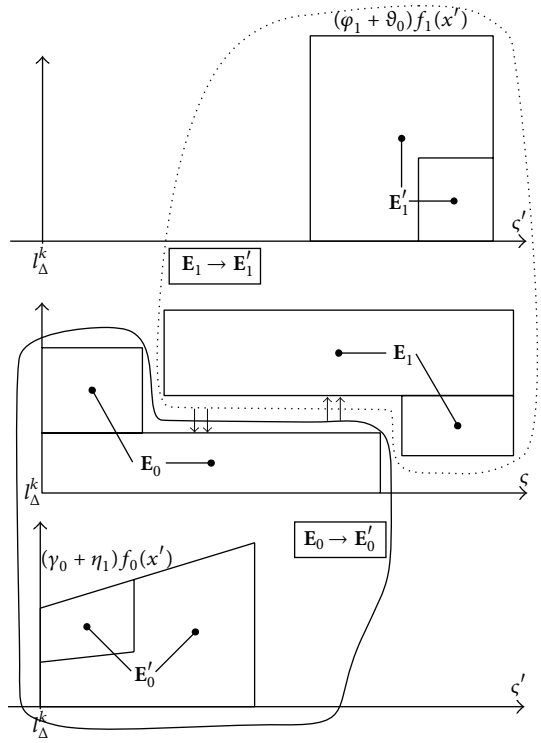


Figure 3: Scheme of redistribution of original samples during quantization, Ω is "true."

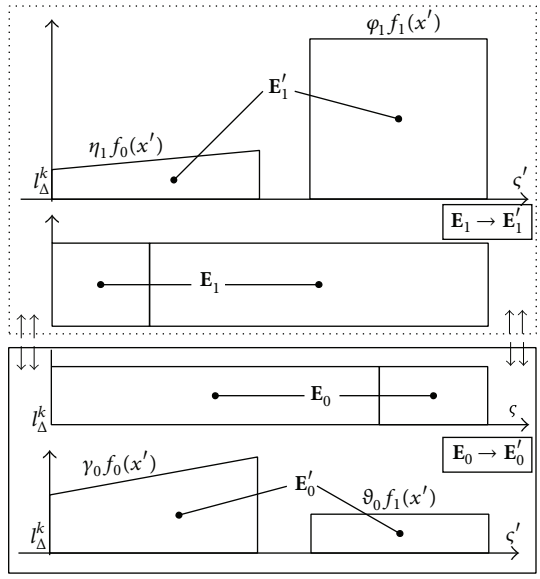


Figure 4: Scheme of redistribution of original samples during quantization, Ω is "false."

Therefore, for successful quantization, we require the following formula $F1$ to be true:

$$F1 \equiv \left(\left(E_0 i x \supset E_0' i x'\right) \& \left(E_1 i x \supset E_1' i x'\right)\right), \quad (\forall i)\,(\forall x)\left(\exists x'\right). \tag{14}$$

As a result of quantization, variables $X_{\mathbf{E}_0}$ and $X_{\mathbf{E}_1}$ are modified in a way that the resulting $X'_{\mathbf{E}_0'}$ and $X'_{\mathbf{E}_1'}$ are distributed according to some desired distributions. For each kind of quantization (depending on the value of Ω), the pair of desired distributions is different. We propose the following distributions that can be expressed as (Figures 3 and 4)

$$f_{\mathbf{E}_0'}\left(x'\right) = \omega f_0\left(x'\right) + (1-\omega)\frac{\gamma_0 f_0\left(x'\right) + \vartheta_0 f_1\left(x'\right)}{\gamma_0 + \vartheta_0},$$
$$f_{\mathbf{E}_1'}\left(x'\right) = \omega f_1\left(x'\right) + (1-\omega)\frac{\eta_1 f_0\left(x'\right) + \varphi_1 f_1\left(x'\right)}{\varphi_1 + \eta_1}. \quad (15)$$

It can be seen that, for any logical value of Ω, the distribution of X' inside $\{\mathbf{E}_0 \cup \mathbf{E}_1\}$ is the same and matches the distribution represented in Figure 1. It means that the efficiency of the procedure of GA recovery (proposed in the previous section) cannot be affected by the selection of Ω.

In addition to the necessity of providing desired distribution of the quantized samples, we need to minimize quantization distortions. Both requirements can be expressed by two two-place predicates U and V:

$$\left(E_0' i x' \equiv E_0 i x \& U x x'\right), \quad (\forall i)(\forall x)\left(\forall x'\right),$$
$$\left(E_1' i x' \equiv E_1 i x \& V x x'\right), \quad (\forall i)(\forall x)\left(\forall x'\right). \quad (16)$$

The idea of minimization of embedding distortions can be explained in the following example. Assuming two samples $x_i, x_j \in \mathbf{E}_0$, $x_i \le x_j$, we infer that quantization in a way in which $x_i' \le x_j'$ implies less distortion than in case when $x_i' > x_j'$. Let us sort elements in $\mathbf{E}_0$ and $\mathbf{E}_0'$ in the dimension of x and x', respectively. Then, for some x_i (index i is an order in a host sequence) the number of elements in $\mathbf{E}_0$ with x value less than x_i should be equal to the number of elements in $\mathbf{E}_0'$ that have x' value less than x_i'. Integration should be used in case we switch from discrete distribution of samples in $\mathbf{E}_0$ and $\mathbf{E}_0'$ to continuous one. Further, throughout the paper we assume that the constant of integration is zero for indefinite integrals. Hence, the truth values for both predicates U and V are defined as

$$\left(Uxx' \equiv \left(\int f_{\mathbf{E}_0}(x)\,dx = \int f_{\mathbf{E}_0'}\left(x'\right)dx'\right)\right), \quad (\forall x)\left(\forall x'\right), \quad (17)$$

$$\left(Vxx' \equiv \left(\int f_{\mathbf{E}_1}(x)\,dx = \int f_{\mathbf{E}_1'}\left(x'\right)dx'\right)\right), \quad (\forall x)\left(\forall x'\right). \quad (18)$$

Further, we introduce logical formula $F2$

$$F2 \equiv \left(\left(\exists x'\right)Uxx' \& \left(\exists x'\right)Vxx'\right), \quad (\forall x) \quad (19)$$

and state that argument

$$F2, (11), (16) \vDash F1 \quad (20)$$

is valid. The task of watermark embedding is to assure that the mentioned argument is sound. For that purpose, a procedure that makes $F2$ true should be proposed.

3.2. Quantization Equations. Quantization equations and their solutions are needed to satisfy formula $F2$ during embedding. For this purpose, we will analyze conditions that enforce qualities of predicates U and V. Due to the large number of variables in the text we recommend to refer to Nomenclature section for clarity. We can rewrite elements of (17) in the following way:

$$\int f_{\mathbf{E}_0}(x)\,dx = \begin{cases} \dfrac{\min(x, L_2)\,\omega(\varphi_1 + \eta_1) + x(\gamma_0 + \vartheta_0)}{\Delta D N_0}, & \text{if } x \le L_1; \\ \omega + \dfrac{(1-\omega)\,x(\gamma_0 + \vartheta_0)}{\Delta D N_0}, & \text{otherwise,} \end{cases}$$
$$\int f_{\mathbf{E}_0'}\left(x'\right)dx' = \begin{cases} \left(\omega + \gamma_0\dfrac{1-\omega}{\gamma_0 + \vartheta_0}\right)\int f_0\left(x'\right)dx', & \text{if } x' \le \Delta\beta; \\ \left(\omega + \gamma_0\dfrac{1-\omega}{\gamma_0 + \vartheta_0}\right) + \vartheta_0\dfrac{1-\omega}{\gamma_0 + \vartheta_0}\left(\int f_1\left(x'\right)dx' + \displaystyle\int_{\Delta\beta}^{0} f_1\left(x'\right)dx'\right), & \text{otherwise.} \end{cases} \quad (21)$$

From (21) it is clear that

$$\int_0^{L_1} f_{\mathbf{E}_0}(x)\,dx = \int_0^{\Delta(\beta-\alpha)} f_{\mathbf{E}_0'}\left(x'\right)dx' = \omega + \gamma_0\frac{1-\omega}{\gamma_0 + \vartheta_0}. \quad (22)$$

The equation above means that the following is true:

$$\left(Uxx' \supset \left(\left((x \le L_1) \& \left(x' \le \Delta\beta\right)\right) \vee \left((x > L_1) \& \left(x' > \Delta\beta\right)\right)\right)\right), \quad (\forall x)\left(\forall x'\right). \quad (23)$$

We introduce two two-place predicates U^1 and U^2:

$$\left(\left(\left(Uxx' \& (x \le L_1) \& \left(x' \le \Delta\beta\right)\right) \equiv U^1 x x'\right) \& \left(\left(Uxx' \& (x > L_1) \& \left(x' > \Delta\beta\right)\right) \equiv U^2 x x'\right)\right), \quad (\forall x)\left(\forall x'\right). \quad (24)$$

According to (21) and (24) the following can be derived:

$$\left(U^1 x x' \equiv \left(\Upsilon_1(x, \omega, \gamma_0, \vartheta_0, \varphi_1, \eta_1) = 0.5 c x'^2 + \tau x'\right)\right),$$

$$\begin{aligned}
&(\forall x)\left(\forall x'\right),\\
&\left(U^2xx' \equiv \left(\Upsilon_2\left(x,\omega,\gamma_0,\vartheta_0,\varphi_1,\eta_1\right) = g\left(x'-\Delta\beta\right)\right)\right),\\
&(\forall x)\left(\forall x'\right),
\end{aligned} \tag{25}$$

where

$$\begin{aligned}
&\Upsilon_1\left(x,\omega,\gamma_0,\vartheta_0,\varphi_1,\eta_1\right)\\
&\quad = \left(\gamma_0+\vartheta_0\right)\frac{\min\left(x,L_2\right)\omega\left(\varphi_1+\eta_1\right)+x\left(\gamma_0+\vartheta_0\right)}{\Delta DN_0\left(\gamma_0+\omega\vartheta_0\right)},\\
&\Upsilon_2\left(x,\omega,\gamma_0,\vartheta_0,\varphi_1,\eta_1\right) = \frac{x\left(\gamma_0+\vartheta_0\right)^2-\gamma_0\Delta DN_0}{\vartheta_0\Delta DN_0}.
\end{aligned} \tag{26}$$

Now, let us analyze conditions that enforce quality of predicate V. Elements of (18) can be represented as

$$\begin{aligned}
\int f_{\mathbf{E}_1}(x)\,dx &= \begin{cases}\dfrac{(1-\omega)\,x\left(\varphi_1+\eta_1\right)}{\Delta\left(1-DN_0\right)}, & \text{if } x\le L_2;\\ \dfrac{\max\left(x-L_1,0\right)\omega\left(\gamma_0+\vartheta_0\right)+\left(x-L_2\right)\left(\varphi_1+\eta_1\right)}{\Delta\left(1-DN_0\right)}, & \text{otherwise,}\end{cases}\\
\int f_{\mathbf{E}_1'}\left(x'\right)dx' &= \begin{cases}\dfrac{(1-\omega)\,\eta_1}{\varphi_1+\eta_1}\displaystyle\int f_0\left(x'\right)dx', & \text{if } x'\le\Delta\beta;\\ \dfrac{(1-\omega)\,\eta_1}{\varphi_1+\eta_1}+\left(\omega+\varphi_1\dfrac{1-\omega}{\varphi_1+\eta_1}\right)\left(\displaystyle\int f_1\left(x'\right)dx'+\int_{\Delta\beta}^{0} f_1\left(x'\right)dx'\right), & \text{otherwise.}\end{cases}
\end{aligned} \tag{27}$$

We can see that according to (25)

$$\int_0^{L_2} f_{\mathbf{E}_1}(x)\,dx = \int_0^{\Delta(\beta-\alpha)} f_{\mathbf{E}_1'}\left(x'\right)dx' = \frac{(1-\omega)\,\eta_1}{\varphi_1+\eta_1}. \tag{28}$$

This means that the following expression is true:

$$\begin{aligned}
&\left(Vxx' \supset \left(\left(\left(x\le L_2\right)\&\left(x'\le\Delta\beta\right)\right)\right.\right.\\
&\quad\left.\left.\vee\left(\left(x>L_2\right)\&\left(x'>\Delta\beta\right)\right)\right)\right),\quad (\forall x)\left(\forall x'\right).
\end{aligned} \tag{29}$$

Next, two two-place predicates V^1 and V^2 are defined as

$$\begin{aligned}
&\left(\left(\left(Vxx'\&\left(x\le L_2\right)\&\left(x'\le\Delta\beta\right)\right)\equiv V^1xx'\right)\right.\\
&\quad\left.\&\left(\left(Vxx'\&\left(x>L_2\right)\&\left(x'>\Delta\beta\right)\right)\equiv V^2xx'\right)\right),\\
&(\forall x)\left(\forall x'\right).
\end{aligned} \tag{30}$$

According to (27) and (30) the following can be derived:

$$\begin{aligned}
&\left(V^1xx' \equiv \left(\Upsilon_3\left(x,\omega,\gamma_0,\vartheta_0,\varphi_1,\eta_1\right) = 0.5cx'^2+\tau x'\right)\right),\\
&(\forall x)\left(\forall x'\right),\\
&\left(V^2xx' \equiv \left(\Upsilon_4\left(x,\omega,\gamma_0,\vartheta_0,\varphi_1,\eta_1\right) = g\left(x'-\Delta\beta\right)\right)\right),\\
&(\forall x)\left(\forall x'\right),
\end{aligned} \tag{31}$$

where

$$\begin{aligned}
\Upsilon_3\left(x,\omega,\gamma_0,\vartheta_0,\varphi_1,\eta_1\right) &= \frac{x\left(\varphi_1+\eta_1\right)^2}{\eta_1\Delta\left(1-DN_0\right)},\\
\Upsilon_4\left(x,\omega,\gamma_0,\vartheta_0,\varphi_1,\eta_1\right) &= \frac{\left(\varphi_1+\eta_1\right)\left(\max\left(x-L_1,0\right)\omega\left(\gamma_0+\vartheta_0\right)+\left(x-L_2\right)\left(\varphi_1+\eta_1\right)\right)-\Delta\left(1-DN_0\right)(1-\omega)\,\eta_1}{\Delta\left(1-DN_0\right)\left(\varphi_1+\omega\eta_1\right)}.
\end{aligned} \tag{32}$$

We can express U using U^1 and U^2 in the following way:

$$\begin{aligned}
&\left(Uxx'\right.\\
&\quad\left.\equiv\left(\left(\left(x\le L_1\right)\supset U^1xx'\right)\&\left(\left(x>L_1\right)\supset U^2xx'\right)\right)\right),\\
&(\forall x)\left(\forall x'\right).
\end{aligned} \tag{33}$$

Also, we can express V using V^1 and V^2:

$$\begin{aligned}
&\left(Vxx'\right.\\
&\quad\left.\equiv\left(\left(\left(x\le L_2\right)\supset V^1xx'\right)\&\left(\left(x>L_2\right)\supset V^2xx'\right)\right)\right),\\
&(\forall x)\left(\forall x'\right).
\end{aligned} \tag{34}$$

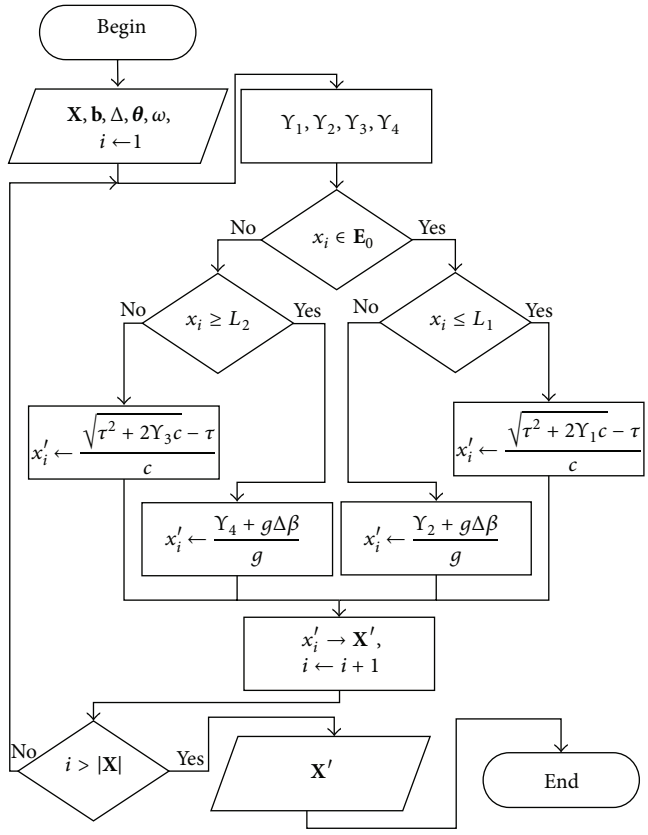


Figure 5: Quantization diagram for the kth embedding interval.

Further, utilizing property $L_2 \le L_1$ we can obtain

$$\begin{aligned}
&\left((x \le L_2) \supset \left(\left(\exists x'\right)\left(U^1 x x'\right) \& \left(\exists x'\right)\left(V^1 x x'\right)\right)\right), \\
&\qquad\qquad (\forall x), \\
&\left(\left(L_2 < x \le L_1\right)\right. \\
&\quad \left. \supset \left(\left(\exists x'\right)\left(U^1 x x'\right) \& \left(\exists x'\right)\left(V^2 x x'\right)\right)\right), (\forall x), \\
&\left(\left(L_1 < x\right) \supset \left(\left(\exists x'\right)\left(U^2 x x'\right) \& \left(\exists x'\right)\left(V^2 x x'\right)\right)\right) \\
&\quad \vDash F2, \quad (\forall x).
\end{aligned} \tag{35}$$

Here, each premise should be true. With the aim to provide this, equations in (25) and (31) should be solvable. It can be seen that the solutions are straightforward:

$$x'_{U^1,V^1} = \frac{\sqrt{\tau^2 + 2\Upsilon_{1,3}c} - \tau}{c}, \tag{36}$$

$$x'_{U^2,V^2} = \frac{\Upsilon_{2,4} + g\Delta\beta}{g}, \tag{37}$$

where, for example, in (36), x'_{U^1,V^1} denotes the values of x' that turn either $U^1 x x'$ or $V^1 x x'$ true for $\Upsilon_1(\cdot)$ or $\Upsilon_3(\cdot)$, respectively. The diagram of quantization is represented in Figure 5. Each ith original sample is chosen from array $\mathbf{X}$ on ith iteration. The corresponding bit of a watermark is chosen from array $\mathbf{b}$. Vector $\boldsymbol{\theta}$ contains parameters of the quantization. At the end of each iteration, quantized value of ith sample is written to array $\mathbf{X}'$.

4. Robustness under AWGN

In this section, we will analytically estimate the robustness of the proposed watermarking scheme under AWGN. Robustness is reflected by the term "extracted information" which denotes mutual information between embedded and detected messages. In contrast to channel capacity, the index of extracted information is practical but depends on the algorithm of detection. Also, throughout this section we assume that the original samples are distributed uniformly inside the quantization interval.

The derivations for extracted information are less involved when Ω is "false." Therefore, only that condition is considered here. In order to estimate extracted information we first find error rates. The rates depend on the attack severity (represented by σ), Δ, and parameter set $\boldsymbol{\theta} = \{\gamma_0, \varphi_1, \eta_1, \vartheta_0, \alpha, \beta\}$. Moreover, we derive a stronger statement that information about Δ/σ and $\boldsymbol{\theta}$ is sufficient to perform analytic estimation of error rates for our watermarking scheme. Finally, we will demonstrate how error rates can be expressed using WNR and $\boldsymbol{\theta}$.

4.1. Estimation of Error Rates. For our estimation, it is considered that, during watermark extraction, in each embedding interval samples that interpret "0" are separated from samples that interpret "1" using a threshold (e.g., hard decision region detector). The position of the threshold in ith embedding interval is $\mathrm{Th} + [\Delta - 2\mathrm{Th}] \bmod (i-k, 2)$ (dashed vertical lines in Figure 1(b)). Therefore, the whole real number line can be seen as a union of two domains:

$$\mathbf{Z} = \bigcup_{m=-\infty}^{\infty} \left[2\Delta m + l_{\Delta}^{k} - \mathrm{Th}, 2\Delta m + l_{\Delta}^{k} + \mathrm{Th}\right), \tag{38}$$

$$\mathbf{O} = \bigcup_{m=-\infty}^{\infty} \left[2\Delta m + l_{\Delta}^{k} + \mathrm{Th}, 2\Delta\left(m+1\right) + l_{\Delta}^{k} - \mathrm{Th}\right). \tag{39}$$

During extraction, all the elements in $\mathbf{Z}$ will be labeled "0" and all the elements in $\mathbf{O}$ will be labeled "1."

After noise is added, elements quantized in kth embedding interval might spread over its limits and other notations should be used. We notate sample values that are affected by noise as ς_n'. Also, ς_n' belongs to some embedding interval and inside this interval we use $x_n' = \varsigma_n' - \Delta\lfloor \varsigma_n'/\Delta \rfloor$. Random variables Σ_n' and X_n' represent ς_n' and x_n', respectively (alternatively we use $\dot{\Sigma}'$ and $\dot{X}'$ to save space in lower subscript part). Therefore, two modified sets are obtained: $\mathbf{E}_0' \xrightarrow{\text{AWGN}} \dot{\mathbf{\Sigma}}_0'$; $\mathbf{E}_1' \xrightarrow{\text{AWGN}} \dot{\mathbf{\Sigma}}_1'$. For noise variance σ^2 we might, for instance, estimate the expected fraction for each of the noisy sets $\dot{\mathbf{\Sigma}}_0'$ and $\dot{\mathbf{\Sigma}}_1'$ in $\mathbf{Z}$. Fractions of $\dot{\mathbf{\Sigma}}_0'$ and $\dot{\mathbf{\Sigma}}_1'$ that belong to $\mathbf{O}$ can be found in a trivial manner. In that way we obtain error rates for "0" and "1."

However, instead of appealing directly to sets $\dot{\mathbf{\Sigma}}_0'$ and $\dot{\mathbf{\Sigma}}_1'$, we use an indirect but computationally lighter approach. In case Ω is "false" we can conclude for the following distributions of quantized samples (not affected by AWGN yet) that

$$f_{\check{\mathbf{E}}_0'}\left(x'\right) = f_{\check{\mathbf{E}}_1'}\left(x'\right) = f_0\left(x'\right), \tag{40}$$

where

$$\begin{aligned} &\left(\check{E}_0' i x' \equiv \left(E_0' i x' \,\&\, \left(x' \le \Delta\left(\beta - \alpha\right)\right)\right)\right), \quad (\forall i)\left(\forall x'\right), \\ &\left(\check{E}_1' i x' \equiv \left(E_1' i x' \,\&\, \left(x' \le \Delta\left(\beta - \alpha\right)\right)\right)\right), \quad (\forall i)\left(\forall x'\right). \end{aligned} \tag{41}$$

Also, we can conclude that the following distributions are also identical:

$$f_{\hat{\mathbf{E}}_0'}\left(x'\right) = f_{\hat{\mathbf{E}}_1'}\left(x'\right) = f_1\left(x'\right), \tag{42}$$

where

$$\begin{aligned} &\left(\hat{E}_0' i x' \equiv \left(E_0' i x' \,\&\, \left(x' \ge \Delta\beta\right)\right)\right), \quad (\forall i)\left(\forall x'\right), \\ &\left(\hat{E}_1' i x' \equiv \left(E_1' i x' \,\&\, \left(x' \ge \Delta\beta\right)\right)\right), \quad (\forall i)\left(\forall x'\right). \end{aligned} \tag{43}$$

For any σ, (40) means that, for example, the fraction of elements from $\check{\mathbf{E}}_0'$ that after AWGN appear in $\mathbf{Z}$ is equal to that of $\check{\mathbf{E}}_1'$ and can be calculated using $f_0(x')$. This fraction will be denoted as $\check{F}_{\mathbf{Z}}$. The PDF of AWGN with variance σ_n^2 is denoted as $f_{\mathcal{N}}[\varsigma_n' - \varsigma', 0, \sigma_n]$ using parameters $\varsigma' = x' + l_{\Delta}^{k}$ and ς_n'. Therefore

$$\begin{aligned} &\check{F}_{\mathbf{Z}} \\ &= \int_{\mathbf{Z}} \int_{0}^{\Delta(\beta-\alpha)} f_0\left(x'\right) f_{\mathcal{N}}\left[\varsigma_n' - x' - l_{\Delta}^{k}, 0, \sigma_n\right] dx' d\varsigma_n'. \end{aligned} \tag{44}$$

Fraction of elements from $\hat{\mathbf{E}}_0'$ that after AWGN appear in $\mathbf{Z}$ will be denoted as $\hat{F}_{\mathbf{Z}}$:

$$\hat{F}_{\mathbf{Z}} = \int_{\mathbf{Z}} \int_{\Delta\beta}^{\Delta} f_1\left(x'\right) f_{\mathcal{N}}\left[\varsigma_n' - x' - l_{\Delta}^{k}, 0, \sigma_n\right] dx' d\varsigma_n'. \tag{45}$$

Error rates are calculated using $\check{F}_{\mathbf{Z}}$ and $\hat{F}_{\mathbf{Z}}$:

$$\begin{aligned} \mathrm{BER}_0 &= \left(1 - \check{F}_{\mathbf{Z}}\right)\frac{\gamma_0}{\gamma_0 + \vartheta_0} + \left(1 - \hat{F}_{\mathbf{Z}}\right)\frac{\vartheta_0}{\gamma_0 + \vartheta_0}, \\ \mathrm{BER}_1 &= \check{F}_{\mathbf{Z}}\frac{\eta_1}{\varphi_1 + \eta_1} + \hat{F}_{\mathbf{Z}}\frac{\varphi_1}{\varphi_1 + \eta_1}. \end{aligned} \tag{46}$$

In order to demonstrate that error rates can be calculated based on Δ/σ, $\boldsymbol{\theta}$ we analyze expression for $\check{F}_{\mathbf{Z}}$ (expression for $\hat{F}_{\mathbf{Z}}$ can be analyzed in a similar way). Function $f_0(x')$ is present in (44). According to (2) it is defined using parameters c, τ. Parameters α, β are also present in (2) as well as in (44). Parameters α, β have clear constraints (the same is true about $\gamma_0, \vartheta_0, \varphi_1, \eta_1$). It is possible to express c, τ using $\alpha, \beta, \gamma_0, \vartheta_0, \varphi_1, \eta_1$. In the realization of our method parameter τ is set as

$$\tau = \frac{\gamma_0 + \vartheta_0}{\Delta\gamma_0}. \tag{47}$$

Defining new parameter $\acute{\tau}$ as

$$\acute{\tau} = \tau\Delta, \tag{48}$$

it can be seen that $\acute{\tau} = (\gamma_0 + \vartheta_0)/\gamma_0$ does not depend on the choice of Δ.

Using property of PDF, the following is obtained from (2):

$$\begin{aligned} \int_{0}^{(\beta-\alpha)\Delta} f_0\left(x'\right) dx' &= c\frac{\left(\beta-\alpha\right)^2\Delta^2}{2} + \tau\Delta\left(\beta-\alpha\right) \\ &= 1. \end{aligned} \tag{49}$$

It is easy to derive from (48) and (49) that

$$c\Delta^2 = 2\frac{1 - \acute{\tau}\left(\beta-\alpha\right)}{\left(\beta-\alpha\right)^2}. \tag{50}$$

According to (50), it is also obvious that parameter

$$\acute{c} = c\Delta^2 \tag{51}$$

is independent of Δ.

One of the properties of PDF of AWGN is

$$f_{\mathcal{N}}[x,0,\sigma_n] = \frac{1}{\sigma_n} f_{\mathcal{N}}\left[\frac{x}{\sigma_n},0,1\right]. \tag{52}$$

Therefore, we can rewrite (44) in the following manner:

$$\begin{aligned}\check{F}_{\mathbf{Z}} &= \int_{\mathbf{Z}}\int_0^{\Delta(\beta-\alpha)} f_0\left(x'\right) f_{\mathcal{N}}\left[\varsigma_n' - x' - l_\Delta^k, 0,\right.\\ &\left.\sigma_n\right] dx' d\varsigma_n' = \frac{\Delta^2}{\sigma_n}\int_{(\mathbf{Z}-l_\Delta^k)/\Delta}\int_0^{(\beta-\alpha)} f_0\left(x'\right)\\ &\cdot f_{\mathcal{N}}\left[\Delta\frac{\left(\varsigma_n' - l_\Delta^k\right) - x'}{\Delta\sigma_n},\right.\\ &\left.0,1\right] d\left\{\frac{x'}{\Delta}\right\} d\left\{\frac{\left(\varsigma_n' - l_\Delta^k\right)}{\Delta}\right\}.\end{aligned} \tag{53}$$

Now, it can be demonstrated that domain

$$\acute{\mathbf{Z}} = \frac{\left(\mathbf{Z} - l_\Delta^k\right)}{\Delta} = \bigcup_{m=-\infty}^{\infty}\left[2m - \frac{\mathrm{Th}}{\Delta}, 2m + \frac{\mathrm{Th}}{\Delta}\right) \tag{54}$$

is independent of Δ if during extraction parameter

$$\acute{\mathrm{Th}} = \frac{\mathrm{Th}}{\Delta} \tag{55}$$

can be set without information about Δ (e.g., $\acute{\mathrm{Th}}$ may be set as $\acute{\mathrm{Th}} = \beta - 0.5\alpha$). Hence, (53) can be represented in the following way:

$$\begin{aligned}\check{F}_{\mathbf{Z}} &= \int_{\acute{\mathbf{Z}}}\int_0^{(\beta-\alpha)} \frac{\Delta}{\sigma_n}\left(\acute{c}\frac{x'}{\Delta} + \acute{\tau}\right) f_{\mathcal{N}}\left[\frac{\Delta}{\sigma_n}\frac{\left(\varsigma_n' - l_\Delta^k\right) - x'}{\Delta},\right.\\ &\left.0,1\right] d\left\{\frac{x'}{\Delta}\right\} d\left\{\frac{\left(\varsigma_n' - l_\Delta^k\right)}{\Delta}\right\}.\end{aligned} \tag{56}$$

Here, for integration we consider $(\varsigma_n' - l_\Delta^k)/\Delta \in \acute{\mathbf{Z}}$ and $x'/\Delta \in [0,(\beta-\alpha)]$, where both domains $\acute{\mathbf{Z}}$ and $[0,(\beta-\alpha)]$ depend only on β,α. Except the terms $(\varsigma_n' - l_\Delta^k)/\Delta$ and x'/Δ, only $\Delta/\sigma_n, \acute{c}, \acute{\tau}$ appear under the integrals. Therefore, the result of integration, $\check{F}_{\mathbf{Z}}$, depends only on $\Delta/\sigma_n, \boldsymbol{\theta}$.

Further we will express Δ/σ_n in terms of WNR and $\boldsymbol{\theta}$ which confirms that BER_0 and BER_1 can be defined using only WNR and $\boldsymbol{\theta}$.

4.2. Estimation of Δ/σ_n. Measure WNR is widely used in watermarking. It expresses relation between watermark and noise energies and in AWGN case is

$$\mathrm{WNR} = 10\log_{10}\left(\frac{D}{\sigma^2}\right), \tag{57}$$

where D is the energy of the watermark. Plot of robustness index in respect to WNR is one of the characteristics that are the most meaningful for practical implementation [4, 11]. Therefore it is important to be able to express error rates using WNR and the set of embedding parameters $\boldsymbol{\theta}$. For this purpose, we first express Δ/σ_n using WNR and $\boldsymbol{\theta}$.

Parameter D in (57) can be seen as a distortion of a host signal, caused by the quantization. There are many different approaches that adequately assess quality degradation for digital images [19, 20]. Nevertheless, in this paper we are using simple and well-known distortion measure based on MSE between original and quantized samples [21]. We will define D and factor it in a form $\Delta^2 Q$, where Q depends only on $\boldsymbol{\theta}$. It is assumed that original samples are distributed uniformly inside embedding interval. Distortion D is needed to obtain quantized sets $\check{\mathbf{E}}_0'$, $\check{\mathbf{E}}_1'$, $\widehat{\mathbf{E}}_0'$, and $\widehat{\mathbf{E}}_1'$. Therefore, further we will consider D as a sum of four kinds of distortion: $D = \check{D}_0 + \check{D}_1 + \widehat{D}_0 + \widehat{D}_1$. Each of the distortion components is defined as follows:

$$\begin{aligned}\check{D}_0 &= \gamma_0\int_0^{\Delta(\beta-\alpha)} f_0\left(x'\right)\left(x' - \frac{\Delta\gamma_0}{\gamma_0+\vartheta_0}\right.\\ &\left.\cdot\int_0^{x'} f_0\left(x'\right)dx'\right)^2 dx',\\ \check{D}_1 &= \eta_1\int_0^{\Delta(\beta-\alpha)} f_0\left(x'\right)\left(x' - \frac{\Delta\eta_1}{\eta_1+\varphi_1}\right.\\ &\left.\cdot\int_0^{x'} f_0\left(x'\right)dx'\right)^2 dx',\\ \widehat{D}_0 &= \vartheta_0\int_{\beta\Delta}^{\Delta} f_1\left(x'\right)\left(x' - \left(\frac{\Delta\vartheta_0}{\gamma_0+\vartheta_0}\int_{\beta\Delta}^{x'} f_1\left(x'\right)dx'\right.\right.\\ &\left.\left.+\frac{\Delta\gamma_0}{\gamma_0+\vartheta_0}\right)\right)^2 dx',\\ \widehat{D}_1 &= \varphi_1\int_{\beta\Delta}^{\Delta} f_1\left(x'\right)\left(x' - \left(\frac{\Delta\varphi_1}{\eta_1+\varphi_1}\int_{\beta\Delta}^{x'} f_1\left(x'\right)dx'\right.\right.\\ &\left.\left.+\frac{\Delta\eta_1}{\eta_1+\varphi_1}\right)\right)^2 dx'.\end{aligned} \tag{58}$$

It can be demonstrated that each of the distortion components can be factored using Δ^2. For instance, considering $\check{D}_0$ the next result can be obtained:

$$\begin{aligned}\check{D}_0 &= \Delta^2\gamma_0\int_0^{(\beta-\alpha)}\left(\acute{c}\frac{x'}{\Delta} + \acute{\tau}\right)\left(\frac{x'}{\Delta}\right.\\ &\left.-\check{\rho}_0\int_0^{x'/\Delta}\left(\acute{c}\frac{x'}{\Delta} + \acute{\tau}\right)d\left\{\frac{x'}{\Delta}\right\}\right)^2 d\left\{\frac{x'}{\Delta}\right\}\\ &= \Delta^2\check{Q}_0,\end{aligned} \tag{59}$$

where

$$\check{Q}_0 = \gamma_0 (\beta-\alpha)^3 \left(\frac{\acute{c}^3 \check{\rho}_0^2}{24} (\beta-\alpha)^3 + \acute{c}^2 \check{\rho}_0 \frac{5\acute{\tau}\check{\rho}_0 - 4}{20} (\beta-\alpha)^2 + \frac{\acute{c}(1-\acute{\tau}\check{\rho}_0)(1-2\acute{\tau}\check{\rho}_0)}{4} (\beta-\alpha) + \frac{\acute{\tau}(1-\acute{\tau}\check{\rho}_0)^2}{3} \right). \quad (60)$$

Here

$$\check{\rho}_0 = \frac{\gamma_0}{\gamma_0 + \vartheta_0}. \quad (61)$$

The rest of the distortion components can also be factored in a similar way, where

$$\begin{aligned}
\check{Q}_1 &= \eta_1 (\beta-\alpha)^3 \left(\frac{\acute{c}^3 \check{\rho}_1^2}{24} (\beta-\alpha)^3 + \acute{c}^2 \check{\rho}_1 \frac{5\acute{\tau}\check{\rho}_1 - 4}{20} (\beta-\alpha)^2 + \frac{\acute{c}(1-\acute{\tau}\check{\rho}_1)(1-2\acute{\tau}\check{\rho}_1)}{4} (\beta-\alpha) + \frac{\acute{\tau}(1-\acute{\tau}\check{\rho}_1)^2}{3} \right), \\
\widehat{Q}_0 &= \vartheta_0 \acute{g}(1-\beta) \left(\frac{(1-\widehat{\rho}_0\acute{g})^2}{3} \left(1+\beta+\beta^2\right) + (1-\widehat{\rho}_0\acute{g})(\widehat{\rho}_0\acute{g}\beta - \check{\rho}_0)(1+\beta) + (\widehat{\rho}_0\acute{g}\beta - \check{\rho}_0)^2 \right), \\
\widehat{Q}_1 &= \varphi_1 \acute{g}(1-\beta) \left(\frac{(1-\widehat{\rho}_1\acute{g})^2}{3} \left(1+\beta+\beta^2\right) + (1-\widehat{\rho}_1\acute{g})(\widehat{\rho}_1\acute{g}\beta - \check{\rho}_1)(1+\beta) + (\widehat{\rho}_1\acute{g}\beta - \check{\rho}_1)^2 \right), \\
\check{\rho}_1 &= \frac{\eta_1}{\eta_1 + \varphi_1}, \\
\widehat{\rho}_0 &= 1 - \check{\rho}_0 = \frac{\vartheta_0}{\gamma_0 + \vartheta_0}, \\
\widehat{\rho}_1 &= 1 - \check{\rho}_1 = \frac{\varphi_1}{\eta_1 + \varphi_1}.
\end{aligned} \quad (62)$$

Factorization in the form $D = \Delta^2 Q$ can be done based on $Q = \check{Q}_0 + \check{Q}_1 + \widehat{Q}_0 + \widehat{Q}_1$. Therefore, according to (57) Δ/σ can be expressed in the following way:

$$\frac{\Delta}{\sigma} = \sqrt{\frac{10^{0.1*\mathrm{WNR}}}{\check{Q}_0 + \check{Q}_1 + \widehat{Q}_0 + \widehat{Q}_1}}. \quad (63)$$

5. Experimental Results

In this section, two different settings are considered for experiments. AWGN attack is investigated assuming the first kind of settings and GA attack is investigated assuming the second kind of settings. For the first type, the obtained results are compared with the results of QIM and DC-QIM. For the second type, the performance is compared with the results of RDM (DC-QIM was not considered here as it is vulnerable to GA). Here, in each type of experiment, the goal is to estimate the highest possible amount of extracted information of the method for a given intensity of attack. We explore optimization of embedding parameters. During watermark embedding, parameters $\gamma_0, \vartheta_0, \varphi_1, \eta_1$, and Ω define sets $\mathbf{E}_0$ and $\mathbf{E}_1$. In addition to the mentioned parameters, c, τ, and g are needed to define $\mathbf{E}_0'$ and $\mathbf{E}_1'$. Extracted information is maximized over $\gamma_0, \vartheta_0, \varphi_1, \eta_1, \Omega, c, \tau$, and g by brute force approach. With the aim to reduce computations, parameters $\eta_1, \vartheta_0, \varphi_1$, and γ_0 are constrained according to our considerations.

5.1. Information Extracted under AWGN. During the experiment, parameters $\eta_1, \gamma_0, \vartheta_0$, and φ_1 were constrained as $\eta_1 + \gamma_0 = 0.5$, $\vartheta_0 + \varphi_1 = 0.5$. It can be explained by our intention to use a detector based on median thresholding inside embedding interval. Therefore, if one detects the watermark message right after embedding, the IDL fraction for "0" and "1" will be ϑ_0 and η_1, respectively. One of the advantages of median thresholding is that no additional information is needed for detection even though the distribution of quantized samples is asymmetric and controlled by many parameters. The parameter Th for median-based (hard decision) detector is calculated using two steps: (a) for all ς_n' find x_n' according to (5); (b) calculate $\mathrm{Th} = \mathrm{median}(X_n')$.

For other methods that were used for comparisons, the standard hard decision detector was used (with equal length of decision intervals for "0" and "1").

For the proposed watermarking method IDL occurs only if the condition $\gamma_0 + \varphi_1 < 1$ holds. In that case, different values of ω cause different robustness characteristics. However, IDL might not be suitable for some application in Digital Watermarking. For instance, in semifragile watermarking a fraction of lost data can be interpreted as the presence of an attack (which increases false positive rate). Therefore, condition $\gamma_0 + \varphi_1 = 1$ has been investigated first.

In Figure 6, amount of extracted information toward WNR is plotted for the proposed method, DC-QIM and QIM [8]. During watermark extraction, the value of the normalized threshold was set to $\acute{\mathrm{T}}\mathrm{h} = \beta - 0.5\alpha$.

Error rates were calculated according to (46). However, only the integers from $[-100, 100]$ were used as a set for m in (38) instead of the whole set $\mathbb{Z}$. The purpose of the limitation is to reduce computational complexity while still maintaining high fidelity of the result. Then, the maximized amount of extracted information C was calculated according to

$$C = \max_{p_{\mathrm{em}}(\sim b)} \left[p(\sim b, b) \log_2 \left(\frac{p(\sim b, b)}{p_{\mathrm{em}}(\sim b)\, p_{\mathrm{ex}}(b)} \right) + p(b, \sim b) \log_2 \left(\frac{p(b, \sim b)}{p_{\mathrm{em}}(b)\, p_{\mathrm{ex}}(\sim b)} \right) \right.$$

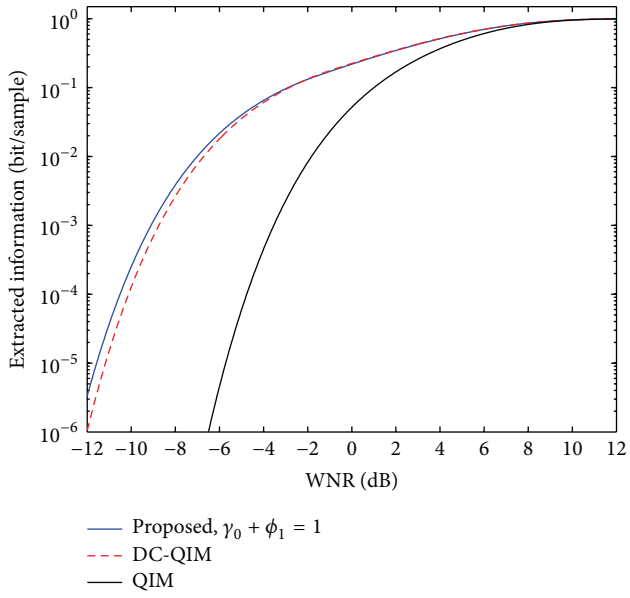


FIGURE 6: Analytic-based estimation of information extracted under AWGN without IDL.

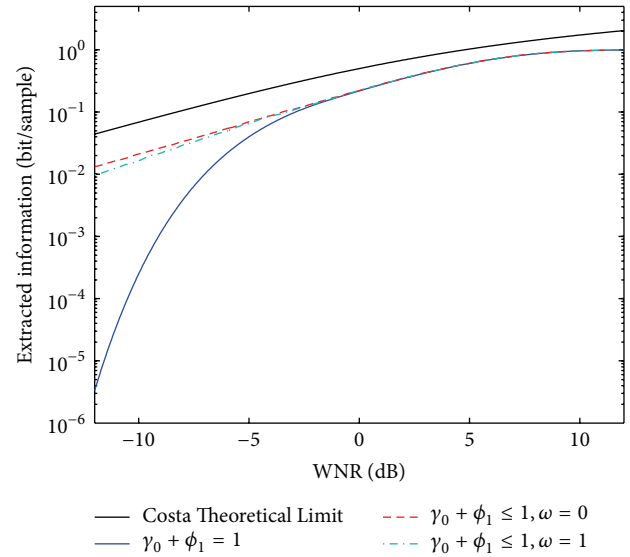


FIGURE 7: Analytic-based estimation of information extracted under AWGN with IDL.

$$+ p(\sim b, \sim b) \log_2 \left(\frac{p(\sim b, \sim b)}{p_{\text{em}}(\sim b)\, p_{\text{ex}}(\sim b)} \right) + p(b, b) \log_2 \left(\frac{p(b, b)}{p_{\text{em}}(b)\, p_{\text{ex}}(b)} \right) \Bigg]. \tag{64}$$

Here, $p(\sim b, b)$ denotes joint probability of embedding symbol $\sim b$ and extracting symbol b; $p_{\text{em}}(b)$ and $p_{\text{ex}}(b)$ denote probabilities of symbol b to be embedded and extracted, respectively. Using joint probabilities, we calculate probabilities of extracting a particular bit:

$$\begin{aligned} p_{\text{ex}}(b) &= p(\sim b, b) + p(b, b), \\ p_{\text{ex}}(\sim b) &= p(b, \sim b) + p(\sim b, \sim b). \end{aligned} \tag{65}$$

Joint probabilities can be expressed using $p_{\text{em}}(\cdot)$ and error rates:

$$\begin{aligned} p(\sim b, b) &= p_{\text{em}}(\sim b)\, \text{BER}_0, \\ p(b, \sim b) &= p_{\text{em}}(b)\, \text{BER}_1, \\ p(\sim b, \sim b) &= p_{\text{em}}(\sim b)\, (1 - \text{BER}_0), \\ p(b, b) &= p_{\text{em}}(b)\, (1 - \text{BER}_1). \end{aligned} \tag{66}$$

As it was mentioned earlier, embedding probabilities are

$$\begin{aligned} p_{\text{em}}(\sim b) &= \gamma_0 + \vartheta_0, \\ p_{\text{em}}(b) &= \eta_1 + \varphi_1. \end{aligned} \tag{67}$$

From Figure 6 it can be seen that with no IDL the proposed method performs better than QIM for WNR $\le$ 9 dB and better than DC-QIM for WNR values less than -2 dB.

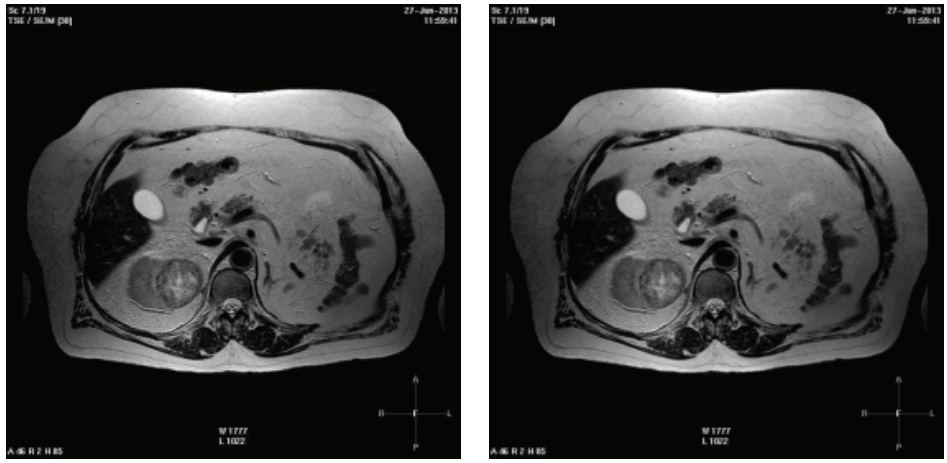
While comparing the designed method with DC-QIM, explanation of a slightly better performance should be found in the new distribution of quantized samples and thresholding technique. Optimal solution for the problem of informed data hiding has been theorized by Costa [6], and every known practical answer, including DC-QIM, uses structured codebook which lacks certain desirable characteristics [11]. We believe that the minor advantage of the presented method is due to larger number of variables that were adjusted for embedding.

In case IDL is acceptable (for a particular watermarking application), much better results are achievable for both "true" and "false" Ω under low WNRs (Figure 7). Obviously, the demonstrated superiority is due to IDL only. Additionally, it can be seen that "false" logical value of Ω provides slightly more beneficial outcome under AWGN compared to when Ω is "true." As a reference, Costa Theoretical Limit (CTL) [6] is plotted in Figure 7:

$$\text{CTL} = \frac{1}{2} \log_2 \left(1 + 10^{0.1 * \text{WNR}}\right). \tag{68}$$

5.2. Information Extracted under GA. In this subsection, we explore performance of the proposed quantization approach under GA. For comparison, RDM is chosen instead of DC-QIM as it is known to be vulnerable to GA. We describe conditions and the results of the simulations based on real images and assuming watermark embedding followed by GA.

For the experimental evaluation, we used 92 natural grayscale images with resolution 512×512. Each image was split on 4×4 blocks and first singular values of Singular Value Decomposition (SVD) were quantized to embed a watermark [22]. The watermarking was arranged without IDL and



(a) (b)

Figure 8: Common MRI imaging of a patient's kidney: (a) original diagnostic image; (b) watermarked image.

$\gamma_0 = \varphi_1 = 0.5$. Document to Watermark Ratio (DWR) was set to 28 dB, where

$$\mathrm{DWR} = 10\log_{10}\left(\frac{\sigma_H^2}{D}\right) \tag{69}$$

and σ_H^2 is the variance of the original coefficients. An example of original and watermarked diagnostic images is given in Figure 8.

For some healthcare organizations, protection of personal data is a high priority task. On the other hand, diagnostic data might need to be shared between experts from other organizations. For that purpose, DIW is a suitable tool [23]. However, an important additional constraint is imposed in that case: an expert conclusion (diagnosis) should not be affected by watermarking. According to the judgements of collaborating group of medical imaging experts, the diagnostic statement for the watermarked image in Figure 8(b) is identical to the statement for the original one in Figure 8(a).

For watermark embedding (encoding), a brute force optimization over α and β was repeated for each new value of σ. Obviously, this needs to be done only once as the optimal parameters can be stored. In addition to the concept of optimal parameters, we investigated efficiency of a constrained version of the proposed quantization approach, where α and β were constant and equal to 0.05 and 0.35, respectively. The common sense behind such a modification is that actual σ of AWGN might not be known on practice during watermark embedding (because the attack happens after embedding).

For watermark extraction (hard decision decoding), these two steps are required: (1) apply GA recovery procedure; (2) define threshold. In accordance with the proposed procedure for GA recovery, criterion C_1 was used for the estimation of actual Δ during the experiment. The condition of GA was simulated by ignoring information about Δ value that was used for embedding. Hence, the value was estimated by the procedure of GA recovery. No information except initial guess interval with $\tilde{\Delta}'_{\min} = 0.9\Delta$, $\tilde{\Delta}'_{\max} = 1.1\Delta$ was used for watermark extraction. In contrast to that, RDM does use the information about the exact value of quantization step. For RDM, the value of a given quantized coefficient was calculated using the information about the previous 100 coefficients.

Two types of thresholding are possible and two types of extraction conditions exist. Under condition when α and β are constants, no additional information needs to be transferred to the decoder. However, if α and β are optimized on encoder's side, information about them might need to be sent. This is necessary if thresholding is established in the way that $\acute{\mathrm{T}}\mathrm{h} = \beta - 0.5\alpha$ (e.g., the threshold is in the middle of separating zone in quantization interval). Since the requirement for additional information seems impractical, we proposed median thresholding $\mathrm{Th} = \mathrm{median}(X'_n)$ as well.

For each method that took part in the experiment, the resulting amount of extracted information is plotted toward AWGN variance (Figure 9).

As it can be seen from Figure 9, under both mentioned conditions of embedding, the proposed approach outperforms RDM. The advantage is more evident for larger AWGN variance.

6. Discussion

In the experimental section, robustness of the proposed quantization method was estimated under AWGN and GA. The proposed approach provides higher amount of extracted information compared to the other state-of-the-art reference methods, like DC-QIM and RDM. The reasons of its superiority will be discussed in more detail in this section.

Asymmetric distribution of quantized samples and the proposed procedure for GA recovery is a successful combination that provides robustness under GA. Compared to other estimations of the scaling factor from the literature, the proposed estimation approach is light. For instance, in order to estimate the scaling factor, a model of a host is used in [14] which complicates estimation and reduces

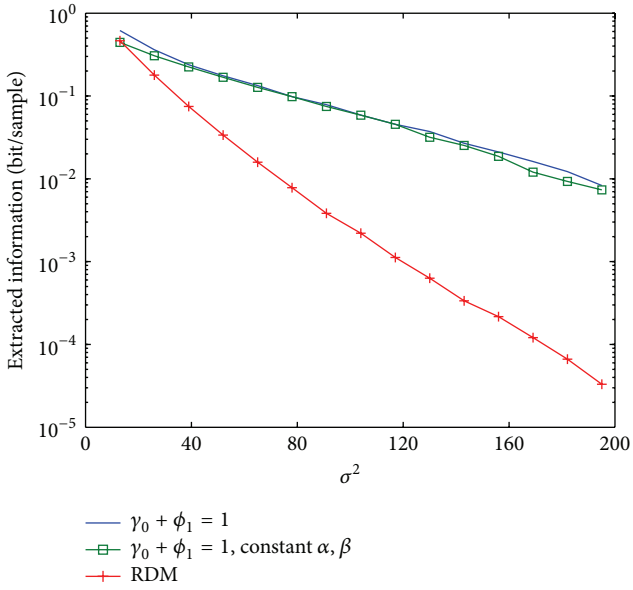


FIGURE 9: Information extracted under GA followed by AWGN.

its precision. On the other hand, an approach different to estimation is exploited by RDM [15]. However, Distortion Compensation is not present in RDM. In contrast to that, the proposed quantization method has Distortion Compensation and outperforms RDM because AWGN is introduced (as a second stage of GA).

The proposed quantization approach demonstrates higher robustness under AWGN compared to well-known DC-QIM. The advancement that causes such superiority is IDL. Parameter Ω was introduced in order to distinguish between two different ways of realization for IDL. Target distribution for all the samples in quantization interval remains the same for any logical value of Ω (which guarantees equally successful recovery from GA). Distributions are different if "ones" and "zeros" are considered separately. This influences the resulting performance. As it has been demonstrated by the experiment, modification of the quantization approach with "false" Ω performs slightly better. In general, usage of IDL is beneficial under low WNRs. The common sense here is that predicting the loss of some information (as a result of an attack) we might accept the scenario when a part of information is lost initially. Compared to DC-QIM, such quantization behavior enables redistribution of embedding distortion from samples that are likely to be misinterpreted to the other (non-IDL) samples that can be more robust.

Unlike DC-QIM, the proposed quantization method has many parameters that need to be set up for watermark embedding. Some additional parameters might be needed depending on the technique for watermark extraction. For instance, the thresholding that depends on $\acute{T}h = \beta - 0.5\alpha$ may be applied, which requires parameters α, β to be communicated to the receiver.

On the other hand, no parameters are needed for extraction if the proposed median thresholding is used (absolutely blind extraction). Parameter Δ can be estimated using procedure for GA recovery taking as input only rough interval $[\tilde{\Delta}'_{\min}, \tilde{\Delta}'_{\max}]$. This is an advantage compared to DC-QIM that always requires Δ to be known to the decoder.

We do not consider any case with malicious attacks (that analyze and deliberately change the watermarked signal) as they are not the objectives of our paper. However in case a key is used to protect a watermark, it will also be needed for decoding.

Lastly, we need to emphasize that the computational cost of our scheme is low. Optimization of embedding parameters conducted in Section 5 is computationally heavy, but it needs to be done only once. The optimized parameters can be used for embedding then. The computational cost of embedding is comparable with that of DC-QIM (please, refer to the quantization diagram). For extraction, complexity of the proposed procedure of GA recovery is $O(n)$.

7. Conclusions

A new scalar QIM-based watermarking method has been proposed in this paper. It provides higher robustness under AWGN and GA compared to other quantization methods. The benefits of the method are due to the introduced procedure of recovery after GA as well as new distribution of quantized samples with IDL.

For the new distribution of quantized samples there is no symmetry inside embedding interval. The nonsymmetric distribution of quantized samples is exploited by the introduced procedure of recovery after GA. Two different criteria are proposed to be used within the procedure. During experiment it has been confirmed that the procedure is computationally light and efficient.

In addition to the new kind of distribution of quantized samples, the proposed QIM-based method benefits from IDL. Utilization of IDL can reduce embedding distortions introduced to a host signal. This is done by letting some watermark bits to be interpreted incorrectly during embedding and before any attack occurs. A model that describes quantization process assumes that IDL can be implemented in two different ways depending on the logical value of parameter Ω. The proposed realization of IDL is beneficial for any Ω under highly intensive AWGN attack. However, "false" value of Ω provides slightly higher robustness compared to "true."

Considerable performance improvements are due to the abovementioned advancements. The amount of information extracted (using hard decision decoder) under AWGN is at the same or of a higher level compared to DC-QIM. Usage of IDL is the most advantageous under AWGN for WNRs close to −12 dB, where it performs up to 10^4 times better than DC-QIM. Under GA, the performance of the proposed method is up to 10^3 times higher than that of RDM. Finally, visual quality degradation caused by the proposed quantization method was also estimated in a subjective way by a group of medical imaging experts. It was confirmed that as a result of watermarking, important diagnostic characteristics did not change.

Nomenclature

i: Unique integer index for each particular sample
Σ: Random variable for the domain of original samples
ς: Particular realization of Σ
Δ: The length of embedding interval
l_Δ^k: The left endpoint of kth embedding interval
X: Random variable for the domain of original samples inside embedding interval
x: Particular realization of X
X': Random variable for the domain of quantized samples inside embedding interval
x': Particular realization of X'
Σ': Random variable for the domain of quantized samples
ς': Particular realization of Σ'
$f_0(x')$, $f_1(x')$: Truncated distributions for quantized samples inside embedding interval
α, β,τ, c, g: Parameters of $f_0(x')$, $f_1(x')$
γ_0, ϑ_0, φ_1, η_1: Fractions of samples that are labeled as non-IDL_0, IDL_0, non-IDL_1, and IDL_1, respectively
$\widetilde{\Delta}$: The length of embedding interval that is required for watermark extraction after GA
$\widetilde{\Delta}'$: Uniformly sampled guessed values of $\widetilde{\Delta}$
$\widetilde{\Delta}''$: Best-fit value from $\{\widetilde{\Delta}'\}$ according to the estimator
Σ_n' (or $\dot{\Sigma}'$): Random variable for the domain of quantized samples affected by attack/noise
ς_n': Particular realization of Σ_n'
X_n' (or $\dot{X}'$): Random variable for the domain of noisy quantized samples inside embedding interval
x_n': Particular realization of X_n'
ω: Parameter of quantization model
$\mathbf{E}$: The set of X
$\mathbf{E}_0$, $\mathbf{E}_1$: Two disjoint subsets of $\mathbf{E}$ defined by $\gamma_0 + \vartheta_0$ and $\varphi_1 + \eta_1$, respectively
$\mathbf{E}'$: The set of X'
$\mathbf{E}_0'$, $\mathbf{E}_1'$: Two disjoint subsets of $\mathbf{E}'$
$X_{\mathbf{E}_0}$: Random variable from $\mathbf{E}_0$
$X_{\mathbf{E}_1}$: Random variable from $\mathbf{E}_1$
$X'_{\mathbf{E}_0'}$: Random variable from $\mathbf{E}_0'$
$X'_{\mathbf{E}_1'}$: Random variable from $\mathbf{E}_1'$
$f_{\mathbf{E}_0}(x)$: PDF of $X_{\mathbf{E}_0}$
$f_{\mathbf{E}_1}(x)$: PDF of $X_{\mathbf{E}_1}$
$f_{\mathbf{E}_0'}(x')$: PDF of $X'_{\mathbf{E}_0'}$
$f_{\mathbf{E}_1'}(x')$: PDF of $X'_{\mathbf{E}_1'}$
Th: The threshold used by the detector
$\acute{\text{Th}}$: The normalized threshold used by the detector, for example, $\acute{\text{Th}} = \text{Th}/\Delta$
$\mathbf{Z}$: Decision region "0" for the detector
$\mathbf{O}$: Decision region "1" for the detector
$\dot{\mathbf{\Sigma}}_0'$: The set of all the elements of $\mathbf{E}_0'$ influenced by an attack
$\dot{\mathbf{\Sigma}}_1'$: The set of all the elements of $\mathbf{E}_1'$ influenced by an attack
$\mathbf{IDL}_0$, $\mathbf{NIDL}_0$: Two disjoint subsets of $\mathbf{E}_0$, defined by ϑ_0 and γ_0, respectively
$\mathbf{IDL}_1$, $\mathbf{NIDL}_1$: Two disjoint subsets of $\mathbf{E}_1$, defined by η_1 and φ_1, respectively
$\check{\mathbf{E}}_0'$, $\hat{\mathbf{E}}_0'$: Two disjoint subsets of $\mathbf{E}_0'$, defined by γ_0 and ϑ_0, respectively
$\check{\mathbf{E}}_1'$, $\hat{\mathbf{E}}_1'$: Two disjoint subsets of $\mathbf{E}_1'$, defined by η_1 and φ_1, respectively
$\check{D}_0$: Embedding distortion necessary to transform $\mathbf{NIDL}_0$ to $\check{\mathbf{E}}_0'$
$\check{D}_1$: Embedding distortion necessary to transform $\mathbf{IDL}_1$ to $\check{\mathbf{E}}_1'$
$\widehat{D}_0$: Embedding distortion necessary to transform $\mathbf{IDL}_0$ to $\hat{\mathbf{E}}_0'$
$\widehat{D}_1$: Embedding distortion necessary to transform $\mathbf{NIDL}_1$ to $\hat{\mathbf{E}}_1'$
BER_0: Bit Error Rate for $\mathbf{E}_0'$
BER_1: Bit Error Rate for $\mathbf{E}_1'$
C: Maximized mutual information between embedded and detected messages.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The paper was supported by University of Tampere. Additional gratitude for provided diagnostic images and postwatermarking quality assessment is to Dr. Oleksandr Iatsyna and other members of collaborating oncological expert group from National Cancer Institute, Kiev, Ukraine.

References

[1] J. Fridrich, *Steganography in Digital Media: Principles, Algorithms, and Applications*, Cambridge University Press, New York, NY, USA, 2009.

[2] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "Robust watermarking of still images for copyright protection," in *Proceedings of 13th International Conference on Digital Signal Processing (DSP '97)*, vol. 2, pp. 499–502, Santorini, Greece, July 1997.

[3] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 430–444, 2006.

[4] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, Morgan Kaufmann Publishers, San Francisco, Calif, USA, 2nd edition, 2007.

[5] T. Bianchi and A. Piva, "Secure watermarking for multimedia content protection: a review of its benefits and open issues," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 87–96, 2013.

[6] M. H. Costa, "Writing on dirty paper," *IEEE Transactions on Information Theory*, vol. 29, no. 3, pp. 439–441, 1983.

[7] B. Chen and G. W. Wornell, "Dither modulation: a new approach to digital watermarking and information embedding," in *Security and Watermarking of Multimedia Contents*, vol. 3657 of *Proceedings of SPIE*, pp. 342–353, San Jose, Calif, USA, April 1999.

[8] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1423–1443, 2001.

[9] E. Esen and A. Alatan, "Forbidden zone data hiding," in *Proceedings of the IEEE International Conference on Image Processing*, pp. 1393–1396, Atlanta, Ga, USA, October 2006.

[10] M. Ramkumar and A. N. Akansu, "Signalling methods for multimedia steganography," *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 1100–1111, 2004.

[11] J. J. Eggers, R. Bäuml, R. Tzschoppe, and B. Girod, "Scalar Costa scheme for information embedding," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1003–1019, 2003.

[12] J. Oostveen, T. Kalker, and M. Staring, "Adaptive quantization watermarking," in *Security, Steganography, and Watermarking of Multimedia Contents VI*, Proceedings of SPIE, pp. 296–303, San Jose, Calif, USA, January 2004.

[13] X. Kang, J. Huang, and W. Zeng, "Improving robustness of quantization-based image watermarking via adaptive receiver," *IEEE Transactions on Multimedia*, vol. 10, no. 6, pp. 953–959, 2008.

[14] I. D. Shterev and R. L. Lagendijk, "Amplitude scale estimation for quantization-based watermarking," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4146–4155, 2006.

[15] F. Pérez-González, C. Mosquera, M. Barni, and A. Abrardo, "Rational dither modulation: a high-rate data-hiding method invariant to gain attacks," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3960–3975, 2005.

[16] F. Ourique, V. Licks, R. Jordan, and F. Pérez-González, "Angle QIM: a novel watermark embedding scheme robust against amplitude scaling distortions," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 2, pp. ii/797–ii/800, IEEE, Philadelphia, Pa, USA, March 2005.

[17] M. Zareian and H. R. Tohidypour, "Robust quantisation index modulation-based approach for image watermarking," *IET Image Processing*, vol. 7, no. 5, pp. 432–441, 2013.

[18] X. Zhu and J. Ding, "Performance analysis and improvement of dither modulation under the composite attacks," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, article 53, 2012.

[19] Q. Sang, X. Wu, C. Li, and Y. Lu, "Universal blind image quality assessment using contourlet transform and singular-value decomposition," *Journal of Electronic Imaging*, vol. 23, no. 6, Article ID 061104, 2014.

[20] S. Chikkerur, V. Sundaram, M. Reisslein, and L. J. Karam, "Objective video quality assessment methods: a classification, review, and performance comparison," *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 165–182, 2011.

[21] M. Petrou and C. Petrou, *Image Processing: The Fundamentals*, John Wiley & Sons, 2010.

[22] Y. Zolotavkin and M. Juhola, "A new blind adaptive watermarking method based on singular value decomposition," in *Proceedings of the International Conference on Sensor Network Security Technology and Privacy Communication System (SNS & PCS '13)*, pp. 184–192, Taipei, Taiwan, March 2013.

[23] W. Pan, G. Coatrieux, N. Cuppens-Boulahia, F. Cuppens, and C. Roux, "Watermarking to enforce medical image access and usage control policy," in *Proceedings of the 6th International Conference on Signal-Image Technology and Internet-Based Systems (SITIS '10)*, pp. 251–260, IEEE, Kuala Lumpur, Malaysia, December 2010.