Hindawi
Security and Communication Networks
Volume 2021, Article ID 6672911, 10 pages
https://doi.org/10.1155/2021/6672911

Research Article

Nowhere to Hide: A Novel Private Protocol Identification Algorithm

Jiantao Shi,[1] **Xiangzhan Yu**,[1] **and Zechao Liu**[2]

[1]*School of Cyberspace Science, Harbin Institute of Technology, Harbin 150001, China*
[2]*School of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China*

Correspondence should be addressed to Jiantao Shi; shijiantao@hit.edu.cn

Received 24 December 2020; Revised 24 January 2021; Accepted 22 February 2021; Published 3 March 2021

Academic Editor: BEN NIU

In recent years, with the rapid development of mobile Internet and 5G technology, great changes have been brought to our lives, and human beings have stepped into the era of big data. These new features and techniques in 5G support many different types of mobile applications for users, which makes network security extremely challenging. Among them, more and more applications involve users' private data, such as location information, financial information, and biological information. In order to prevent users' privacy disclosure, most applications choose to use private protocols. However, such private protocols also provide a means for malware and malicious applications to steal users' privacy and confidential data. From a more secure point of view, we need to provide a way for users to know how many private protocols are running on their mobile phones and distinguish which are authorized applications and which are not. Therefore, the analysis and identification of private protocols have become a hot topic in current research. How to extract the characteristics of network protocol effectively and identify the private protocol accurately becomes the most important part of this research. In this paper, we combine genetic algorithm and association rule algorithm and then propose a set of feature extraction algorithm and protocol recognition algorithm for unknown protocols. The experimental analysis based on the actual data shows that these methods can effectively solve the problems of feature extraction and recognition for unknown protocols and can greatly improve the accuracy of private protocol recognition.

1. Introduction

In recent years, the rapid development of Internet, especially mobile Internet and 5G technology, has brought great and far-reaching impact on our life. The Internet application on mobile phone has become an indispensable part of human life, and it is an important way for people to exchange information and process data. The information may come from individuals, companies, and even governments and may be associated with personal, corporate, or national privacy data. In order to prevent users' privacy disclosure, most applications choose to use private protocol. The privatization of communication protocol makes it difficult for the intermediate nodes on the communication route to obtain the transmitted content directly. In addition, the private protocols also generally use multilayer encryption technology to prevent the user's privacy data from being leaked in the communication process by means of cipher text. It can be seen that the correct use of private protocols is relatively safe for user privacy. Unfortunately, some illegal users and malwares also use private protocols in data transmission. Because of the concealment of these malware and illegal services, users cannot know what they are transmitting. A large number of such private protocols are illegally stealing private data of individual users and enterprise users. Therefore, from the perspective of security, we need to provide a way for users to know how many private protocols are running on their mobile phones and distinguish which are authorized applications and which are not. It is necessary to track and detect the network traffic and private protocol, understand the source and destination of user's personal data, analyze the characteristics of different network information, and effectively help us adjust the network.

The analysis and identification of private protocols have become a hot topic in current research. How to effectively extract the characteristics of network protocol effectively and identify the private protocol accurately becomes the most important part of this research. There has been a lot of research on traditional network traffic identification and protocol identification. The most common method is Deep Packet Inspection (DPI). Such DPI technology needs to obtain the characteristics of the detection content in advance and then detect the message according to the characteristics. For unknown protocols, the current reverse analysis methods and protocol identification methods are usually completed manually. However, with the increasing number of private network protocols and applications, as well as the complexity brought about by the binary design of private network protocols, the method of extracting data features manually is extremely inefficient, and the accuracy of the analysis results is difficult to guarantee. More seriously, for the traditional manual detection methods, there is still a lack of necessary automatic verification scheme to check the accuracy of identification.

Therefore, how to extract the characteristics of private network protocols effectively has become the top priority of research. Protocol identification technology mainly includes the following aspects: fixed-ports-based protocol identification technology [1], traffic-loading-based protocol identification technology [2], machine-learning-based protocol identification technology [3], and format-reverse-analysis-based identification technology [4–6]. The first two technologies are usually used to identify known protocols, the third technology can be used for both known protocols and unknown protocols, and the fourth is used to identify unknown protocols.

This paper systematically analyzes the existing protocol identification and analysis methods, combines genetic algorithm and association rule algorithm, and proposes a set of characteristic extraction and protocol identification algorithms for private protocols. Then, aiming at the protocol, which is difficult to extract effective fixed features, we propose a regular feature extraction algorithm based on genetic programming. It solves the problem of feature extraction and recognition of unknown protocol effectively and improves the accuracy of protocol recognition rate. The main work of this paper is as follows:

(1) This paper first summarizes the relevant methods of protocol identification and protocol reverse analysis, compares the advantages and limitations of various methods, and then introduces related concepts such as information theory, pattern matching, data mining algorithms, and genetic programming.

(2) This paper then proposes a complete feature extraction and protocol recognition algorithm for private protocols. The main processes include data preprocessing, data stream block cutting, multilevel filtering of frequent patterns, periodic feature generation, and feature verification based on association rules. In this paper, the detailed design and experiment of each algorithm are carried out, and the practical significance of extracting features is analyzed based on the actual data, and good experimental results are obtained.

(3) Finally, according to the correlation algorithm of genetic programming and the characteristics of regular expressions, this paper proposes a regular feature extraction algorithm based on tree genetic programming. This method can obtain better regular characteristics by genetic iteration.

2. Related Works

Firstly, the domain of protocol identification classifies protocols based on different ports. However, with the development of the Internet, many new protocols emerge, which often adopt dynamic port numbers. In addition, a large number of network attack traffic and malicious codes deliberately use common ports to avoid traffic detection, which brings about challenges to traditional port-based protocol identification [7, 8]. The accuracy rate of such kind of protocol recognition is getting lower and lower, and the protocol recognition technology based on port is almost obsolete. Subsequently, the load-based traffic recognition technology DPI (Deep Packet Inspection) came into being [9–12]. DPI technology firstly extracts the features according to the target traffic, after obtaining the feature strings of the traffic payload; string-matching method is carried out according to the features. If the corresponding feature strings appear in the network traffic, the specific protocol is identified; otherwise, it is not. The accuracy of protocol identification technology based on load is directly proportional to the accuracy of feature strings. Therefore, the premise of using DIP technology is to obtain the accurate fingerprint information of the protocol. Traditional fingerprint extraction and maintenance often rely heavily on manual labor, so the efficiency is very low. In addition, because DPI technology requires string matching for each data stream, it consumes a lot of space and time. In genetics, researchers use multisequence comparison technology to extract similar fragments in DNA [4], and in protocol reverse engineering, message fields with specific formats are also extracted from a large number of messages. Due to this similarity, researchers often apply multisequence comparison technology to protocol format inference and obtain format information of protocol message by extracting variable and immutable fields in the message [13, 14]. With the development of machine learning, protocol recognition based on machine learning has gradually become an important direction in protocol recognition field. In the field of protocol identification technology, it is necessary to calibrate the traffic data. Because the traffic recognition technology based on machine learning does not need to analyze the load content, it is applied more widely and has a good recognition effect for the encryption protocol. Protocol identification technologies based on machine learning can be divided into two categories. One is protocol identification technology based on supervised learning [15], and this technique requires marked training samples, continuous training of the model, and optimization of the results through iteration to

achieve the highest recognition rate. The other is protocol identification technology for unsupervised learning, which does not require marked training samples [16].

The above algorithms have played an important role in different stages of Internet development, among which port-based methods have been gradually phased out. The load-based and measure-based algorithms are still two hot spots in the future protocol recognition field.

3. Overall Algorithm Architecture

In this paper, we propose a set of feature extraction algorithms for unknown protocols based on the research status of feature extraction for unknown protocols. Figure 1 describes the overall block diagram of the unknown protocol feature extraction and recognition algorithm. It can be seen clearly that the algorithm includes three processes. The first is the data preparation and pretreatment process, which includes data acquisition and division. The data are randomly divided into experimental sets and test sets to prepare data for subsequent feature extraction and feature verification. The second is the feature extraction process, which is the core part of the algorithm, including data segmentation module, feature filter module, feature weight assignment module, and regular feature extraction module. Data segmentation module and feature selection module mainly cut data stream into fixed-length blocks carrying information payload. The *N*-gram algorithm is used for data segmentation, and the threshold value of filtering short frequent patterns is determined by Jaccard coefficient. The threshold value is used to filter the pattern strings with low frequency, and the pattern strings are filtered by the position entropy and information gain of the pattern strings. On the basis of frequent pattern set, the feature weight assignment module uses genetic algorithm to assign weights to each pattern string on the pattern set and further filters the pattern string. Regular feature extraction module generates regular expression features according to association relations by mining multiple association rules between pattern strings under the same category. The last part mainly validates the features obtained from the second stage and mainly adopts the following two methods: the feature validation based on pattern matching and the feature validation based on clustering algorithm. The advantages and disadvantages of feature extraction are judged by these two feature verification methods.

4. Key Algorithms

4.1. Stream Segmentation Algorithm. Combined with the disadvantages of *N*-gram, this paper proposes an improved *N*-gram algorithm using multiple sliding windows (1, 2, . . ., *N*); the position information of gram is recorded at the same time, to facilitate the subsequent feature fusion and association rule analysis between features and finally obtain the candidate feature pattern set with location and frequency (Algorithm 1.

The frequent pattern set with location information is extracted by the above algorithm. Each term in the resulting pattern set is a fixed-length pattern string of length $1 - n$. However, due to the multiple sliding windows adopted by *N*-gram in data segmentation, there must be a certain overlap between the modes. As a result, there is redundancy in the frequent pattern set, so it is necessary to carry out pattern fusion. According to the unknown information of the feature item and its frequency information, the feature item with redundant substring and the feature item whose position difference is equal to the length of the redundant substring are combined, and finally the fused feature pattern set is obtained.

If the frequent pattern set obtained at the time of protocol sharding contains the three patterns, "42ad," "2ad2," and "c200," and the positions of these three feature strings are close, the frequency difference is not big, and the difference position is all 1. It is clear that "42ad" and "2ac2" contain redundant strings of "2ad," and the redundant string happens to be the suffix of the previous string and the prefix of the following string. Then the merged string "42ad2" may be the characteristic string of this protocol. Similarly, "2ac200" may also be the characteristic string of this protocol, and the two merged characteristic strings are further merged to obtain "43ac200."

For some protocols of character stream class, this paper adopts the method of using fixed delimiter (e.g., ","; " "; "*n*"; etc.) for segmentation. The position of the word after segmentation is different from *N*-gram algorithm, but the position of the word is in the line. This method effectively solves the character class protocol, similar to HTTP protocol.

4.2. Feature Screen Algorithm. After the *N*-gram data is segmented, the candidate feature pattern set will be obtained, and, from the characteristics of the feature set, there are a lot of redundant and useless features in the set. Therefore, feature screening is required. On the one hand, it is necessary to screen out useless features and reduce the dimension of feature set for subsequent analysis to improve the time efficiency of the algorithm; on the other hand, after filtering out useless features, it can effectively improve the accuracy of features. Finally, the protocol is identified by more accurate feature set after filtering.

Each data stream is segmented into a fixed-length set of data strings. We divided these sets into two random and equal parts. In theory, after statistics, the similarity between the two should be very high. However, if there is some noise string, the similarity of the two samples data will decrease. Therefore, if these redundant strings are removed during the calculation, the calculation results will be improved accordingly. In order to reduce the effect of redundant strings, strings with frequencies lower than the threshold can be filtered in advance. This ensures that the two sample sets are more similar. In the experiment, different thresholds should be selected according to different situations, and the Jaccard coefficients of these two sets after filtering through the threshold should be calculated, respectively. In this way, the similarity of the two sets is the highest when the Jaccard coefficient reaches the maximum. The corresponding

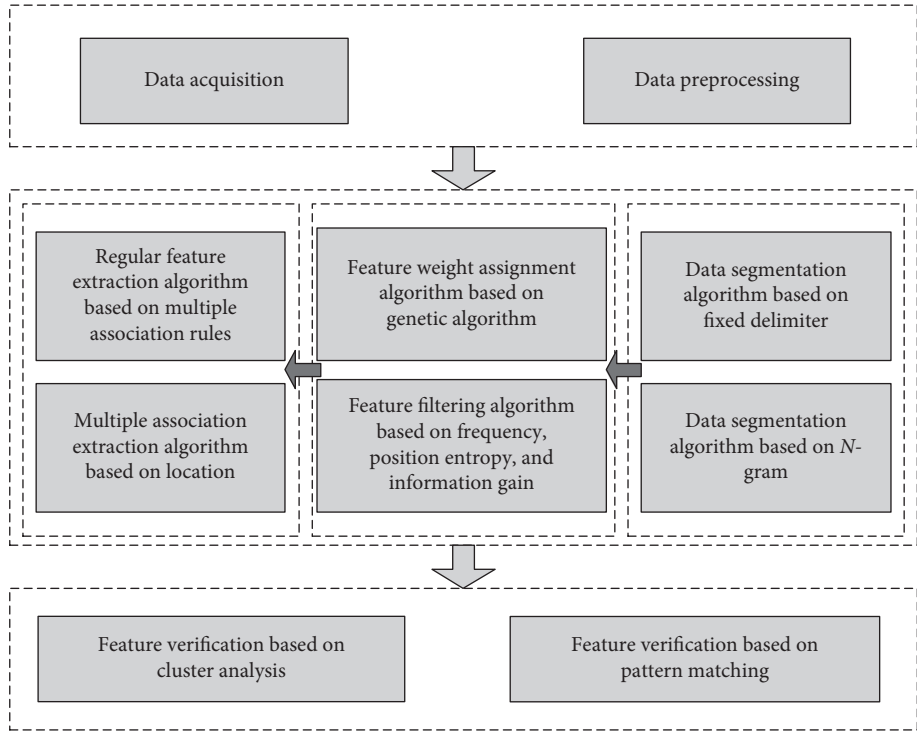

Figure 1: Protocol feature extraction and protocol recognition algorithm architecture.

```
Input: protocol data frame set-DataSet = {I1, I2, I3, . . ., In},
Sliding window length = N
Output: the segmented N-gram dataset-gram_dic
(1) Initialize gram_dic = {}
(2) for each data frame I in DataSet:
(3)     for i in range(len(I) − N + 1):
(4)         gram = I[i : i + N]//Slide window to shard data frames
(5)         gLen = 1
(6)             while(len < N):
(7)                 for j in range(len(gram) − gLen + 1):
(8)                     gram_i = gram[j : j + gLen]
(9)                     if gram in dic:
(10)                        gram_dic[gram].append(i)//Count the locations of each gram
(11)                    else: gram_dic[gram] = [i]
(12)                    end if
(13)                    gLen + = 1
(14)        if gram in dic:
(15)                gram_dic[gram].append(i)//Count the locations of each gram
(16)        else: gram_dic[gram] = [i]
(17)        end if
(18)    end for
(19) end for
(20) return gram_dic
```

Algorithm 1: N-gram stream segmentation algorithm.

threshold is the frequency threshold that needs to be determined in this paper. Finally, according to this threshold, the pattern whose frequency is less than this threshold is eliminated for the subsequent operation.

According to the characteristics of protocol flow data and combined with the results of the N-gram experiment, the Jaccard coefficient needs to be redefined specifically as follows.

Set the whole protocol flow dataset as DataSet, and then perform N-gram segmentation for the whole dataset and divide the results into two subsets of the same size DataSet1 and DataSet2 randomly and equally, and then sort the pattern strings in descending order according to the frequency of occurrence of each pattern string. The two sorted datasets can be as follows:

$$\begin{aligned}\text{DataSet 1} &= \{a\text{Item}_1\colon a\theta_1, a\text{Item}_2\colon a\theta_2, \ldots, a\text{Item}_n\colon a\theta_n\},\\ \text{DataSet 2} &= \{b\text{Item}_1\colon b\theta_1, b\text{Item}_2\colon b\theta_2, \ldots, b\text{Item}_n\colon b\theta_n\},\end{aligned} \tag{1}$$

where θ_i represents the frequency of each feature in the set. According to the above sets, the Jaccard coefficient can be specifically defined in this paper. The definition is shown in the following equation:

$$\text{protocol\_Jaccard} = \frac{\sum_{i=1}^{n} a\theta_i * b\theta_i}{\sum_{i=1}^{n} a\theta_i^2 + \sum_{i=1}^{n} b\theta_i^2 - \sum_{i=1}^{n} a\theta_i * b\theta_i}. \tag{2}$$

We can see from equation (2) that, similar to the original Jaccard coefficient, the value of the redefined Jaccard coefficient is also between 0 and 1. We can see from the formula that the more similar the two sets are, the larger the value will be and the smaller the value will be. The pseudocode of the calculated Jaccard coefficient is shown in Algorithm 2.

According to the above algorithm, the frequency of corresponding maximum Jaccard value can be obtained, and the frequency is set as the filtering threshold, according to which frequent pattern sets can be filtered.

4.3. Regular Expression Extraction Algorithm. In the unknown protocol flow, there are fixed position pattern string and fuzzy string. The final result of the regular expression automatic extraction algorithm is to generate a regular tree to concatenate the fixed string and fuzzy string together. In the genetic programming algorithm, tree coding method is adopted, in which every individual in the population is an effective regular expression tree, and nonleaf nodes in the tree are set as regular expression operators:

(1) Connect nodes ".." to connect leaf nodes or other nodes

(2) Quantity modifiers including "∗+," "++," and "?+"

(3) Group operator "()"

(4) The splitter is "|," which means that the two nodes conduct or operate

The leaves in the tree are as follows:

(1) Characters, numbers, or common symbols

(2) Character ranges such as "[A–Z]" and "[a–z]"

(3) Character class symbols such as "\w" and "d"

(4) Wildcard characters such as "."

The initial population is obtained through random initialization, global space is searched through cross variation between chromosomes and random combination, and the whole algorithm process of "outperforming" is carried out through fitness function.

4.4. Rationality Verification of the Algorithm. We can see from the paper that if the genetic programming can converge to a stable state after a certain number of iterations, applying the problems in this chapter will converge to a more "representative" regular expression. Therefore, to prove the rationality of the algorithm is to prove the above assumptions. To solve the above problems, we need to introduce a Markov chain, which is defined as follows:

(1) The symbol p^t represents the probability of different states at time t

(2) The symbol P represents the matrix for the state transition, where $p_{i,j}$ represents the probability of transition from the ith state to the jth state

(3) The state value of Markov chain at time $t+1$ is only related to the state at time t and can be expressed by a certain probability: $p^{t+1} = p^t P$

(4) If there is a natural number k such that all elements in the matrix P_k are greater than 0, then P is called a prime matrix

Definition 1. Let C, M, and S be the probability transition matrices, where there must be a term greater than 0 in all columns in S, and all elements in M are greater than 0; then all elements in product CMS are greater than zero.

Definition 2. The state transition matrix P is the prime matrix. As k approaches infinity, P_k converges to $P\infty = 1Tp^\infty$, where $p^\infty = p^0 \lim_{k\longrightarrow\infty} Pk = p^0$ is the unique value independent of the initial state, and all elements are greater than 0. This is actually determined by the Markov chain steady-state theorem.

In the proof, the state of the whole population is regarded as a state S of Markov chain, in which the probabilities of selection operation, crossover operation, and mutation operation are combined into a probability transfer matrix. In general, $0 < pm < 1$, $0 \le pc \le 1$, and let M, C, and S, respectively, represent the probability transfer caused by variation, crossover, and selection operation, so the overall probability transfer matrix $P = CMS$.

Probability $M_{i,j}$ of population state evolving from S_i to S_j after individual variation is shown in the following formula:

$$M_{i,j} = (pm)^h (1 - pm)^{n\times 1-h} > 0. \tag{3}$$

In the above equation, h is the sum of the number of genes with different values among each individual of the two populations, from which it can be concluded that M is the prime matrix.

Through individual selection operation, we assume that the probability of population state S_i remaining unchanged is $S_{i,j}$, and the definition is shown in the following formula:

Input: A set of frequent patterns after protocol sharding Set = {S_1, S_2, S_3, . . ., S_n}
Filter the threshold list ThresholdList = [th_1, th_2, . . .]
Output: Threshold value for the highest Jaccard coefficient value
(1) Initialize the Result = {}
(2) Randomly Set two sets, Set*A* and Set*B*:
(3) The *N*-Gram algorithm is called to obtain the frequent pattern set after the two subsets are segmented as gram_list
(4) $J = 0$
(5) for each of these thresholdst in ThresholdList:
(6) for ita in gram_listA and itb in gram_listB://Traverse the two gram_list
(7) if ita < *t*:gram_listA.remove(ita) end if
(8) if itb < *t*:gram_listB.remove(itb) end if//Eliminate items that are smaller than the threshold
(9) if *J* < CAL()://Calculation of Jaccard coefficient
(10) *J* = CAL()
(11) return Result

Algorithm 2: Calculate protocol Jaccard coefficient.

$$S_{i,i} = \frac{\prod_{i=1}^{n} f(I_i)}{\left(\prod_{i=1}^{n} f(I_i)\right)^n}, \quad \left(S_{i,i} > 0\right). \tag{4}$$

All the columns of S must have one element greater than 0. According to Definition 1, we know that the probability transfer matrix P is the prime matrix.

When analyzing the traditional optimization algorithm, the first problem to be considered is whether the optimization algorithm can converge to the global optimal point. Assuming that the fitness value of the global optimal point is max f, the convergence to the global optimal point is defined as

$$\lim_{k \longrightarrow \infty} P\left(\max_{I \in s^k} (\text{fitness}(I)) = \max f\right) = 1. \tag{5}$$

According to Definition 2, it can be known that the typical genetic algorithm will converge to a probability distribution where the probability of all population states is greater than 0. The population obtained by each iteration will eventually have the highest evaluation value of the individual. The continuous practice of finding the optimal solution will make the above formula true, so eventually we will obtain the optimal solution in the whole search space.

5. Experiments

5.1. Environment Settings. This section will verify the algorithms proposed above, test the feasibility and effectiveness of the feature extraction algorithm for private protocols, and obtain the efficiency of algorithm operation. The test environment of this algorithm is shown in Table 1.

The data sources in the experiment are divided into two parts: the known protocol data and the unknown format protocol data. The known protocol data is selected from the DARPA-2000 dataset. This dataset contains 58 kinds of typical attack data streams. These streams can be divided into five typical attack categories: DOS, U2R, Probe, R2L, and Data. As one of the most comprehensive protocol datasets, this dataset is widely used in intrusion detection, protocol analysis, protocol identification, and other fields. In our experiments, three known protocols, ARP protocol, ICMP protocol, and HTTP protocol, were selected from the experimental data.

Table 1: Experiments environment table.

Environment	Configuration information
Operating system	Windows 10 Professional (64-bit)
Processor	Intel(R) Core(TM) i7-6700 @3.4 GHz
Memory size	16 GB

For other types of unknown protocol, the packets are mainly captured from the multiple log system running under Linux. After that, the private protocol data frame is obtained through the data preprocessing module. Then feature extraction and validation are performed for the private protocol used during log transfer. Thus, the method proposed in this paper is verified. The unknown format protocol data used in this paper is the data transmitted by FASP protocol. As an efficient big data transmission technology, FASP protocol performs well in various WAN transmission speed tests and has been applied in many different fields such as life science, cloud computing, and media. Because FASP protocol is a patent protocol, the format of the protocol is not disclosed, and the data format of its transmission is not disclosed, so the protocol data is suitable for use as the test data of the system.

5.2. Experimental Results

5.2.1. Experiment of Data Stream Segmentation. First, the experimental data head is processed to obtain effective experimental data. FASP, HTTP, ARP, and ICMP protocols were used in the experiment. For the data segmentation algorithm, the Jaccard coefficient was calculated for the four protocols mentioned above, and the threshold value was calculated as shown in Figures 2–5. The abscissa represents the frequency, and the ordinate is the Jaccard coefficient.

We can see that the corresponding frequency threshold is different, when the Jaccard coefficient of the four protocols is the highest. The threshold for ARP protocol is about 600, the peak value of threshold for FASP protocol is about 2200,

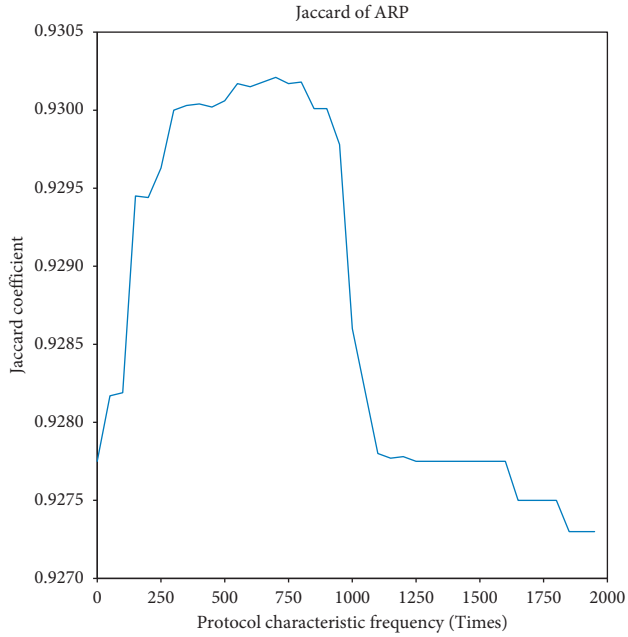

Figure 2: Jaccard coefficient of ARP protocol.

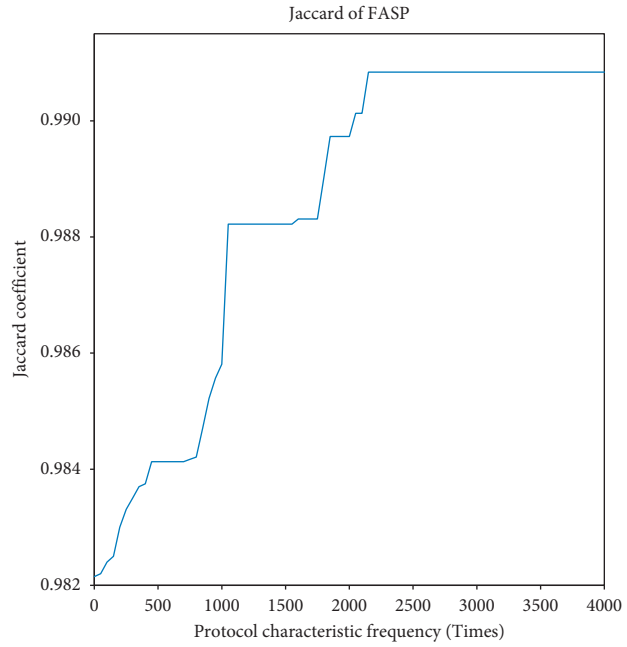

Figure 3: Jaccard coefficient of FASP protocol.

the threshold of ICMP protocol is 700, and the threshold for the HTTP protocol is about 370, when the Jaccard coefficient reaches its maximum. Therefore, for these four protocols, after N-gram segmentation, we can carry out frequency filtering and screen the candidate pattern string according to the above experimental results. Thus, the efficiency and accuracy of subsequent algorithms can be improved.

5.2.2. Experiment of Pattern Set Extraction. The candidate schema sets extracted by the four protocols are shown in Table 2: Each protocol has several candidate schema sets, and a large number of redundant features are removed after the schema fusion. Take FASP protocol for example. FASP is an unknown protocol, but, after the preliminary artificial reverse analysis and compared with the public information, its protocol format information can be obtained roughly. Based on the artificial reverse result, the frequent string "0×56" is the transport flag, and the frequent strings "0×19" and "$0 \times 1A$" are the initial transmission identity and the retransmission data identity, respectively. Similar results can be achieved with other protocols. According to the above results, the feature pattern set after data segmentation and screening has certain representativeness. However, there are still some noise data, which need further processing.

5.2.3. Experiment of Regular Expression Extraction. The association relation between pattern strings is selected according to the occurrence location of pattern strings. The regular rules obtained through association rules are shown in Table 3. The association relationship between the mining feature strings can be seen from Table 3. The regular expression can be successfully generated based on the association relationship and the position difference between the pattern strings. Since we did not mine associations between frequent patterns sets of the ICMP protocol, the regular features degenerate to string features.

5.2.4. Analysis of Feature Verification Results. Figures 6 and 7, respectively, show the experimental results obtained by using frequent pattern sets and regular features for these four protocols. The experiments used pattern matching and cluster analysis as the feature verification methods. The input data is mixed data.

It can be seen from Figure 6 that, for the four selected protocols, when using pattern matching algorithm for feature verification, since HTTP protocol and ARP protocol extract long feature strings, the recognitions of ARP and HTTP by using frequent pattern set and regular pattern can both reach 100%. For ICMP protocol, when using frequent pattern set, the recognition rate is 96.7%, and when using regular pattern based on association rule, the recognition rate is 97.9%. With the increase of rule strength, the recognition rate increases. For FASP protocol, the recognition rates are 92.4% and 93.8%, respectively, which are slightly lower than the recognition rates of known protocols. It can also be observed that the recognition rate increases significantly with the increase of feature intensity.

As can be seen from Figure 7, the recognition rate of the four protocols is relatively high when using cluster analysis. When using various features to classify FASP protocols, the recognition rate of the protocols is slightly lower than that when using pattern matching.

To sum up, it can be seen that the effect of frequent pattern feature and association rule feature in recognition and classification gradually becomes higher. For the unknown protocol FASP, the result of clustering analysis using mined features is better than that of matching analysis. For ARP and HTTP protocols, the above methods have

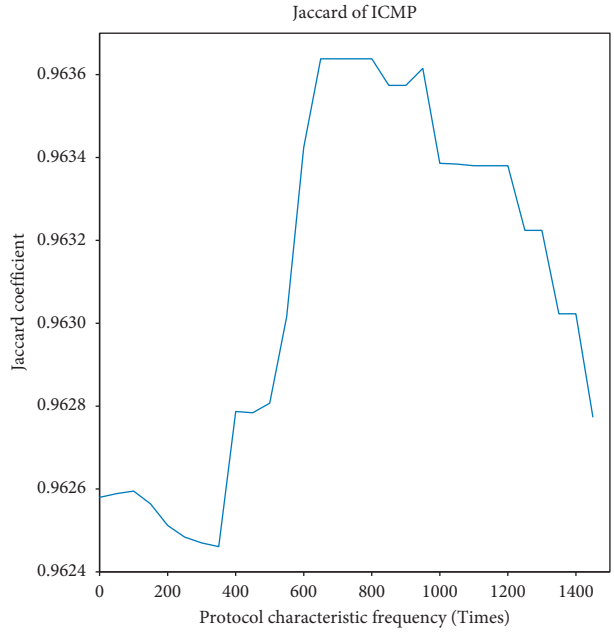

Figure 4: Jaccard coefficient of ICMP protocol.

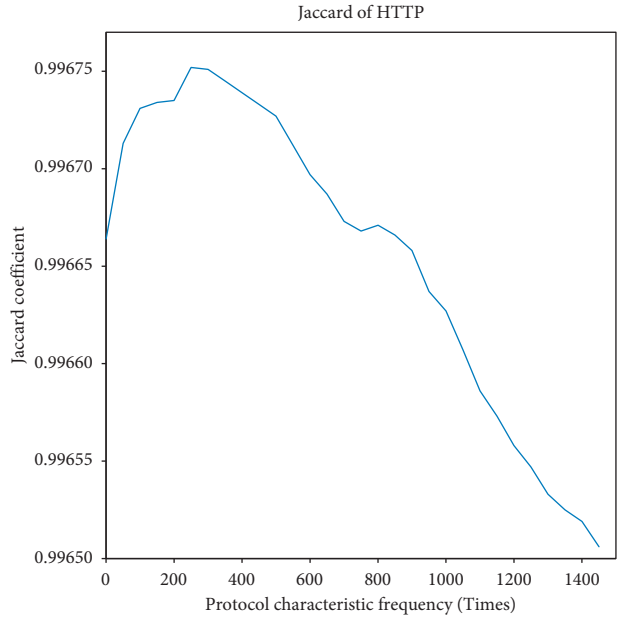

Figure 5: Jaccard coefficient of HTTP protocol.

Table 2: Frequent pattern extraction result instances with location information.

Protocol	Frequent pattern instance
FASP	[(“0 × 56,” 0), (“0 × 00000,” 8), (“0 × 0000,” 16), (“0 × 19,” 2), (’0x1a′, 2), (“0 × 2,” 4), (“0 × 3.” 4), (“0 × 0000,” 20), (“0xff,” 20),.....]
ICMP	[(“0 × 0501,” 0), (“0xac10720245,” 8), (“0 × 0,” 20), (“0 × 000,” 29), (“0 × 00,” 18), (“0 × 02,” 21), (“0 × 04,” 21),.....]
HTTP	[(“0 × 474554202f,” 0), (“0 × 485454502f312e31,” 0), (“0 × 4163636570743a,” 16), (“0 × 486f73743a,” 32), (0“x446174653a,” 32), (“0 × 5365727665723a,” 16),.....]
ARP	[(“0 × 000108000604000,” 0), (“ × 1,” 15), (“0 × 2,” 15), (“0 × 0,” 16), (“0xac107,” 28), (“0 × 0100,” 34),.....]

Table 3: Protocol regular features for the four protocols.

Protocol	Protocol regular feature
FASP	$0\times56(0\times19\|0\times1a).+0\times00000.+0\times0000$
ICMP	0×0501
HTTP	$((0\times474554202f.+0\times4163636570743a.+0\times486f73743a)\|(0\times485454502f312e31.+0\times5365727665723a.+0\times446174653a)$
ARP	$0\times000108000604000(0\times1\|0\times2)$

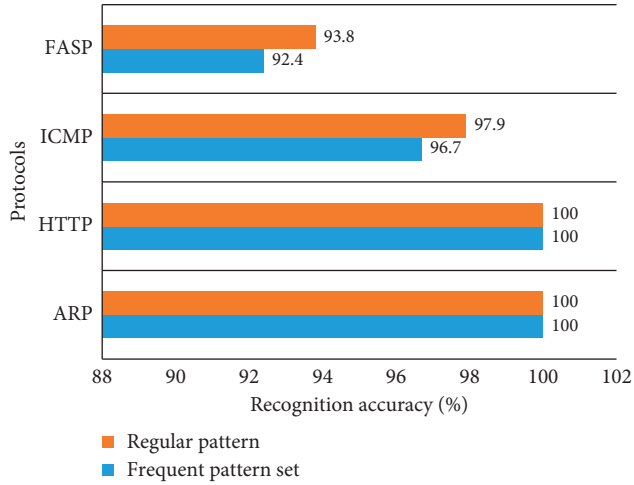

Figure 6: Feature verification algorithm based on pattern matching.

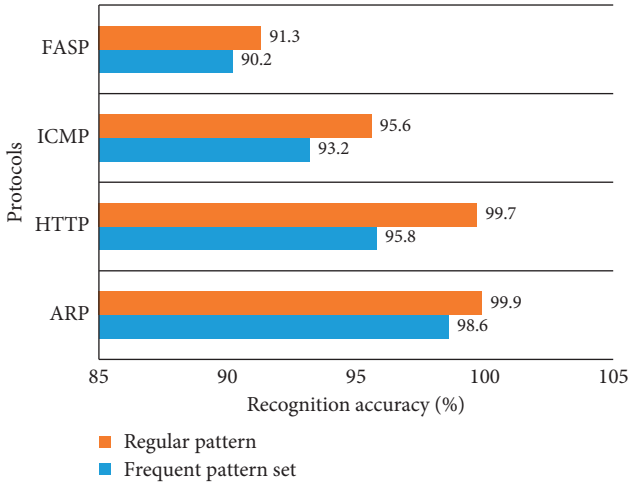

Figure 7: Feature verification algorithm based on cluster analysis.

recognition rates close to 100%. Meanwhile, for FASP, a private protocol, the recognition rate can reach about 93.8%.

6. Conclusion

The identification of private protocols is of great significance to prevent the abuse of users' private data in 5G network. This paper summarizes the current research methods for protocol recognition and analysis and proposes a set of feature extraction and recognition algorithms for unknown private protocols combined with genetic algorithm and association rule algorithm. The detailed design and experiment of each algorithm are carried out. Finally, the practical significance of the feature is extracted based on the analysis of the actual data and verified through the experiments. Experimental results show that the proposed method is effective in identifying private protocols.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Key R&D Program of China (no. 2016QY05X1000), the National Natural Science Foundation of China (nos. 61872111 and 61402137), and Basic Research Program (no. JCKY2019210B029).

References

[1] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: multilevel traffic classification in the dark," in *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM 2005*, pp. 229–240, New York, NY, USA, August 2005.

[2] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in *Proceedings of the 6th ACM SIGCOMM on Internet measurement, IMC 06*, p. 189, Rio de Janeiro, Brazil, October 2006.

[3] A. V. Aho and M. J. Corasick, "Efficient string matching," *Communications of the ACM*, vol. 18, no. 6, pp. 333–340, 1975.

[4] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963–1971, 2020.

[5] Y. Fan, Y. Zhu, and L. Yuan, "Automatic reverse engineering of unknown security protocols from network trace," in *Proceedings of the 4th IEEE International Conference on Computer and Communications (ICCC)*, IEEE, Chengdu, China, December 2018.

[6] H. Gascon, C. Wressnegger, and F. Yamaguchi, *Pulsar: Stateful Black-Box Fuzzing of Proprietary Network Protocols Security and Privacy in Communication Networks*, pp. 330–347, Springer International Publishing, Berlin, Germany, 2015.

[7] Z. Tian, W. Shi, Y. Wang et al., "Real-time lateral movement detection based on evidence reasoning network for edge computing environment," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4285–4294, 2019.

[8] Q. Tan, Y. Gao, J. Shi, X. Wang, B. Fang, and Z. Tian, "Toward a comprehensive insight into the eclipse attacks of tor hidden services," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1584–1593, 2019.

[9] V. D. J. Merwe, R. Caceres, Y. Chu et al., "mmdump: a tool for monitoring internet multimedia traffic," *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 5, pp. 48–59, 2000.

[10] H. J. Kang, M. S. Kim, and W. K. Hong, *A Method on Multimedia Service Traffic Monitoring and Analysis*, Springer, Berlin, Germany, 2003.

[11] A. G. Medrano-Chávez, E. Pérez-Cortés, and M. Lopez-Guerrero, "A performance comparison of chord and kademlia DHTs in high churn scenarios," *Peer-to-Peer Networking and Applications*, vol. 8, no. 5, pp. 807–821, 2015.

[12] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures," in *Proceedings of the 13th International WWW Conference*, New York, NY, USA, May 2004.

[13] Z. Zhang, Z. Zhang, P. P. C. Lee, Y. Liu, and G. Xie, "Toward unsupervised protocol feature word extraction," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 10, pp. 1894–1906, 2014.

[14] Y. Wang, X. Yun, M. Z. Shafiq et al., "A semantics aware approach to automated revers engineering unknown protocols," in *Proceedings of the 2012 20th IEEE International Conference on Network Protocols (ICNP)*, IEEE, Austin, TX, USA, November 2012.

[15] W. H. Turkett, A. V. Karode, and E. W. Fulp, "In-the-dark network traffic classification using support vector machines," in *Proceedings of the 23rd Conference on Artificial Intelligence, AAAI 2008*, Chicago, IL, USA, July 2008.

[16] W. Zhou, L. Dong, L. Bic et al., "Internet traffic classification using feed-forward neural network," in *Proceedings of IEEE 2011 International Conference on Computational Problem-Solving (ICCP)*, pp. 641–646, Chengdu, China, October 2011.